

Zustandsorientierte Modellierung

Grundgedanken zur Umsetzung dieses Themas im Informatikunterricht der Mittelstufe

Matthias Spohrer

Institut für Informatik der
Technische Universität München
Boltzmannstr. 3, 85748 Garching
Matthias.Spohrer@in.tum.de

Abstract: Zustandsorientierte Modellierung und endliche Automaten als Lehreinheit im Informatikunterricht der Sekundarstufe? Folgender Aufsatz soll einerseits kurz deren Anspruch als Teil des neuen Lehrplans rechtfertigen, andererseits einen Vorschlag zu ihrer Umsetzung im Unterricht machen. Hierzu werden zuerst theoretische Grundüberlegungen erläutert, die z.B. eine verbindliche Notation und den Grad der Abstraktheit betreffen, im Anschluss wird dann konkret ein möglicher Stundenverlauf vorgestellt.

1 Einleitung

Demnächst wird Informatik Pflichtfach an Bayerns Gymnasien. Über die Lehr- und Lerninhalte des neuen Fachs ist bereits viel diskutiert worden. „Fähigkeiten wie das gezielte Finden, das Selektieren, Interpretieren und Präsentieren von Informationen, das Nutzen und Erkennen von Querbezügen gewinnen mehr und mehr an Bedeutung; eine tragende Rolle spielt dabei der Einsatz moderner Informationstechnologie. Kenntnisse über Grundlagen der automatischen Darstellung und Verarbeitung von Information sind Voraussetzung für eine sinnvolle, zweckgerichtete und verantwortungsbewusste Nutzung und Mitgestaltung informationstechnischer Systeme und müssen heute als Bestandteile der Allgemeinbildung betrachtet werden.“ Diese Aussage des Ministerialdirigenten Dr. Peter Müller¹ spiegelt ganz deutlich die Intention des neuen Pflichtfaches Informatik wider. Hubwieser [Hu00, S.78] bezeichnet sie als „informationszentrierten Ansatz“, welcher nicht den Algorithmus, sondern die Information, ihre Darstellungsformen und Interpretationsmöglichkeiten in den Mittelpunkt stellt. Im Entwurf des neuen Lehrplans für die 10. Jahrgangsstufe liest man „Mit der Zerlegung von Abläufen in Zustände und Zustandsübergänge lernen die Jugendlichen eine zweckdienliche Methode zur Strukturierung von Prozessabläufen kennen. Die Beschreibung der Zustandsübergänge führt sie zu den Grundprinzipien von Algorithmen.“ [S3, S.5]. Die zustandsorientierte Modellie-

¹ [S1]: Dr. Peter Müller im Rundschreiben 75412 des Bayerischen Staatsministeriums für Unterricht und Kultus an die Leiterinnen und Leiter der Gymnasien in Bayern.

rung leistet also einen wichtigen Beitrag zum strukturiertem Denken, aber auch zum Verständnis von Algorithmen. In einer immer mehr technisch orientierteren Welt mit immer schnelleren Abläufen wird man das „ruhige Denken in Zuständen“ oft brauchen können und schnell zu schätzen lernen.

Wie wird die zustandsorientierte Modellierung dem Alltag gerecht? Jeder kennt die Situation: man sitzt vor irgendeinem technischen Gerät, beispielsweise einem Videorekorder, und wundert sich, dass dieser nicht so funktioniert, wie man es sich eigentlich vorstellt. In den seltensten Fällen handelt es sich hierbei um einen technischen Defekt, meist ist nur die Bedienung fehlerhaft. Man drückt verzweifelt eine oder mehrere Tasten, ohne dass das Gerät wie gewünscht reagiert. Der Rekorder befindet sich augenblicklich in einem Zustand, aus dem das Drücken der betätigten Tasten nicht herausführt. Nur das Aktivieren bestimmter Knöpfe führt den Rekorder in einen neuen (gewünschten oder unerwünschten) Zustand über. Automaten sind ein Bestandteil des alltäglichen Lebens. Jeder von uns gerät immer wieder an die verschiedensten Automaten, seien es welche für Fahrscheine, Getränke, Süßigkeiten, Parktickets, Geld oder ähnlichem. Manche Geräte entpuppen sich erst bei genauerem Hinsehen als Maschinen, die ebenfalls verschiedene Zustände annehmen können und auf die unser Modell ebenso übertragbar wäre. Dazu gehören Fahrstühle, Ampeln, Alarmanlagen und Mixer, eigentlich nahezu jedes andere technische Gerät, z.B. der Computer als Beispiel für einen sehr komplexen Automaten. Ja sogar ein Kugelschreiber lässt sich als ein sehr einfacher Automat mit zwei Zuständen auffassen. Auch Schülerinnen und Schüler der Mittelstufe verfügen diesbezüglich sicher über ausreichend Erfahrung und hatten schon mal Schwierigkeiten mit der Bedienung irgendeines Automaten, jedoch kaum jemand wird sich bislang grundlegende oder weiterführende Gedanken darüber gemacht haben.

2 Theoretischer Hintergrund

Die formale Definition eines endlichen Automaten (EA) als 5-Tupel (nachzulesen z.B. bei [Sc97]) wird für Schüler kaum einsichtig sein (insbesondere die abstrakte Definition der Überfunktionsfunktion) und sollte daher auf ein für Schüler der Mittelstufe geeignetes Niveau vereinfacht werden. Diese „Reinform“ eines EA wird im Folgenden durch die Akzeptoren (s. u.) realisiert, bildet jedoch für den kompletten Entwurf die Grundlage. Oft wird auch bei *Übergangsdigrammen mit endlich vielen Zuständen*, die darüber hinaus noch Bedingungen oder ausgelöste Aktionen beinhalten, irreführender Weise von „endlichen Automaten“ gesprochen, jedoch sind diese von o.g. EA zur Prüfung regulärer Sprachen im Sinne der Chomsky-Klassifizierung zu trennen, da allgemeine Übergangsdigramme mit endlichen vielen Zuständen z.B. auch Turingmaschinen beschreiben könnten (vgl. dazu [Hu00, S.172]). Da jedoch selbst in der Fachliteratur keine strikte Trennung zu finden ist, ist meines Erachtens in der Schule eine Unterscheidung nicht erforderlich und auch ich spreche hier stets von „endlichen Automaten“, selbst wenn diese später (um ausgelöste Aktionen erweitert) nicht mehr vollständig den o.g. „Chomsky-Typ-3-Maschinen“ genügen. Wie folgend erläutert, können Schüler über Gemeinsamkeiten der ihnen bekannten Automaten selbstständig ein abstraktes Modell erschaffen, welches ohne Kenntnis und Verwendung der formalen Definition einen endlichen Automaten eindeutig charakterisiert.

Die graphische Darstellung eines endlichen Automaten, ein so genanntes *Zustandübergangsdiagramm* (oder kurz *Zustandsdiagramm* oder *Übergangsdiagramm*), beschreibt einen endlichen Automaten ebenfalls vollständig und ist aufgrund dessen Anschaulichkeit sicherlich die geeignetste Möglichkeit für die Darstellung eines solchen im Unterricht, so lange sich die Anzahl der verschiedenen Zustände in einem übersichtlichen Rahmen hält. Daher bietet sich für die Schule folgende „Definition“ an, welche auch gleich eine verbindliche Notation für die Zustandsdiagramme enthält:

Ein *endlicher Automat* hat folgende Eigenschaften:

- Er kann eine feste Anzahl von verschiedenen **Zuständen** annehmen; diese werden von Kreisen oder Ellipsen dargestellt, die die Namen oder Nummern der Zustände enthalten².
- Genau ein Zustand ist der Ausgangs- oder **Startzustand**; dieser wird durch einen kurzen Pfeil gekennzeichnet, dessen Spitze in den Zustand zeigt.
- Manche Zustände sind **Endzustände**; Endzustände werden durch Doppelkreise bzw. Doppellellipsen dargestellt.
- Es besteht die Möglichkeit, den Automaten durch verschiedene Eingaben zu bedienen. Die Menge der möglichen Eingaben wird **Eingabealphabet**³ genannt, ihre Elemente nennt man auch (Eingabe-) *Zeichen*. Die Anzahl der verschiedenen Zeichen ist endlich.
- Bei Eingabe eines Elements des Eingabealphabets erfolgt stets ein **Übergang** von einem Zustand 1 in einen Zustand 2. Dieser wird durch einen Pfeil von Zustand 1 nach Zustand 2 dargestellt, der das entsprechende Zeichen der auslösenden Eingabe als Markierung trägt. Da beide Zustände nicht unbedingt verschieden sein müssen, kann ein Zustand auch auf sich selbst zeigen, was bedeutet, dass der Automat bei der entsprechenden Eingabe im selben Zustand bleibt.
- Von einem Zustand dürfen keine zwei Pfeile mit der gleichen Markierung auf verschiedene Zustände verweisen, d.h. der Übergang von einem Zustand auf einen anderen muss bei jeder Eingabe eindeutig sein. Genauso muss für jede möglich Eingabe ein Übergang existieren⁴.

Da nach dem Satz von Rabin und Scott [Sc97, S.32f]. jeder nichtdeterministische Automat in einen äquivalenten deterministischen Automaten transformierbar ist, ist die Beschränkung auf Letztere meines Erachtens nach sinnvoll, da nichtdeterministische Maschinen zwar flexibler, aber auch deutlich schwerer begreifbar sind. Die Möglichkeit,

² Im Programm Charon besteht aus Gründen der Übersicht auch die Möglichkeit die Namen oben rechts neben die Zustände zu setzen.

³ [Hu00], S. 171 spricht von „*auslösenden Aktionen*“ bei o.g. allg. „Automaten mit endlich vielen Zuständen“.

⁴ Soll der Automat bei Eingabe von bestimmten Zeichen gar keine Reaktion zeigen, oder modelliert man nur einen Teilausschnitt eines Automaten, der bei Eingabe dieses Zeichens in einen Zustand außerhalb jenes Teilautomaten übergeht, so kann der entsprechende Pfeil im Diagramm auch weggelassen werden.

dass eine Maschine bei identischer Bedienung mal so und mal so reagieren soll, erscheint sicherlich nicht nur Schülern als fragwürdig. Daher sollen diese hier und im Unterricht nicht weiter behandelt werden, zumal selbst in der Software-Technik nichtdeterministische Automaten keine Rolle spielen [Bal96, S. 277].

In der Informatik sind insbesondere solche Automaten interessant, die bei Eingabe einer bestimmten Kombination aus Zeichen des Eingabealphabets entweder in einen gültigen Endzustand übergehen und die Eingabe daher akzeptieren, oder bei einem Nicht-Endzustand stehen bleiben und die Eingabe deswegen ablehnen. Derartige endliche Automaten nennt man deswegen auch **Akzeptoren**, eine Kombination aus mehreren (aber endlich vielen) Zeichen des Eingabealphabets bezeichnet man als **Wort**, und die Fragestellung, ob ein eingegebenes Wort vom Automaten akzeptiert wird oder nicht ist allgemein als „**Wortproblem**“ in der theoretischen Informatik bekannt. Dahinter verbirgt sich zugleich die Fragestellung, ob das entsprechende Wort Element der *formalen Sprache* ist, die der zugehörige Automat beschreibt. Da Sprachen und Grammatiken aufgrund ihrer hohen Abstraktheit der theoretischen Informatik anzusiedeln sind, und im Mittelstufenunterricht, wo der Modellierungscharakter im Vordergrund steht, nicht behandelt werden sollten, möge dem interessierten Leser an dieser Stelle der Verweis auf entsprechende Fachliteratur, z.B. [Sc97] oder [Br98], genügen.

Alle Automaten, die unseren bisherigen Eigenschaften genügen, haben gemein, dass man sie zwar durch verschiedene Eingaben bedienen kann, sie aber für den Anwender keine relevanten Aktionen auslösen. Einzig Akzeptoren geben, wie eben gesehen, quasi einen booleschen Wert aus, der besagt, ob das eingegebene Wort erkannt wurde oder nicht. Im nun folgenden Abschnitt werden die Automaten dadurch erweitert, dass sie bei Erreichen eines gewissen Zustandes eine Aktion auslösen können. Ein Cola-Automat sollte in einem bestimmten Zustand das gewünschte Getränk und das Rückgeld herausgeben, von einer Musicbox erwartet man das Abspielen der gewählten Schallplatte und eine Ampelanlage sollte bei Erreichen eines neuen Zustandes entsprechendes Licht anzeigen. Normalerweise wird die während eines Zustandes ausgelöste Aktion im Übergangdiagramm direkt im Zustand innerhalb des Kreises unterhalb dessen Namen geschrieben, aus Platzgründen können sämtliche Aktionen jedoch auch gesondert zusammen mit dem dazugehörenden Zustand angegeben werden.⁵

Anstelle der bisher behandelten graphischen Lösung in Form des Diagramms mit Kreisen und Pfeilen lassen sich endliche Automaten ebenso mit **Zustand(übergang)tabellen** vollständig beschreiben. Diese sollten auch mit den Schülern behandelt werden, da sie eine einfache, aber eindeutige Notation ermöglichen und Übergangdiagramme bei sehr vielen Zuständen schnell unübersichtlich werden. Jeder Zustand erhält eine Zeile, jedes Eingabezeichen eine Spalte der Tabelle. In das Feld $[i, j]$ der Tabelle wird derjenige Zustand eingetragen, in den der Automat bei Erhalt des Zeichens der Spalte j übergehen würde, solange er sich im Zustand der Zeile i befindet. Um sich Schreibarbeit zu ersparen, können die Namen der Zustände auch abgekürzt oder durchnummeriert werden. Anstelle des Namens wird im Programm die eindeutig vergebene Nummer des gewünschten Zustandes in die Übergangstabelle eingetragen. Die Tabelle kann außerdem

⁵ Auf eine Differenzierung zwischen „Moore“- und „Mealy“-Automaten wird hier verzichtet.

um weitere Spalten ergänzt werden, so dass die bisher reine Zustandübergangstabelle zu einer **Zustandstabelle** erweitert wird, die sämtliche wesentlichen Eigenschaften der Zustände beinhaltet. Im später vorgestellten Werkzeug „Charon“ kann man Kommentare zu den einzelnen Zuständen, die eine genauere Beschreibung zulassen, editieren, den jeweiligen Typ anzeigen oder auch die ausgelösten Aktionen eingeben. Ich schlage daher eine ähnliche Notation auch für den Unterricht vor.

3 Der Unterrichtsentwurf

In diesem Kapitel wird nun eine kurze Unterrichtssequenz vorgestellt, in der die Schüler mit Hilfe von Charon erste Erfahrungen mit deterministischen endlichen Automaten machen. Der Vorschlag enthält keine exakten Zeit- oder Stundenangaben, da sich diese aufgrund fehlender Hintergrundinformationen wie Vorwissen der Schüler oder Rechnerausstattung der Schule schwer einschätzen lassen. Als Richtlinie kann aber jedes Unterkapitel als eigene Stunde angesehen werden, wobei diese im einstündigen Fall, insbesondere bei eventuellem Fachraumwechsel, Rechnerhochfahren etc., jedoch äußerst knapp bemessen wäre. Aus Platzgründen wird hier nur ein stark gekürzter, knapper Verlauf ohne Materialien o.ä. angegeben. Es existiert auch ein konkreter Vorschlag für eine komplette Unterrichtseinheit inklusive verschiedener Materialien wie Übungsblätter, sowie dem eigens dazu entwickeltem Programm „Charon“, mit dessen Hilfe beliebige endliche Automaten sehr einfach erzeugt und getestet werden können. Mehr zu Charon im 4. Kapitel. Außerdem gibt es weitere Infos sowie eine Downloadmöglichkeit voraussichtlich in Kürze unter www.spohrer.net/charon.

3.1 Einführung

Die Hinführung der Schüler auf ein neues Thema bereitet erfahrungsgemäß die größten Schwierigkeiten. Zielgerichtet sollten die Schüler auf die neue Problematik vorbereitet und zum Auffinden von Lösungsmöglichkeiten motiviert werden. An dieser Stelle bietet es sich oftmals an, die Schüler mit einem Problem aus ihrem alltäglichen Leben zu konfrontieren und sie selbstständig nach Lösungen suchen zu lassen. Ihre Ideen und Ergebnisse bieten dann einen Einstieg für die gesuchte Disziplin.

Hier wäre folgende Möglichkeit denkbar: Fast jeder Schüler besitzt eine eigene Armbanduhr. Jeder tauscht nun seine Uhr mit der seines Nachbarn, sofern es sich um ein Modell handelt, welches mehrere Funktionen oder Einstellmöglichkeiten besitzt. Falls nicht ausreichend Uhren zur Verfügung stehen, kann der Lehrer noch Stoppuhren aus der Sport- oder Physksammlung besorgen. Die Schüler bekommen nun den Auftrag, innerhalb der nächsten 10 bis 15 Minuten eine Art Bedienungsanleitung für die Uhr zu erstellen. Da die Uhren über verschieden großen Funktionsumfang verfügen, ist zu erwarten, dass einige Schüler nur einen Teil der Funktionen erarbeiten können, was jedoch keine Rolle spielt. Im Anschluss sollten einige ihren Entwurf vorlesen, wobei Auffälligkeiten und Gemeinsamkeiten festgehalten werden. Aus Sätzen wie *„durch Drücken der Taste 2 gelangt man in die Stopp-Funktion. Die LCD-Anzeige blinkt“* lassen sich später

sehr schön die Begriffe „Eingabe“ („Drücken der Taste 2“), „Zustand“ („Stopp-Funktion“) und „ausgelöste Aktion“ („LCD-Anzeige blinkt“) aufgreifen.

Manche Schüler werden feststellen, dass die gestellte Aufgabe gar nicht so einfach zu lösen ist, während andere aufgrund der geringen Einstellmöglichkeiten ihrer Uhr weniger Probleme hatten. Die Schüler erkennen, dass die Bedienung der Uhr desto schwieriger ist, je größer ihr Funktionsumfang ist. Der Lehrer könnte als weiteres Beispiel eine ähnliche Geschichte, wie über den in der Anleitung erwähnten Videorekorder erzählen und so das Thema auf „Automaten“ ausweiten. Im anschließenden Gespräch sollen die Schüler weitere Beispiele für Automaten nennen, vielleicht ihre eigenen Erfahrungen schildern und schließlich die Gemeinsamkeiten für alle von ihnen vorgestellten Automaten herausfiltern. Dabei könnten verschiedene in der Schule vorhandenen Automaten (Video, Getränkeautomat) als Anschauungsmaterial dienen. Die Ergebnisse werden stichpunktartig an der Tafel festgehalten. Folgende Erkenntnisse sollten erreicht werden:

- jeder Automat wird durch Eingaben bedient,
- ein Automat kann verschiedene Zustände annehmen, in denen er auf weitere Eingaben wartet,
- zu jedem beliebigen Zeitpunkt befindet sich der Automat in einem bestimmten Zustand (hierbei wird stillschweigend vorausgesetzt, dass Übergänge unmittelbar ablaufen, d.h. ohne Zeitverlust),
- einige Automaten liefern eine Ausgabe,
- um ein gewünschtes Ergebnis zu erhalten, muss eine bestimmte Eingabereihenfolge eingehalten werden.

Sicher werden auch andere Punkte genannt werden, oben genannte Eigenschaften könnten von den Schülern natürlich auch noch deutlich unschärfer formuliert sein. Da die bisherige Aufstellung eine strukturlose Zusammenstellung von Charakteristika verschiedener Automaten ist, ist es im Folgenden nun Sache des Lehrers, einen deterministischen Automaten exakt zu definieren (siehe Abschnitt 3.2). Hausaufgabe könnte sein, einen weiteren Automaten (z.B. einen Fahrstuhl) mit Hilfe eben genannter Punkte genauer zu charakterisieren. Überlegungen wie „Welches sind die möglichen Eingaben?“ oder „Was sind die möglichen Zustände?“ führen zu Beginn der nächsten Stunde wieder zurück zur Thematik und unterstützen nachfolgende „Definition“.

3.2 Definition endlicher Automaten und Übergangsdigramme

Nachdem die Schüler nun selbständig auf verschiedene Art und Weise versucht haben, eine geeignete Beschreibung für die Funktionsweise ihrer (Stopp-)Uhr zu finden und daraufhin einige Eigenschaften von Automaten entdeckt haben, ist es nun an der Zeit, eine einheitliche Darstellung für einen endlichen Automaten festzulegen. Dies muss wohl sehr lehrerzentriert vonstattengehen, um sicher zu stellen, dass alle wesentlichen Merkmale und ihre Bezeichnungen scharf herausgearbeitet werden. Als Hefteintrag bietet sich hierzu die unter Abschnitt 2 dargestellte Liste an, welche einen deterministischen endlichen Automaten schülergerecht und genau genug charakterisiert.

Da die graphische Notation bislang nicht erwähnt worden ist, kann diese nun parallel zu den Eigenschaften und der daraus resultierenden „Definition“ des Automaten entwickelt werden. Der Lehrer hat hier die Möglichkeit, die entsprechenden Symbole gleich an

einem konkreten Beispiel einzubauen und so zu veranschaulichen. Anbieten würde sich hierzu ein sehr einfaches Beispiel, wie folgender „Fahrstuhl-Automat“, der vereinfacht über drei Zustände verfügt: „Erdgeschoss“, „1. Stock“ und „2. Stock“. Sein Eingabealphabet besteht demnach aus den Zeichen „EG“, „1“ und „2“. Zu Beginn befindet sich der Fahrstuhl im Erdgeschoss, was dem Startzustand entspricht. Durch Eingabe des Zeichens „2“ (welche durch das Drücken des Knopfes „2“ realisiert wird) geht der Fahrstuhl in den Zustand „2. Stock“ über, fährt also in das zweite Obergeschoss, bleibt dort stehen und wartet auf eine neue Eingabe. Beachten Sie, dass der Fahrstuhl beim Übergang zwar räumlich den 1. Stock passiert, den Zustand jedoch nicht einnimmt (er bleibt ja nicht stehen), sondern direkt ins obere Stockwerk fährt. Dieser Fahrstuhl könnte so schrittweise zusammengesetzt werden. Dies kann, solange sich die Größe und Komplexität des gewählten Automaten noch im Rahmen hält, entweder an der Tafel, mit Hilfe einer vorbereiteten Präsentation oder gleich mit Charon geschehen⁶. Weitere Beispiele sollten das Gelernte vertiefen. Als Hausaufgabe bietet sich beispielsweise folgende Übung an: Die Schüler sollen die wörtliche Beschreibung eines Automaten in ein Zustandsübergangsdiagramm übersetzen und umgekehrt ein weiteres Diagramm interpretieren.

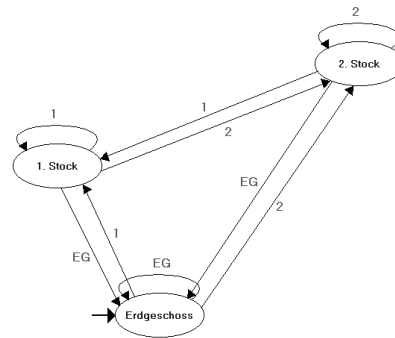


Abbildung 1: Zustandsübergangsdiagramm eines Fahrstuhls, erstellt mit Charon

3.3 Zustandstabellen und ausgelöste Aktionen

Nach Verbesserung der Hausaufgabe, für die man durchaus einige Zeit einkalkulieren muss, bietet es sich an, gleich die Darstellung eines Automaten in Tabellenform zu behandeln, da man sehr bald sieht, dass Zustandsübergangsdiagramme bei vielen Zuständen und Eingabezeichen schnell unübersichtlich werden.

Hat man die Tabelle den Schülern nahegebracht, wird es keine Schwierigkeit sein, diese um weitere Spalten

Zustand	Name	EG	1	2
1	Erdgeschoss	1	2	3
2	1. Stock	1	2	3
3	2. Stock	1	2	3

für die ausgelösten Aktionen zu erweitern. Auch hier kann wieder der Fahrstuhl als Einführungsbeispiel dienen. Wenn die Schüler beschreiben sollen, was der Fahrstuhl zu tun hat, wenn er einen bestimmten Zustand (Stockwerk) erreicht, werden sie bald feststellen, dass sich die Aktionen des Liftes auf „Türe öffnen“ und vielleicht einer Sprachausgabe wie „2. Obergeschoss. Spielwaren, Herrenbekleidung und Bademoden...“, so in manchen Kaufhäusern üblich, beschränken. Der Unterschied zum Getränkeautomaten wird schnell offensichtlich. Dort erwartet der Benutzer den Auswurf des gewünschten Drinks, doch dieser soll nur erfolgen, nachdem genügend Geld eingeworfen wurde. Geeignete Bei-

⁶ Zustände und Übergänge können hier sehr einfach – wie auf einem Blatt Papier – gezeichnet werden.

spielautomaten werden bei Charon mitgeliefert. Zum Üben im Unterricht und zu Hause bieten sich insbesondere Aufgaben zum Umwandeln zwischen Tabellen- und Diagrammform endlicher Automaten an.

3.4 Akzeptoren und Mustererkennung

In der letzten Lehrinheit dieser Sequenz sollen diejenigen endlichen Automaten betrachtet werden, die in der theoretischen Informatik besonders von Interesse sind, die Akzeptoren. Diese lassen sich mit Charon besonders gut realisieren und ihr Einsatz ist nach folgender Einführung auch für Schüler nicht nur legitim, sondern hoffentlich auch interessant, da diese sehr theoretischen Maschinen viele Möglichkeiten bieten. Als Motivation hierfür mag ein beliebiges Textverarbeitungsprogramm oder ein Editor, der über eine Suchfunktion verfügt, dienen. Alternativ kann man auch die in Windows integrierte Suche für Dateien und Ordner verwenden. Wird eine bestimmte Zeichenkette in die Suchmaske eingegeben, prüft das Programm, ob und wo das geforderte Muster im Text bzw. in den Dateinamen wiederzufinden ist. Man spricht dabei von „Mustererkennung“ (pattern matching) [FS97, S.6]. Diese Funktion soll im Folgenden mit endlichen Automaten realisiert werden. Dazu muss ein Automat entwickelt werden, der erkennt, ob ein eingegebenes Wort die gewünschte Zeichenkombination enthält oder nicht. Diese Erkenntnis kann im Schüler-Lehrer-Gespräch durchaus von den Schülern selbst gewonnen werden und ist aufgrund des offensichtlichen Alltagsbezuges für die Schüler sicher motivierend. Auch das Parsen könnte an dieser Stelle erwähnt werden. Fächerübergreifend bieten sich hier auch Aufgaben mit mathematischem Hintergrund an, z.B.

folgende: Gesucht ist ein Automat, der nur die Zahlen akzeptiert, die durch 4 teilbar sind. Die Zahlen werden als Binärzahlen dargestellt, das zugrundeliegende Eingabealphabet lautet also $\{0, 1\}$. Alle durch 4 teilbare Zahlen sind genau diejenigen Binärzahlen, die auf „00“ enden. Folgender endlicher Automat leistet das Gewünschte (siehe Abb. 2): Die Aufgabenstellung ist daraufhin klar und entspricht dem in Kapitel 2 angesprochenem Wortproblem. Man kann diesbezüglich unzählig viele Beispiele konstruieren, auch hier wird in Charon bereits eine große Auswahl mitgeliefert.

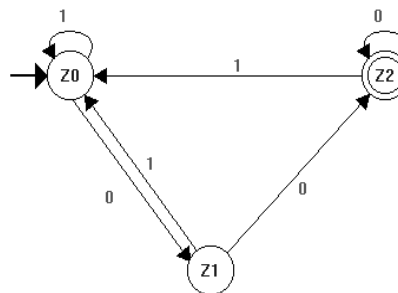


Abbildung 2: Binärzahl durch 4 teilbar?

3.5 Übungsphase, Projektmöglichkeit, Lernzielkontrolle und Ausblick

An dieser Stelle kann mindestens eine weitere Übungsstunde eingeplant werden, in der die Schüler alleine oder in Partnerarbeit weitere Aufgaben bearbeiten und neue Automaten konstruieren.

Zur Abrundung dieser Unterrichtssequenz bietet sich außerdem an, einen komplexeren Automaten in Kleingruppen zu erstellen. So kann – vielleicht als Unterrichtsprojekt aufgezogen – ein weiterer Automat konstruiert werden. Die gesamte Klasse entschließt sich für ein, ihrer Meinung nach, geeignetes und interessantes Modell und erstellt einen Grobentwurf für diesen Automaten. Verschiedene Kleingruppen zeigen sich dann für einen „Teilautomaten“ verantwortlich, dessen Realisierungen das Gesamtsystem ergeben sollte. Hier spielt natürlich die vom Lehrplan für diese Unterrichtseinheit zur Verfügung gestellte Zeit eine wesentliche Rolle. Daran anknüpfend könnten im Anschluss „hierarchische Automaten“ als Einstieg für das Thema der Modularisierung dienen. Schließlich wäre hier, egal ob mit oder ohne projektorientierte Gruppenarbeit, ein geeigneter Zeitpunkt für eine Lernzielkontrolle, etwa in Form einer Stegreifaufgabe.

Der von mir nun gebrachte Vorschlag für eine Unterrichtssequenz „Zustandsorientierte Modellierung“ lässt eine Fortführung dieser offen. So könnte in den folgenden Stunden die Umsetzung endlicher Automaten mit iterativen Programmen besprochen werden und im Anschluss die Beschreibung eines geeigneten Ablaufs durch ein Zustandsübergangsdiagramm und durch einen Algorithmus gegenübergestellt und miteinander verglichen werden. In meinem Entwurf habe ich mich bewusst auf die Darstellung von Automaten beschränkt, die den wenigen, aber klaren Regeln (festes Eingabealphabet, Determinismus, keine Übergangsbedingungen) genügen. Daher scheinen diese Automaten oftmals unflexibel und zur Simulation realer Maschinen muss didaktisch auf geeignetem Wege reduziert werden. Doch auch komplexere Maschinen können durch schrittweise Verfeinerung sehr anschaulich dargestellt werden.

4 Das Werkzeug „Charon“

Charon bietet sich sowohl als Hilfsmittel zum Erstellen von Automaten in Form von Übergangsdiagrammen an, als auch zur Simulation derjenigen, d.h. Abläufe bei der Bedienung eines Automaten können visuell nachvollzogen werden. Es besteht sowohl die Möglichkeit, ein Wort auf seine Sprachzugehörigkeit hin zu überprüfen, als auch einen Automaten mit ausgelösten Aktionen zu simulieren. Da ein derartiges Programm bislang nicht zur Verfügung stand, hatte ich beschlossen, eines für meinen Unterrichtsentwurf zu entwerfen, in der Hoffnung, dass es zum einen zum Verständnis endlicher Automaten beiträgt, zum anderen die Schüler nachhaltig motiviert, sich mit der Materie auseinander zu setzen und vielleicht auch zu Hause selbst kreativ zu werden. Außerdem kann Charon, wie im 3. Kapitel erläutert, ein

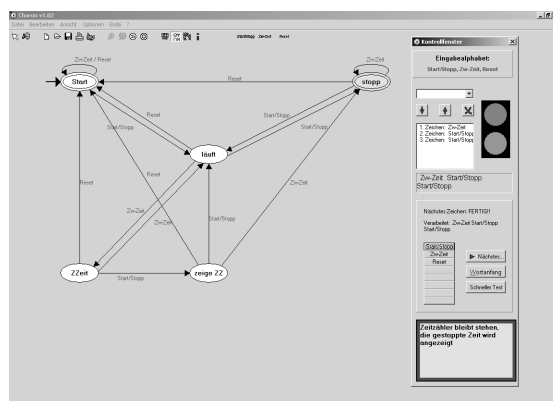


Abbildung 3: Charon

hilfreiches Werkzeug bei der Demonstration von Automaten und Einführung der zustandsorientierten Modellierung im Unterricht sein.

Beim Erstellen von Automaten zeigt sich Charon sehr flexibel. Das Hauptformular dient als Zeichenfläche. Die Übergangsdiagramme können sehr einfach per „Drag and Drop“ auf diese platziert werden. Alternativ kann man auch die Übergänge in der Zustandübergangstabelle eingeben bzw. dort editieren. Jeder Automat lässt sich auf der Zeichenfläche verändern. Zustände können markiert und bei gedrückter Maustaste beliebig verschoben werden. Die Abläufe innerhalb des Automaten können auf verschiedene Art und Weise simuliert und verfolgt werden. Möglich ist eine Bedienung „wie im Alltag“, sprich durch Drücken eines Knopfes wechselt der Automat den Zustand. Auch die Überprüfung eines Wortes kann einfach nachvollzogen werden. Sämtliche Automaten können gespeichert, ausgedruckt und in ihrer Gestaltung und Darstellung individuell verändert werden. Die Möglichkeit, das Übergangsdiagramm als Bitmap zu speichern, lässt eine Verwendung dieser Graphiken in anderen Programmen zu. Das Programm bietet eine Vielzahl weiterer Möglichkeiten – über den Einsatz im Rahmen der zustandsorientierten Modellierung hinaus –, die jedoch aus Platzgründen in einem gesonderten Artikel vorgestellt werden müssten.

Literaturverzeichnis

- [Bal96] Balzert, H.: Lehrbuch der Softwaretechnik. Band 1. Spektrum, Berlin, 1996.
- [Bau96] Baumann, R.: Didaktik der Informatik. Klett, Stuttgart, 1996.
- [Br98] Broy, M.: Informatik. Eine grundlegende Einführung Band 2. Springer, Berlin, 1998.
- [En93] Engesser, H. u.a. (Hrsg.); Schwill, A. / Claus, V. (Bearb.): Duden Informatik. Dudenverlag, Mannheim, 1993.
- [FS97] Fischer, N.; Schenk, H.: Endliche Automaten. Baustein zur Didaktik der Informatik. Staatsinstitut für Schulpädagogik und Bildungsforschung, München, 1997.
- [Hu00] Hubwieser, P.: Didaktik der Informatik: Grundlagen, Konzepte, Beispiele. Springer, Berlin, 2000.
- [Sc97] Schöning, U.: Theoretische Informatik – kurzgefaßt. Spektrum, Heidelberg, 1997.

Sonstige Quellen:

- [S1] Bayerisches Staatsministerium für Unterricht und Kultus: Rundbrief VI/7-S5502-6/75412 an die Leiterinnen und Leiter der Gymnasien in Bayern
- [S2] Hubwieser, P.: Objektorientierte Modellierung in der Sekundarstufe 1, Vortrag am Informatiklehrertag NRW an der Uni Dortmund, 23.7.2002 (<http://ddi.in.tum.de/material/vortraege/dortmund-07-02/OOM-Sek1.zip>)
- [S3] Entwurf des Lehrplan Informatik für die Gymnasien in Bayern (Stand: 10.05.02)
- [S4] Spohrer, M.: Zustandsorientierte Modellierung – Vorschlag für einen Einstieg in dieses Thema im Informatikunterricht der Mittelstufe und Charon – ein interaktives Programm zur Erstellung und Simulation endl. Automaten; Schriftliche Hausarbeit am Hans-Sachs-Gymn. Nürnberg 2001