Institut für Informatik und Praktische Mathematik
der
Christian-Albrechts-Universität zu Kiel
D-24098 Kiel

Automaten und Logiken zur Beschreibung zeitabhängiger Systeme

Thomas Wilke

Bericht Nr. 9408
Juli 1994

Dieser Bericht enthält die Dissertation des Verfassers.

Referent: Professor Dr. Wolfgang Thomas

Korreferent: Professor Dr. Willem-Paul de Roever

Inhaltsverzeichnis

Abbildungsverzeichnis

Einleitung

Zeitkritisch sind viele informationstechnische Systeme, wie z.B. die Steuerung für die Schranke an einem Bahnübergang, die Steuerung für die Ampel an der nächsten Kreuzung oder der Autopilot eines Jets. Die Informatik setzt sich mit der formalen Beschreibung — der Spezifikation — und der Überprüfung der Korrektheit — der Verifikation — solcher Systeme auseinander.

Im Mittelpunkt der vorliegenden Abhandlung stehen solche Formalismen zur Beschreibung des Verhaltens zeitabhängiger Systeme, die eine effektive Behandlung erlauben. Es werden strukturell einfache Modelle, sogenannte Automaten, und geeignete Logiken untersucht. Zum einen geht es um die Frage, wie entsprechende Automaten und Logiken aussehen und welche Eigenschaften sie auszeichnen, zum anderen darum, die Grenzen aufzuzeigen, innerhalb derer ein effektiver Umgang mit Automaten und Logiken zur Beschreibung zeitabhängiger Systeme möglich ist.

Formalismen zur Beschreibung zeitabhängigen Verhaltens benutzen als formales Modell Zustandsverläufe (‚timed state sequences', ‚observation sequences'). Dabei handelt es sich um Folgen der Form

$$(s_0, t_0)(s_1, t_1)(s_2, t_2) \ldots , \tag{0.1}$$

in denen s_i jeweils ein Zustand und $t_0, t_1, t_2, \ldots$ eine streng monoton steigende, mit Null beginnende Folge von reellen Zahlen, sogenannten beobachtbaren Zeitpunkten, ist. In der vorliegenden Arbeit werden wir uns, wie dies auch in vielen anderen Arbeiten geschieht, auf eine endliche Zustandsmenge beschränken.[1] Wir werden Folgen wie in (0.1) als reelle Zustandsverläufe bezeichnen. Eine Einschränkung auf ganzzahlige Werte t_i führt zu sogenannten ganzzahligen Zustandsverläufen, und wenn nur die Folge 0, 1, 2, ... für t_0, t_1, t_2, ... zugelassen wird, dann sprechen wir von diskreten Zustandsverläufen. Dann kann (0.1) unmittelbar mit der Folge s_0, s_1, s_2, ... identifiziert werden.

Formal ist ein zeitabhängiges Verhalten durch eine Menge von reellen Zustandsverläufen gegeben.

[1] Einen entgegengesetzten Ansatz wählt Leslie Lamport in [AL91] und [Lam93].

Historische Einführung

Die zuerst entwickelten Formalismen zur Beschreibung des Verhaltens zeitabhängiger Systeme sind geeignete logische, sogenannte zeitlogische Sprachen. So benutzen Ron Koymans, Jan Vytopil und Willem Paul de Roever schon 1983 in [KVdR83] (später in [KdR86]) eine Spezifikationssprache, die man gemäß der Klassifizierung von Alur und Henzinger ([AH91b]) als eine lineare Zeitlogik mit beschränkten Zeitoperatoren bezeichnen würde. Weitere zeitlogische Spezifikationsformalismen folgten: RTL (‚real-time logic', [JM86]), RTTL (‚real-time temporal logic', [Ost87] und [Ost90]), GCTL (‚global clock temporal logic', [Har88]), ‚metric temporal logic' ([Koy89] bzw. [Koy90]) und ECTL (‚explicit clock temporal logic', [Zho93]). Zum Teil werden diese Logiken, wie z. B. RTTL, nur über ganzzahligen Zustandsverläufen interpretiert.

Aus prinzipieller Sicht können zeitlogische Sprachen und ihre jeweilige Ausdrucksstärke danach bewertet werden, ob die entsprechende Theorie entscheidbar ist, d. h., ob die Menge aller (in allen Zustandsverläufen) gültigen Formeln eine entscheidbare Menge bildet. In ihrer wegweisenden Arbeit [AH90], in der es um die Ausdrucksstärke von Zeitlogiken geht, konnten Rajeev Alur und Thomas Henzinger nachweisen, daß die Theorien der bislang aufgezählten Logiken unentscheidbar sind. Die meisten Theorien sind sogar Π^1_1-schwierig. Einige Zeitlogiken, die über ganzzahligen Zustandsverläufen interpretiert werden und deren Theorie entscheidbar ist, wurden in den Jahren 1989 und 1990 entwickelt: TPTL (‚timed propositional temporal logic', [AH89a] und [AH89b]), MTL (‚metric temporal logic', [AH90]) und XCTL (‚explicit clock temporal logic', [HLP91]).

Die Frage nach der Entscheidbarkeit der Theorie einer unter Negation abgeschlossenen Logik ist äquivalent zu der Frage, ob eine gegebene Formel ein Modell besitzt, also zu dem sogenannten Erfüllbarkeitsproblem. Anschaulich gesprochen handelt es sich dabei um die Frage, ob eine gegebene Spezifikation in dem Sinne konsistent ist, daß sie mindestens ein gültiges Verhalten zuläßt (und damit nicht die leere Menge spezifiziert).

Im Kontrast zu den angesprochenen negativen Ergebnissen zu dem Problem der Entscheidbarkeit von zeitlogischen Theorien stehen Resultate, die das sogenannte Modellproblem betreffen. Dabei wird danach gefragt, ob ein gegebenes System $\mathcal{S}$ ein vorgegebenes Verhalten $\mathcal{V}$ zeigt. Genauer lautet die Frage: Ist die Menge der Zustandsverläufe, die $\mathcal{S}$ beschreiben, eine Teilmenge der Zustandsverläufe, die durch $\mathcal{V}$ gegeben sind? In der Regel werden für die Beschreibung des Systems und die Beschreibung des gewünschten Verhaltens unterschiedliche Formalismen verwendet.

In den Arbeiten [ACD90] und [Lew90] werden zeitabhängige Systeme durch Graphen — man könnte auch von Automaten sprechen — (‚timed graphs' bzw. ‚timed state diagrams') beschrieben und die Spezifikationen in verzweigenden zeitlogischen Sprachen (TCTL, ‚timed computation tree logic', in [ACD90]; die Logik in [Lew90] ist

namenlos) formuliert.[2] In beiden Fällen stellt sich heraus, daß das Modellproblem entscheidbar ist. Ein vergleichbares Resultat wurde in [HNSY92] für die Logiken TCTL bzw. Tμ (‚timed μ-calculus') und GCP (‚guarded-command real-time programs') erzielt.

Dem Ergebnis von Rajeev Alur, Costas Courcoubetis und David Dill ([ACD90]) liegt ein Automatenmodell (‚timed automata') zur Erkennung von Mengen reeller Zustandsverläufe zugrunde, das in [AD90] von Alur und Dill vorgestellt wurde und für das in der gleichen Arbeit als wichtigste Eigenschaft die Entscheidbarkeit des Leerheitsproblems nachgewiesen wurde. Im wesentlichen handelt es sich bei dem Modell um einen endlichen Automaten, der mit nicht anhaltbaren Stoppuhren versehen ist. Normalerweise schreiten diese mit der Systemzeit voran, sie können aber auch mit einer Transition des Automaten zurückgesetzt werden. Außerdem kann das Durchführen einer Transition von den aktuellen Uhrenständen abhängig gemacht werden. Und zwar dürfen die Uhrenstände mit natürlichen Zahlen verglichen werden. Diese Automaten aus [AD90] wollen wir Uhrenautomaten nennen, und durch sie definierbare Mengen von reellen Zustandsverläufen sollen als uhrenerkennbar (‚regular' in der Sprechweise von [AD91]) bezeichnet werden.

In [AFH91] (siehe auch [Alu91]) nutzen Alur, Tomás Feder und Henzinger dann die Entscheidbarkeit des Leerheitsproblems für Uhrenautomaten aus, um die erste Zeitlogik mit entscheidbarer Theorie zu entwickeln: MITL (‚metric interval temporal logic'). Die Sprache resultiert (syntaktisch gesehen) aus der in [AH90] vorgestellten Logik MTL (‚metric temporal logic') dadurch, daß diejenigen beschränkten Zeitoperatoren verboten werden, die exakte Messungen zeitlichen Abstands erlauben.[3] Einen alternativen, übersichtlicheren Beweis für die Entscheidbarkeit von MITL geben Alur und Henzinger in [AH92], wo sie Zweiwege-Uhrenautomaten vorstellen. Dort beweisen sie sogar die Entscheidbarkeit der Theorie von MITL$^{\mathrm{p}}$, einer Erweiterung von MITL um Vergangenheitsoperatoren.

Motivierender Hintergrund und zentrale Fragestellung

Aus der Theorie der diskreten Zustandsverläufe und Zustandsbäume sind fundamentale Zusammenhänge zwischen endlichen Automaten und monadischen Logiken zweiter Stufe bekannt, die schon in den sechziger Jahren im Zusammenhang mit Fragen der mathematischen Logik erkannt wurden: J. Richard Büchi und Calvin C. Elgot konnten nämlich zeigen, daß eine Menge von diskreten Zustandsverläufen von einem endlichen Automaten (Büchi- bzw. Rabin-Scott-Automat) genau dann erkannt wird,

[2] Dies erfordert eine differenziertere Sichtweise bei der Definition des Verhaltens eines Systems. Anstelle einer mit Zeitpunkten versehenen Zustandsfolge wie in (0.1) arbeitet man mit einem entsprechenden Zustandsbaum.

[3] Es sei hier noch einmal angemerkt, daß auch MTL zu den Logiken gehört, deren Theorie Π^1_1-schwierig ist (wenn sie über reellen oder rationalen Zustandsverläufen interpretiert wird).

wenn sie durch einen Satz in der Logik S1S (‚monadic second-order logic with one successor') definiert wird (siehe [Büc60], [Elg61] und [Büc62]). Erwähnenswert ist hier, daß Büchi- bzw. Rabin-Scott-Automaten aus Uhrenautomaten durch Weglassen der Uhren entstehen. Michael O. Rabin konnte nachweisen, daß Automaten auf unendlichen Bäumen (Rabin-Automaten) gleichwertig mit der Logik S2S (‚monadic second-order logic with two successors') sind (siehe [Rab69]). Darüberhinaus ist in beiden Fällen die Konstruktion eines „äquivalenten" Automaten aus einer gegebenen Formel bzw. die Konstruktion einer „äquivalenten" Formel aus einem gegebenen Automaten effektiv durchführbar. Die Entscheidbarkeit des Leerheitsproblems für das jeweilige Automatenmodell bringt damit auch die Entscheidbarkeit der entsprechenden Theorie mit sich, woraus die Entscheidbarkeit des Erfüllbarkeitsproblems und die Entscheidbarkeit aller Modellprobleme, bei denen Formalismen beteiligt sind, die in die jeweiligen Automaten oder in S1S bzw. S2S eingebettet werden können, folgen. Dazu gehören die folgenden linearen und verzweigenden Zeitlogiken, die in diskreten Zustandsverläufen interpretiert werden: PTL (‚propositional temporal logic', [Pnu77]), ETL (‚extended temporal logic', [Wol83]), CTL (‚computation tree logic', [CES86]), CTL* (‚full branching time logic', [EH86]).

Für reelle Zustandsverläufe war bislang keine monadische Logik bekannt, die die universelle Rolle von S1S oder S2S hätte übernehmen können. Hier setzt die vorliegende Arbeit an. Hauptanliegen ist die Entwicklung und Untersuchung monadischer Logiken,

1) die vollständig in dem Sinne sind, daß sie das Verhalten zeitabhängiger Systeme, insbesondere das Verhalten der Uhrenautomaten, exakt charakterisieren, und darüberhinaus

2) die Interpretation der Ergebnisse über reelle Zeitlogiken in dem Sinne erlauben, wie dies bei diskreten Zeitlogiken durch S1S bzw. S2S geschehen kann.

Hauptergebnisse

Um Ziel 1) zu verwirklichen, wird in dieser Arbeit ein abstraktes, parametrisiertes Automatenmodell eingeführt und parallel dazu eine Familie von korrespondierenden monadischen Logiken entwickelt, deren existentiell-monadische Sätze die durch das jeweilige Automatenmodell erkannten Mengen charakterisieren. Das abstrakte Automatenmodell ist dergestalt, daß sich bei geeigneter Parameterwahl konkrete Automatenmodelle ergeben, die mit bekannten Automaten zur Beschreibung zeitabhängiger Systeme übereinstimmen. Insbesondere gibt es einen Satz von Parametern, der genau das Modell des Uhrenautomaten trifft.

Ziel 2) wird dadurch erreicht, daß MITL$^{\mathrm{p}}$ in die zu den Uhrenautomaten korrespondierende Logik (genauer: deren existentiell-monadische Sätze) eingebettet wird, womit ein alternativer Beweis für das zentrale Ergebnis aus [AH92] gegeben wird.

Ferner gelingt der Nachweis, daß die von Zohar Manna und Amir Pnueli in [MP93] vorgestellte Logik TL_Γ eine entscheidbare Theorie besitzt.

Im folgenden sollen die Themenbereiche, mit denen sich die vorliegende Arbeit im einzelnen auseinandersetzt, kurz angerissen und jeweils die wichtigsten Ergebnisse vorgestellt werden. Am Schluß der Einleitung ist eine knapp gehaltene Gliederungsübersicht zu finden.

Eine monadische Logik mit entscheidbarem Erfüllbarkeitsproblem

Ein Hauptergebnis dieser Arbeit ist die Definition einer monadischen Logik, deren Ausdrucksstärke mit der Ausdrucksstärke der Uhrenautomaten übereinstimmt. Wir bezeichnen sie mit $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$, wobei „MSO“ für „monadic second-order“ steht, „$\exists$“ ausdrückt, daß wir uns auf existentiell-monadische Formeln beschränken und Π die zugrundeliegende Signatur bezeichnet, die mit der von S1S übereinstimmt. Das Suffix „$\overleftarrow{\mathrm{d}}$“ steht für die speziellen Prädikate, die es erlauben, zeitliche Abstände zu messen: „d“ steht für die zeitliche Abstandsfunktion, und der Linkspfeil „$\leftarrow$“ deutet an, daß Abstandsmessungen relativ zur Vergangenheit erfolgen. Wir nennen $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ auch schwache Abstandslogik, wobei der Zusatz „schwach“ darauf hindeuten soll, daß nur existentiell quantifizierte Formeln zugelassen sind und daß durch die spezielle Form der speziellen Prädikate nur eingeschränkt Abstandsmessungen durchgeführt werden können, was weiter unten erläutert wird. Durch die Äquivalenz mit den Uhrenautomaten erhalten wir die Entscheidbarkeit des Erfüllbarkeitsproblems für $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$.

Bei der Konstruktion einer monadischen Logik für reelle Zustandsverläufe (mit entscheidbarem Erfüllbarkeitsproblem) muß man behutsam vorgehen, wie schon Überlegungen in [AH90] nahelegen: Alur und Henzinger zeigen, daß die Logik, die aus S1S dadurch entsteht, daß ein zusätzliches, zweistelliges Prädikat mit der Bedeutung „der zeitliche Abstand zweier beobachtbarer Zeitpunkte ist genau Eins“ hinzugenommen wird, eine Π^1_1-schwierige Theorie hat. Dies ist sogar schon für die entsprechende Logik erster Stufe der Fall. Daraus kann man schließen, daß mögliche Kandidaten für die gesuchte Logik keinen direkten Bezug zu zeitlichen Abständen nehmen dürfen.[4]

Die Lösung, die in dieser Arbeit vorgestellt wird, liegt in einer indirekten Zeitmessung unter Verwendung von existentiell quantifizierten Mengenvariablen. Anstelle von Prädikaten erster Stufe, etwa der Form $\mathrm{d}(x,y) = 1$ oder $\mathrm{d}(x,y) > 3$, in denen d für den zeitlichen Abstand zweier Zeitpunkte steht, erlauben wir Prädikate wie

$$d(X,x) \;=\; 1\,,^{5} \tag{0.2}$$

[4] Für die Zwecke dieser Einleitung stelle man sich das Universum einer dem Zustandsverlauf (0.1) entsprechenden prädikatenlogischen Struktur als die Menge $\{t_0, t_1, t_2, \ldots\}$ der beobachtbaren Zeitpunkte vor.

in dem X für eine Mengenvariable steht und das die folgende Bedeutung hat: „Der zeitliche Abstand zwischen x und dem letzten beobachtbaren Zeitpunkt vor x, der zu der Menge X gehört, ist genau Eins." Außerdem beschränken wir uns auf existentiell-monadische Formeln, d. h. auf Formeln, die ein Präfix aus existentiell quantifizierten Mengenvariablen haben, auf das ein „Kern" ohne Quantifizierungen von Mengenvariablen folgt. Man kann zwar z. B. für $\mathrm{d}(x,y) = 1$ alternativ die Formel „X ist die Einermenge $\{x\}$ und $\mathrm{d}(X,y) = 1$" benutzen, die mit einem existentiellen Mengenquantor auskommt, wird aber über x universell quantifiziert, so kann dieser Quantor nicht als existentieller Mengenquantor wie gefordert in das Präfix der gesamten Formel geschoben werden.

Weitere monadische Logiken zur Beschreibung zeitabhängiger Systeme

Die Konzeption der Logik $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ unterliegt einem allgemeinen Prinzip, das in dieser Arbeit im einzelnen dargestellt wird. Es ermöglicht unter anderem auch die exakte Charakterisierung komplizierterer zeitabhängiger Automaten, die in jüngeren Studien behandelt werden: die sogenannten hybriden Automaten. Dabei handelt es sich um Uhrenautomaten, die zusätzliche Register benutzen, deren Inhalt sich kontinuierlich mit der fortschreitenden Systemzeit ändert. Hierbei ist die Wertefolge jedes Registers eine lineare Funktion der Zeit (mit ganzzahliger Steigung), die an den beobachtbaren Zeitpunkten einer gegebenen Zustandsfolge Sprungstellen besitzen und ihre Steigung ändern darf. Hybride Automaten werden z. B. in [KPSY93] unter dem Begriff ‚constant slope hybrid system' und in [ACHH93] unter dem Namen ‚linear hybrid automaton' geführt. Die in der vorliegenden Arbeit definierte Logik zur exakten Charakterisierung von hybriden Automaten trägt die Bezeichnung $\exists\mathrm{MSO}\Pi\int$.

In $\exists\mathrm{MSO}\Pi\int$ sind nicht nur Abstandsterme wie in (0.2) erlaubt, sondern kompliziertere Ausdrücke, die eine Integration über die Zeit erlauben. Beispielhaft soll hier die Formel

$$\int_X^x 2Y - 4Z \;>\; 3 \tag{0.3}$$

betrachtet werden. Die Bedeutung der Formel kann wie folgt erklärt werden: (0.3) ist erfüllt, wenn es einen beobachtbaren Zeitpunkt in X gibt, der kleiner als der beobachtbare Zeitpunkt $x = t_j$ ist, und wenn für den größten Zeitpunkt dieser Art, etwa t_i, gilt:

$$\sum_{i \le i' < j : i' \in Y} 2(t_{j+1} - t_j) - \sum_{i \le i' < j : i' \in Z} 4(t_{j+1} - t_j) \;>\; 3\,.$$

[5] Da es sich hier nicht um eine kommerzielle Arbeit handelt, sollten Formeln nicht abschrecken — ein Phänomen, das in Stephen W. Hawking, *Eine kurze Geschichte der Zeit: Die Suche nach der Urkraft des Universums*, Reinbek bei Hamburg: Rowohlt, 1988, beschrieben wird.

Das äußerliche Erscheinungsbild von (0.3) erinnert sehr an Formeln des ‚duration calculus' (siehe [CHR91], [HC91] und [RRH93]). Die Bedeutungen, die dem Integralzeichen in beiden Formalismen zukommen, sind vergleichbar. Man beachte jedoch, daß im ‚duration calculus' keine explizite Quantifizierung über die Integralgrenzen möglich ist.

Da das Leerheitsproblem für hybride Automaten unentscheidbar ist (siehe z. B. [KPSY93]), kommt eine effektive Behandlung im Sinne der Arbeiten von Alur, Courcoubetis, Dill, Feder, Henzinger, Lewis u. a. (siehe oben) nicht in Frage. Das jüngst in [ACH93] von Alur, Courcoubetis und Henzinger erzielte Ergebnis zeigt jedoch, daß sehr starke Einschränkungen des Modells — oder sehr schwache Erweiterungen des Modells des Uhrenautomaten — unter Beibehaltung eines entscheidbaren Leerheitsproblems möglich sind.

Dieses Ergebnis versetzt uns in die Lage, eine Teilklasse der Sprache $\exists\mathrm{MSO}\Pi\int$ auszusondern, die ausdrucksstärker als $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ ist und dennoch ein entscheidbares Erfüllbarkeitsproblem besitzt.

Eine Klasse von Mengen reeller Zustandsverläufe mit guten Eigenschaften

Die Klasse der durch Uhrenautomaten erkennbaren Mengen von reellen Zustandsverläufen ist nicht gegen Komplementbildung abgeschlossen, und das zu dem Leerheitsproblem duale Universalitätsproblem ist für Uhrenautomaten unentscheidbar. Damit kann $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ auch keine unter (semantischer) Negation abgeschlossene Logik sein, und sie kann auch keine entscheidbare Theorie besitzen. Würde man sie unter Negation abschließen, so würde man die Entscheidbarkeit des Erfüllbarkeitsproblems verlieren. Eine interessante Frage in diesem Zusammenhang ist also die Frage nach Einschränkungen von $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$, die gegen Negation abgeschlossen sind bzw. die Frage nach einem Automatenmodell unterhalb des Modells des Uhrenautomaten, das Komplementabschluß besitzt. Diese Frage wurde schon in [AH91a] aufgeworfen:

> „Let a *real-time property* be a set of *timed* state sequences (reeller Zustandsverlauf, Anmerkung des Autors). The main question we would like to see answered asks:
>
> > *Is there an agreeable notion of 'finite-state real-time property'?* The set of such properties ought to be closed under all boolean operations, have an elementary decidable emptiness problem, and be, in a suitable sense, 'maximal'."

Eine Antwort versuchen Alur und Henzinger in [AH92] selbst zu geben. Dort stellen sie Zweiwege-Uhrenautomaten vor und zeigen, daß eine Teilklasse der deterministischen Zweiwege-Uhrenautomaten, die durch eine semantische Bedingung ausgezeichnet ist, eine Klasse von Mengen von reellen Zustandsverläufen beschreibt, die

die gewünschten Eigenschaften besitzt. Wie jedoch in der vorliegenden Arbeit gezeigt wird, ist die semantische Eigenschaft, mit der Alur und Henzinger arbeiten, nicht entscheidbar, d. h., es ist nicht entscheidbar, ob ein gegebener deterministischer Zweiwege-Uhrenautomat zu der betreffenden Klasse gehört. Insofern ist der Wert ihrer Antwort zumindest fraglich.

Auch in der vorliegenden Arbeit können wir leider keine erschöpfende Antwort auf die interessante und vermutlich schwierige Frage aus [AH91a] geben. Wir versuchen, uns einer Lösung dadurch anzunähern, daß wir zeigen, daß die Logik MITL$^{\text{P}}$ um Automatenoperatoren erweitert werden kann, ohne daß dabei die Eigenschaft, eine entscheidbare Theorie zu besitzen, verloren geht. Damit zeichnen wir eine große Klasse aus, die den Anforderungen von Alur und Henzinger — abgesehen von der Maximalität — genügt.

Außerdem werfen wir die Frage auf, ob nicht die Klasse der *eindeutigen* Uhrenautomaten in die Nähe der gesuchten Automatenklasse kommt. Dazu stellen wir fest, daß alle Mengen, die durch die von Alur und Henzinger untersuchten Zweiwege-Uhrenautomaten erkannt werden, auch durch eindeutige Uhrenautomaten erkannt werden. Weiterhin schließen wir aus, daß die Klasse der durch eindeutige Uhrenautomaten erkannten Mengen mit der Klasse aller uhrenerkennbaren Mengen übereinstimmt. Darüberhinaus zeigen wir, daß gerade die Beispielmengen, die dazu dienen, den Nichtabschluß der Klasse aller uhrenerkennbaren Mengen unter Komplementbildung nachzuweisen, nicht eindeutig erkennbar sind. Die Frage nach dem Komplementabschluß für die von eindeutigen Automaten erkannten Mengen müssen wir unbeantwortet lassen. Wir zeigen aber, daß die ebenfalls semantische Bedingung der Eindeutigkeit entscheidbar ist.

Abschwächung des Modells der reellen Zustandsfolge

Die Unentscheidbarkeit des Universalitätsproblems für Uhrenautomaten ist im wesentlichen darin begründet, daß bei reellen Zustandsverläufen beliebig viele beobachtbare Zeitpunkte in ein Intervall fester Länge fallen können. Wir untersuchen deshalb, wie sich die Situation verhält, wenn man Uhrenautomaten für eine vorgegebene natürliche Zahl k ausschließlich über reellen Zustandsverläufen interpretiert, in denen zwei Zeitpunkte, zwischen denen mindestens $k-1$ beobachtbare Zeitpunkte liegen, einen zeitlichen Abstand haben, der größer als Eins ist, d. h., wir betrachten Zustandsverläufe, in denen $t_{i+k} - t_i > 1$ für jedes i gilt. Wir sprechen von Zustandsverläufen mit beschränkter Variation k. Man beachte, daß dies nicht beliebig kleine zeitliche Abstände zwischen zwei benachbarten Beobachtungszeitpunkten verbietet (sofern k größer als Eins ist).

Auf der positiven Seite verbuchen wir, daß die Klasse der uhrenerkennbaren Mengen von Zustandsverläufen einer beschränkten Variation gegen Komplementbildung abgeschlossen ist. Aus logischer Sicht bedeutet dies, daß eine Erweiterung von S1S

um direkte Abstandsvergleiche der Form $\mathrm{d}(x,y) \sim d$, in denen $\sim$ eine der Vergleichsrelationen $=, \neq, <, \leq, >, \geq$ und d eine natürliche (ganze) Zahl ist, bei Interpretation in reellen Zustandsverläufen einer beschränkten Variation zu einem entscheidbaren Erfüllbarkeitsproblem und damit auch zu einer entscheidbaren Theorie führt. Die Logik wird volle Abstandslogik genannt und erhält die Bezeichnung MSOIId. Im Kontrast dazu steht das Ergebnis, daß für die in dem hier vorgegebenen Rahmen einfachsten Erweiterungen des Uhrenautomatenmodells das Leerheitsproblem wieder unentscheidbar ist, auch wenn man sich auf reelle Zustandsverläufe einer beschränkten Variation zurückzieht.

Automatentheoretische Untersuchungen

Die herausragende Rolle, die der schwachen Abstandslogik $\exists\mathrm{MSOI}\overleftarrow{\mathrm{d}}$ aufgrund der Ergebnisse dieser Arbeit zukommt, motiviert eingehende Untersuchungen ihrer Ausdrucksstärke. Die Korrespondenz zwischen uhrenerkennbaren Mengen von reellen Zustandsverläufen und durch $\exists\mathrm{MSOI}\overleftarrow{\mathrm{d}}$-Formeln definierbaren Mengen erlaubt es, diese Untersuchungen zuerst an Uhrenautomaten durchzuführen und dann eine Übertragung auf die logische Ebene vorzunehmen. In der vorliegenden Arbeit werden zwei in dieser Hinsicht interessante Resultate erzielt: Zum einen wird mit Hilfe eines Iterationslemmas für Uhrenautomaten zum ersten Mal ein Beweis für den Nichtabschluß unter Komplementbildung und damit den Nichtabschluß von $\exists\mathrm{MSOI}\overleftarrow{\mathrm{d}}$ unter Negation gegeben. Zum anderen wird gezeigt, daß die Anzahl der Uhren in Uhrenautomaten nicht beschränkt werden kann, was auf logischer Seite zu einer unendlichen Hierarchie führt, wenn man $\exists\mathrm{MSOI}\overleftarrow{\mathrm{d}}$-Formeln nach der Anzahl der existentiell quantifizierten Mengenvariablen mißt. In diesem Zusammenhang wird auch ein aus automatentheoretischer Sicht interessantes Ergebnis erzielt: Nimmt man als Maß für die Komplexität eines Uhrenautomaten die Anzahl seiner Uhren, so kann es kein effektives, an diesem Maß orientiertes Minimierungsverfahren geben.

Das abstrakte, parametrisierte Automatenmodell

Das eingangs erwähnte abstrakte, parametrisierte Automatenmodell, das Grundlage dieser Arbeit ist, soll zum Abschluß dieser Einleitung näher erläutert werden.

Die besagten Automaten arbeiten nicht auf herkömmlichen Wörtern (Folgen von Buchstaben), sondern auf bewerteten Wörtern: Jedem Vorkommen eines Buchstabens wird ein Wert aus einer vorgegebenen Wertemenge (diese ist einer der Parameter) zugeordnet. Außerdem stehen jedem Automaten zur Verarbeitung der Bewertung eines Buchstabens Register, die Elemente eines vorgegebenen Speicherbereichs (dieser ist der zweite Parameter) aufnehmen können, zur Verfügung. Die Registerinhalte werden beim Lesen eines bewerteten Wortes in Abhängigkeit von ihrem aktuellen Inhalt und der Bewertung des aktuellen Buchstabens unter Mitwirkung sogenannter Schritt-

funktionen einer vorgegebenen Menge (dem dritten Parameter) verändert. Außerdem können sie auf ausgewählte Elemente des Speicherbereichs (dem vierten Parameter) zurückgesetzt werden. Schließlich kann eine Transition abhängig von dem Ausgang sogenannter Registertests (des fünften und letzten Parameters) gemacht werden.

Im Falle zeitabhängiger Systeme wird die Menge aller positiven reellen Zahlen als Wertemenge genommen. Dem Zustandsverlauf aus (0.1) entspricht dann das bewertete Wort

$$(s_0, t_1)(s_1, t_2 - t_1)(s_2, t_3 - t_2)(s_3, t_4 - t_3) \ldots$$

Es wird auch als Ereignisverlauf bezeichnet. Denn wir wollen eine solche Folge als zeitlich aufeinander folgende Ereignisse der Dauer t_1, $t_2 - t_1$, $t_3 - t_2$, ... und des Typs s_0, s_1, s_2, ... ansehen.

Der Speicherbereich ist in diesem Fall die Menge der reellen Zahlen, d. h., ein Register eines Automaten kann beliebige reelle Zahlen aufnehmen. Der einzige erlaubte Rücksetzwert ist Null. Im Spezialfall der Uhrenautomaten ist die Additionsfunktion als einzige Schrittfunktion erlaubt. D. h., daß beim Lesen eines bewerteten Buchstabens (s, t) — in unserer späteren Sprechweise: eines Ereignisses des Typs s und Dauer t — jedes Register um die Dauer des Ereignisses vorgestellt wird. Daraus resultiert auch die Bezeichnung „Uhr“ für ein Register in diesen Automaten. Im Spezialfall der hybriden Automaten lassen wir zu, daß auf ein Register ein ganzzahliges Vielfaches der Dauer des aktuellen Ereignisses addiert wird. Die Registertests sind jeweils von der Form $r \sim d$, wobei r für ein Register steht, $\sim$ für eine der Vergleichsrelationen $=, \neq, <, \leq, >, \geq$ und d eine ganze Zahl ist. Also kann ein Register mit jeder ganzen Zahl verglichen werden.

An dieser Stelle sind einige Bemerkungen zur bestehenden Literatur über abstrakte (parametrisierte) Automatenmodelle angebracht. Dana Scott stellt in [Sco67] einen allgemeinen Begriff von Maschine und Programm vor, dessen Universalität auch eine Modellierung des hier vorgestellten Begriffs des Σ-Automaten erlaubt. Die charakteristischen Eigenschaften unseres Automatenmodells, die sich maßgeblich in seiner logischen Charakterisierung niederschlagen, verlieren sich aber in Scotts Modell. Auch das aus [EH93] bekannte allgemeine Automatenmodell des ‚X-automaton‘ unterscheidet sich in einem entscheidenden Punkt von dem hier vorgestellten Automatenkonzept: Es verarbeitet nur herkömmliche, diskrete Wörter ohne Bewertung.

Gliederungsübersicht

Die Arbeit teilt sich in zwei Teile. Im ersten wird das abstrakte, parametrisierte Automatenmodell mit seinen wichtigsten Eigenschaften vorgestellt (Kapitel 1) und die allgemeine logische Charakterisierung erzielt (Kapitel 2). Dieser Teil ist absichtlich knapp gehalten. Er wird als die notwendige Voraussetzung für die im zweiten Teil der

Arbeit erfolgende Auseinandersetzung mit Automaten und Logiken zur Beschreibung zeitabhängiger Systeme betrachtet.

Der zweite Teil besteht aus drei Kapiteln, von denen das erste (Kapitel 3) bekannte Automatenmodelle mit Zeitabhängigkeit in das Schema einordnet, das in Kapitel 1 durch die Definition des abstrakten, parametrisierten Automatenmodells vorgegeben wurde. Das folgende Kapitel (Kapitel 4) ist automatentheoretischen Aspekten der Uhrenautomaten und ihrer Abwandlungen gewidmet, während das letzte Kapitel (Kapitel 5) die monadischen und temporalen Logiken zur Beschreibung von zeitabhängigen Systemen behandelt. Dort fügen sich in natürlicher Weise die automatentheoretischen Ergebnisse aus Kapitel 4 ein.

Dank

Dank gebührt an erster Stelle Wolfgang Thomas, der mich nun schon seit fast sechs Jahren wissenschaftlich betreut und auch bei der Entstehung der vorliegenden Abhandlung unermüdlich mit Rat und Tat beiseite stand; ich danke ihm für das Vertrauen, daß er in mich und meine Arbeit gesetzt hat. An seinem Lehrstuhl konnten Ideen in einer angenehmen Atmosphäre gedeihen. Danken möchte ich an dieser Stelle auch meinen Kollegen Andreas Potthoff, Erich Valkema, Helmut Lescow und Sebastian Seibert, die immer ein offenes Ohr für mich und meine Probleme hatten.

Durch die freundliche Bereitschaft zur Zusammenarbeit, die mir Willem Paul de Roever und die Mitarbeiter seines Lehrstuhls entgegen brachten, verstärkte sich mein Interesse für das Thema der Arbeit, und der Einstieg in die Materie wurde mir erleichtert. Nicht zuletzt verdanke ich Willem Paul de Roever die Gelegenheit zu einem für die Arbeit wichtigen Gespräch mit Professor Amir Pnueli vom Weizmann Institut, Rehovot, Israel.

Die Wissenschaftler des Kieler Instituts für Logik — Arnold Oberschelp, Klaus Potthoff und Philipp Rothmaler — taten ein übriges, um meinen Horizont auf dem Gebiet der mathematischen Logik zu erweitern.

Marion danke ich für das fleißige und sorgsame Korrekturlesen und die Aufmunterung auf Durststrecken.

Notation

Folgen – endliche oder unendliche – werden durch Unterstreichen gekennzeichnet; so steht $\underline{m}$ für $m_0, m_1, m_2, \ldots$, wobei die drei Pünktchen „…“ auch eine endliche Aufzählung zulassen. Sind $\underline{m}$ und $\underline{n}$ Folgen gleicher Länge, so ist $(\underline{m}, \underline{n})$ eine Kurzschreibweise für die Folge $(m_0, n_0), (m_1, n_1), (m_2, n_2), \ldots$ Für die Indexmenge einer Folge $\underline{m}$ wird $|\underline{m}|$ geschrieben. Da wir die von-Neumannsche Darstellung der natürlichen Zahlen wählen, d. h., daß die natürliche Zahl n für die Menge $\{0, \ldots, n-1\}$ steht, bezeichnet $|\underline{m}|$ dann auch die Länge der Folge. Das Symbol ϵ steht für die leere Folge. Wird im Text eine Variable v mit fortlaufenden Indizes versehen, so steht $\underline{v}$ für die entsprechende Variablenfolge.

Ein *Segment* einer Folge $\underline{m}$ ist ein Paar (i, j) natürlicher Zahlen, für das $i \leq j \leq |\underline{m}|$ gilt. (Insbesondere ist also im Falle unendlicher Folgen auch $j = \omega$ erlaubt.) Mit dem Segment (i, j) wird die Teilfolge $m_i, m_{i+1}, m_{i+2}, \ldots$ assoziiert, falls j unendlich ist, und $m_i, m_{i+1}, \ldots, m_{j-1}$ andernfalls. Diese Teilfolgen werden jeweils mit $\underline{m}(i, j)$ bezeichnet. Mit $\underline{m}(i, *)$ ist das Segment $m_i, m_{i+1}, m_{i+2}, \ldots$ gemeint; es handelt sich also um eine alternative Schreibweise für $\underline{m}(i, |\underline{m}|)$. Ist (i, j) ein Segment und j endlich, dann steht $\underline{m}(j, i)$ für $\underline{m}(i, j)$ in umgekehrter Reihenfolge. Die *Länge* eines Segmentes ist die Länge der mit ihm assoziierten Folge.

Für die Potenzmenge einer Menge M wird $\wp(M)$ geschrieben. Sind A und B zwei Mengen, dann steht A^B wie üblich für die Menge aller totalen Funktionen $B \to A$. Der Definitionsbereich einer Funktion f wird mit $\mathrm{dom}(f)$ bezeichnet.

Die Menge der natürlichen Zahlen, ganzen Zahlen, reellen Zahlen, positiven reellen Zahlen, nichtnegativen reellen Zahlen wird mit $\mathbf{N}$, $\mathbf{Z}$, $\mathbf{R}$, $\mathbf{P}$ bzw. $\mathbf{P}_0$ bezeichnet. Erweiterung um eine der oder beide uneigentlichen Zahlen $+\infty$ und $-\infty$ wird durch einen entsprechenden Index angezeigt. Tritt $\mathbf{N}$ als Indexmenge einer Folge auf, so schreiben wir abweichend ω für $\mathbf{N}$.

Teil I

Automaten und Logiken auf bewerteten Wörtern

Kapitel 1

Automaten auf bewerteten Wörtern

In diesem ersten Kapitel wird das abstrakte, parametrisierte Automatenmodell eingeführt, das Grundlage aller weiteren Untersuchungen der vorliegenden Arbeit ist. Wir beginnen mit der Definition der Begriffe ‚bewertetes Wort' und ‚Sprache bewerteter Wörter' (Abschnitt 1.1), fahren mit der Definition des Begriffes ‚Σ-Automat' und der von einem Σ-Automaten ‚erkannten' Sprache fort (Abschnitt 1.2). Daran anschließend studieren wir grundlegende Abschlußeigenschaften der von Σ-Automaten erkannten Sprachen (Abschnitt 1.3) und die Besonderheiten, die sich bei deterministischen Σ-Automaten ergeben (Abschnitt 1.4), und gehen schließlich auf eine alternative Definitionsmöglichkeit für den Begriff des Σ-Automaten ein (Abschnitt 1.5).

1.1 Bewertete Wörter

Es sei A eine endliche Symbolmenge, auch *Alphabet* genannt. Ihre Elemente werden als Buchstaben bezeichnet. Außerdem sei W eine beliebige, nichtleere Menge, die *Wertemenge* heißt. Beide Mengen, A und W, werden festgehalten.

Ein *diskretes Wort* über A ist eine endliche oder unendliche Folge von Buchstaben aus A. Im üblichen Sprachgebrauch werden diskrete Wörter schlicht als ‚Wörter' bezeichnet.

Ein *W-bewertetes Wort über A* oder auch kurz *W-Wort über A* ist eine endliche oder unendliche Folge von Paaren (a, w) mit $a \in A$ und $w \in W$. Ein solches Paar wird auch *W-Buchstabe* genannt. Seine erste Komponente ist der *Buchstabenname*, die zweite Komponente ist der *Buchstabenwert*. Ist $u = (\underline{a}, \underline{w})$ ein W-Wort, so wird das diskrete Wort $\underline{a}$ *Beschriftung* und die Folge $\underline{w}$ *Bewertung* von u genannt.

Eine *diskrete Sprache* über A ist eine Menge von endlichen, von ϵ verschiedenen[1]

[1]Da Strukturen mit leeren Universen Umstände bereiten, ist es im Zusammenhang mit der logi-

oder eine Menge von unendlichen diskreten Wörtern. Entsprechend ist eine *W-Sprache über A* eine Menge von endlichen, von ϵ verschiedenen oder eine Menge von unendlichen W-Wörtern über A. Damit ist jede W-Sprache über A auch eine W-Sprache über A', wenn A' ein A umfassendes Alphabet ist. Die *diskrete Projektion einer W-Sprache L* ist die Menge der Beschriftungen aller Wörter aus L.

Wir nehmen die einelementige Menge triv $= \{\perp\}$ als gegeben hin und identifizieren in natürlicher Weise die diskreten Wörter über A mit den triv-Wörtern über A. Ist L eine Sprache diskreter Wörter, so sei das *W-Urbild* von L die Menge aller W-Wörter mit Beschriftung in L.

Ist das Alphabet A einelementig, dann sprechen wir von einem *unären Alphabet* und von *unären Wörtern*. In diesem Falle ist bei W-Wörtern in der Regel nur die Bewertung von Bedeutung, weshalb wir für $(\underline{a}, \underline{w})$ einfach $\underline{w}$ schreiben. (Man beachte, daß $\underline{a}$ dann eine konstante Folge ist.)

1.2 Σ-Automaten

In diesem Abschnitt wird der klassische Begriff des endlichen Automaten zur Erkennung von Sprachen bewerteter Wörter erweitert. Dazu werden Mechanismen eingeführt, die eine Verarbeitung der Elemente des vorgegebenen Wertebereichs erlauben. Dies geschieht durch Einbeziehung von ‚Berechnungsstrukturen'.

1.2.1 Definition (Berechnungsstruktur) Eine *W-Berechnungsstruktur* Σ ist ein Quadrupel

$$\Sigma \;=\; (D, D_0, \mathcal{C}, \mathcal{F}),$$

bestehend aus einer Menge D, ihrem *Speicherbereich* mit den *Speicherelementen*, dem *Anfangsbereich* $D_0 \subseteq D$ mit den *Anfangselementen*, einer Menge $\mathcal{C}$ von Teilmengen $C \subseteq D$, sogenannten *Testprädikaten*, und einer Menge $\mathcal{F}$ von *Schrittfunktionen* $D \times W \to D$. Die Mengen D, D_0 und $\mathcal{F}$ dürfen nicht leer sein.

Sind D_0, $\mathcal{C}$ und $\mathcal{F}$ endlich, so heißt Σ *endliche Berechnungsstruktur*.

Ist $\Sigma' = (D, D_0', \mathcal{C}', \mathcal{F}')$ eine W-Berechnungsstruktur mit $D_0' \subseteq D_0$, $\mathcal{C}' \subseteq \mathcal{C}$ und $\mathcal{F}' \subseteq \mathcal{F}$, so heißt Σ' *Unterstruktur von* Σ.

Der angesprochene Mechanismus zur Verarbeitung der Bewertung eines Wortes arbeitet mit *Registern*, die Elemente des Speicherbereichs D aufnehmen und mit den Werten des Anfangsbereichs initialisiert werden können. Wir gehen im folgenden von einer endlichen Menge R von Registern aus.

Eine *Registerbelegung* ist eine Funktion $\rho\colon R \to D$.

---

schen Charakterisierung von Sprachen angebracht, auf das leere Wort zu verzichten.

Ein *Registertest* ist ein boolescher Ausdruck in atomaren Formeln der Form Cr mit $C \in \mathcal{C}$ und $r \in R$. (Insbesondere sind die logischen Konstanten, verum, in Zeichen `tt`, und falsum, in Zeichen `ff`, erlaubt.) Ein atomarer Registertest der Form Cr wird von ρ erfüllt, falls $\rho(r) \in C$ gilt. Die Erfüllbarkeitsrelation setzt sich in kanonischer Weise auf zusammengesetzte Registertests fort.

Wir betrachten zwei Arten von Registeroperationen. Zum einen können Register zurückgesetzt werden, zum anderen können sie mit Hilfe der Schrittfunktionen der Berechnungsstruktur fortgeschrieben werden.

Eine partielle Funktion $\zeta\colon R \to D_0$ heißt *Rücksetzung*. Ihre Anwendung auf eine Registerbelegung ρ liefert eine mit $\rho\zeta$ bezeichnete Registerbelegung:

$$(\rho\zeta)(r) \quad = \quad \begin{cases} \zeta(r)\,, & \text{falls } r \in \mathrm{dom}(\zeta)\,, \\ \rho(r)\,, & \text{sonst,} \end{cases} \qquad \text{für } r \in R.$$

Eine (totale!) Funktion $\theta\colon R \to \mathcal{F}$ heißt *Fortschreibung*. Sie kann nur im Zusammenhang mit einem Wert w der Wertemenge W auf eine Registerbelegung angewendet werden. Dann ergibt sich eine neue Registerbelegung, die die Bezeichnung $\rho \overset{w}{\theta}$ erhält und definiert ist durch:

$$(\rho \overset{w}{\theta})(r) \quad = \quad \theta(r)\,(\rho(r), w) \qquad \text{für } r \in R.$$

1.2.2 Definition (Σ-Automat) Ein *Σ-Automat* $\mathfrak{A}$ über A ist ein Quintupel

$$\mathfrak{A} \quad = \quad (Q, R, (s, \sigma), \Delta, F)\,, \tag{1.1}$$

das aus einer endlichen *Zustandsmenge* Q, einer endlichen Menge R von Registern, einer *Anfangskonfiguration* (s, σ) mit einem *Anfangszustand* $s \in S$ und einer *Anfangsbelegung* $\sigma\colon R \to D_0$, einer endlichen *Übergangsrelation* Δ und einer *Endzustandsmenge* $F \subseteq Q$ besteht. Die Übergangsrelation enthalte ausschließlich *Transitionen* der Form

$$(q, a, \theta, \phi, \zeta, q')\,, \tag{1.2}$$

wobei gelten soll: q und q' sind Zustände, a ist ein Buchstabe des Alphabets, θ ist eine Fortschreibung, ϕ ist ein Registertest und ζ ist eine Rücksetzung.

Im folgenden beziehen wir uns auf den Σ-Automaten $\mathfrak{A}$ aus (1.1).

Die Zustände q und q' werden auch *Quell-* bzw. *Zielzustand* der Transition (1.2) genannt und der Buchstabe a heißt *Beschriftung der Transition*.

Eine *Konfiguration* des Automaten ist ein Paar (q, ρ), bestehend aus einem Zustand q und einer Registerbelegung ρ, also ein Element von $Q \times D^R$.

Durch Lesen eines bewerteten Buchstabens (a, w) kann der Automat von einer Konfiguration (q, ρ) zu einer *Folgekonfiguration* (q', ρ') übergehen. Wir schreiben

$$(q, \rho) \xrightarrow[(a,w)]{\delta} (q', \rho')$$

für eine Transition $\delta = (q, a, \theta, \phi, \zeta, q')$, falls $\rho \overset{w}{\theta}$ den Test ϕ erfüllt und $\rho' = \rho \overset{w}{\theta} \zeta$ gilt. D. h., zuerst wird die Fortschreibung angewendet, dann wird überprüft, ob der Test erfüllt ist, und falls das der Fall ist, darf zum nächsten Zustand übergegangen werden, wobei noch die Rücksetzung nachgeschaltet wird.

Dürfen wir

$$(q_0, \rho_0) \underset{(a_0, w_0)}{\overset{\delta_0}{\longrightarrow}} (q_1, \rho_1) \underset{(a_1, w_1)}{\overset{\delta_1}{\longrightarrow}} \cdots \underset{(a_{m-2}, w_{m-2})}{\overset{\delta_{m-2}}{\longrightarrow}} (q_{m-1}, \rho_{m-1}) \underset{(a_{m-1}, w_{m-1})}{\overset{\delta_{m-1}}{\longrightarrow}} (q_m, \rho_m) \tag{1.3}$$

schreiben und ist $u = (\underline{a}, \underline{w})$ ein endliches W-Wort, so heißt (1.3) eine *durch ρ_0 und $\underline{\delta}$ gegebene Berechnung von $\mathfrak{A}$ auf u*; die Folge ist nämlich durch u, $\underline{\delta}$ und ρ_0 eindeutig bestimmt. Sie erfüllt die *Anfangsbedingung*, wenn (q_0, ρ_0) die Anfangskonfiguration von $\mathfrak{A}$ ist, sie erfüllt die *Endbedingung*, wenn q_m ein Endzustand ist, und sie heißt *akzeptierend*, falls sie sowohl die Anfangs- wie auch die Endbedingung erfüllt. Ein endliches *W-Wort u wird von $\mathfrak{A}$ akzeptiert*, wenn es eine endliche akzeptierende Berechnung von $\mathfrak{A}$ auf u gibt.

Für unendliche Wörter erfolgt die Definition sinngemäß bei Verwendung einer geänderten Endbedingung. Gilt

$$(q_0, \rho_0) \underset{(a_0, w_0)}{\overset{\delta_0}{\longrightarrow}} (q_1, \rho_1) \underset{(a_1, w_1)}{\overset{\delta_1}{\longrightarrow}} (q_2, \rho_2) \underset{(a_2, w_2)}{\overset{\delta_2}{\longrightarrow}} \cdots \tag{1.4}$$

und ist $u = (\underline{a}, \underline{w})$ ein unendliches W-Wort, so heißt die Folge (1.4) eine durch ρ_0 und $\underline{\delta}$ bestimmte Berechnung von $\mathfrak{A}$ auf u. Sie erfüllt die Anfangsbedingung, wenn (q_0, ρ_0) die Anfangskonfiguration ist, sie erfüllt die Endbedingung, wenn es unendlich viele i gibt, für die q_i ein Endzustand ist, und sie heißt akzeptierend, wenn sie sowohl die Anfangs- wie auch die Endbedingung erfüllt. Ein unendliches W-Wort u wird von $\mathfrak{A}$ akzeptiert, wenn es eine unendliche akzeptierende Berechnung von $\mathfrak{A}$ auf u gibt. (Es wird demnach die Büchische Akzeptierbedingung (siehe [Büc62]) verwendet. Weiteres dazu in Abschnitt 1.4.)

Ist von einer durch eine Transitionsfolge bestimmten Berechnung die Rede, ohne daß eine Registerbelegung für die erste Konfiguration der Berechnung erwähnt wird, dann wird dafür die Anfangsbelegung des Automaten genommen und es ist dafür gesorgt, daß der Quellzustand der ersten Transition der Anfangszustand des Automaten ist.

Die in (1.3) und (1.4) auftretenden Transitionsfolgen werden ‚konsistent' genannt. Allgemein heißt eine Transitionsfolge *konsistent*, wenn der Zielzustand jedes Gliedes der Quellzustand des nächsten Gliedes ist, sofern es dies gibt.

Ein Σ-Automat $\mathfrak{A}$, der ein unendliches W-Wort akzeptiert, akzeptiert immer auch ein endliches W-Wort. Da wir aber entweder an dem endlichen oder dem unendlichen Verhalten von $\mathfrak{A}$, nicht aber jedoch an der Kombination beider interessiert sind, legen wir fest:

1.2.3 Definition (Σ-erkennbare Sprache) Sei $\mathfrak{A}$ ein Σ-Automat. Sprechen wir von $\mathfrak{A}$ als einem Σ-*Rabin-Scott-Automaten*, so wird mit $\mathrm{L}(\mathfrak{A})$ die Menge aller von $\mathfrak{A}$ akzeptierten endlichen W-Wörter bezeichnet. Sprechen wir von $\mathfrak{A}$ als einem Σ-*Büchi-Automaten*, so wird mit $\mathrm{L}(\mathfrak{A})$ die Menge aller von $\mathfrak{A}$ akzeptierten unendlichen W-Wörter bezeichnet. Die Sprache $\mathrm{L}(\mathfrak{A})$ heißt jeweils die *von* $\mathfrak{A}$ *erkannte Sprache endlicher* bzw. *unendlicher Wörter*. Eine W-Sprache L heißt Σ-*erkennbar*, falls $L = \mathrm{L}(\mathfrak{A})$ für einen Σ-Automaten gilt.

Ist in Zukunft von einem Σ-Automaten die Rede, so wollen wir uns je nach Zusammenhang entweder einen Σ-Rabin-Scott-Automaten, einen Σ-Büchi-Automaten oder beides vorstellen.

Zwei Automaten heißen *äquivalent*, wenn sie die gleichen Sprachen erkennen. Heißt es, daß es zu einem Σ-Automaten $\mathfrak{A}$ einen äquivalenten Σ-Automaten mit einer gewissen Eigenschaft gibt, dann ist damit gemeint, daß es einen Σ-Rabin-Scott- und einen Σ-Büchi-Automaten gibt, so daß der erste die von $\mathfrak{A}$ erkannte Sprache endlicher Wörter und der zweite die von $\mathfrak{A}$ erkannte Sprache unendlicher Wörter erkennt.

1.2.4 Bemerkung 1) Ist L eine W-Sprache über A und eine W-Sprache über einem anderen Alphabet A', so ist L genau dann Σ-erkennbar über A, wenn L Σ-erkennbar über A' ist.

2) Ist Σ' eine Unterstruktur von Σ und ist L eine Σ'-erkennbare W-Sprache, so ist L auch Σ-erkennbar.

Eine weitere Bemerkung betrifft den Zusammenhang mit regulären Sprachen diskreter Wörter. Offensichtlich kann jeder diskrete endliche Automat (d. h. jeder Rabin-Scott-Automat bzw. jeder Büchi-Automat im eigentlichen Sinne) von einem Σ-Automaten simuliert werden, der einfach die Bewertung eines Eingabewortes ignoriert:

1.2.5 Bemerkung 1) Das W-Urbild jeder regulären Sprache diskreter Wörter ist Σ-erkennbar.

2) Für die triviale Berechnungsstruktur $\Sigma_{\mathrm{triv}} = (\mathrm{triv}, \mathrm{triv}, f, \{\mathrm{triv}\})$ mit $\mathrm{triv} = \{\bot\}$ und $f(\bot, \bot) = \bot$ gilt: Die Σ_{triv}-erkennbaren Sprachen korrespondieren zu den regulären Sprachen diskreter Wörter.

Wegen Punkt 1) kann jede reguläre Sprache diskreter Wörter auch als Σ-erkennbare Sprache aufgefaßt werden. Die Umkehrung von Punkt 1) — daß die diskrete Projektion einer Σ-erkennbaren Sprache immer regulär wäre — gilt nicht (vgl. 3.4.3).

Die letzte Bemerkung ist informeller Art und betrifft die Reihenfolge der Registeroperationen in einer Transition. Auf den ersten Blick mutet es vielleicht sonderbar an, daß der Test erst nach der Fortschreibung durchgeführt wird. Dies läßt sich nicht vermeiden, denn andernfalls hätte man nämlich im Fall der endlichen Wörter keine Kontrolle über den Wert des letzten Buchstabens eines bewerteten Wortes (und müßte einen Endmarker einführen).

1.3 Abschlußeigenschaften

Die Klasse der regulären Sprachen diskreter Wörter besitzt eine große Zahl von sprachtheoretischen Abschlußeigenschaften. Sie ist z. B. unter booleschen Operationen, unter Homomorphismen, unter inversen Homomorphismen, unter dem Shuffle-Produkt, der Konkatenation, dem Kleene-Stern, der Quotientenbildung, usw. abgeschlossen.

Nicht willkürlich stehen der Abschluß unter den booleschen Operationen und der Abschluß unter Homomorphismen an erster Stelle der Aufzählung, denn sie finden Entsprechungen in den elementaren logischen Operationen: Durchschnitt, Vereinigung und Komplement korrespondieren zu 'und'-Verknüpfung, 'oder'-Verknüpfung bzw. Negation, während der Abschluß unter Homomorphismen (genauer: Projektion) der existentiellen Quantifizierung von Mengenvariablen entspricht.

Es wird in diesem Abschnitt bewiesen, daß die Klasse der Σ-erkennbaren W-Sprachen unter Vereinigung, Schnitt und Projektion abgeschlossen ist. Der Komplementabschluß ist im allgemeinen nicht gegeben (vgl. 4.2.5).

1.3.1 Lemma *Die Klasse der Σ-erkennbaren W-Sprachen endlicher bzw. unendlicher Wörter ist effektiv unter der Bildung von endlichen Durchschnitten und endlichen Vereinigungen abgeschlossen.*

Unter „effektiv“ ist hier zu verstehen, daß man aus zwei gegebenen Σ-Automaten effektiv einen dritten konstruieren kann, der die Vereinigung bzw. den Durchschnitt der Sprachen erkennt, die von den beiden gegebenen Automaten erkannt werden.

Damit eine solche Behauptung Sinn ergibt, müssen D_0, $\mathcal{C}$ und $\mathcal{F}$ abzählbare Mengen sein, denn andernfalls ist die Menge aller Σ-Automaten zu mächtig, um als Eingabemenge für einen Algorithmus dienen zu können. Ist eine der Mengen nicht abzählbar, dann wollen wir in Behauptungen wie der aus dem Lemma den Zusatz über die Effektivität ignorieren.

Beweis. Seien zwei Σ-Automaten

$$\mathfrak{A} = (Q, R, (s, \sigma), \Delta, F) \quad \text{und} \quad \mathfrak{A}' = (Q', R', (s', \sigma'), \Delta', F')$$

über A gegeben.

1) Durchschnitt: Wir beschreiben einen Σ-Rabin-Scott-Automaten $\mathfrak{A}_\cap$, der $\mathrm{L}(\mathfrak{A}) \cap \mathrm{L}(\mathfrak{A}')$ erkennt, und erläutern daran anschließend, welche Modifikationen im Falle der unendlichen W-Wörter nötig werden. Der Automat $\mathfrak{A}_\cap$ simuliert gleichzeitig beide Automaten. Dazu benötigt er sowohl die Register von $\mathfrak{A}$ als auch die Register von $\mathfrak{A}'$, weshalb wir o. B. d. A. davon ausgehen, daß R und R' disjunkt sind.

Die Zustandsmenge von $\mathfrak{A}_\cap$ ist das kartesische Produkt $Q \times Q'$, die Registermenge von $\mathfrak{A}_\cap$ ist die Vereinigung $R \cup R'$. Der Automat simuliert $\mathfrak{A}$ in der ersten Komponente des Zustandsraumes und auf dem Registersatz R, während $\mathfrak{A}'$ in der zweiten Komponente und auf dem Registersatz R' simuliert wird. Deshalb wird $((s, s'), (\sigma \cup \sigma'))$ als

Anfangskonfiguration für $\mathfrak{A}_\cap$ gewählt. Und es wird in die Transitionsrelation von $\mathfrak{A}_\cap$ die Transition

$$((q,q''),a,\theta\cup\theta',\phi\wedge\phi',\zeta\cup\zeta',(q',q'''))$$

aufgenommen, wenn

$$(q,a,\theta,\phi,\zeta,q') \quad \text{und} \quad (q'',a,\theta',\phi',\zeta',q''')$$

Transitionen aus Δ bzw. Δ' sind.

Im Fall der unendlichen Wörter ist zu beachten, daß bei der Simulation beider Automaten Endzustände nicht notwendigerweise gleichzeitig in beiden Komponenten erreicht werden. Deshalb muß in die endliche Kontrolle ein zusätzlicher Mechanismus eingebaut werden. Die bekannte, für diskrete Büchi-Automaten benutzte Konstruktion läßt sich unmittelbar übertragen (siehe [Tho90a]).

2) Vereinigung: Wir gehen wie im Fall der Durchschnittsbildung vor und beschreiben zuerst einen Σ-Rabin-Scott-Automaten $\mathfrak{A}_\cup$, der $\mathrm{L}(\mathfrak{A})\cup\mathrm{L}(\mathfrak{A}')$ erkennt. Er simuliert in jeder seiner Berechnungen, startend von einem neuen Anfangszustand q_0, einen der beiden Automaten $\mathfrak{A}$ und $\mathfrak{A}'$, für den er sich in der ersten Transition nichtdeterministisch entscheidet. Da prinzipiell in beiden Anfangsbelegungen σ und σ' völlig unterschiedliche Werte möglich sind, muß der Automat mit der Berechnung auf dem gemeinsamen Registersatz $R\cup R'$ beginnen, wobei wieder o.E. angenommen wird, daß R und R' disjunkt sind. Nach der ersten Transition, mit der er sich für einen der beiden Automaten entschieden hat, wird dann auf dem Registersatz des anderen beliebig operiert, d.h., es wird eine beliebige Fortschreibung angewendet.

Der Zustandsraum von $\mathfrak{A}_\cup$ ist also $Q\cup Q'\cup\{q_0\}$, wobei o.E. vorausgesetzt wird, daß Q und Q' disjunkt sind und $q_0\notin Q\cup Q'$ gilt.

Die neue Anfangskonfiguration ist $(q_0,\sigma\cup\sigma')$. Von q_0 geht für jede Transition $(s,a,\theta,\phi,\zeta,q)$ aus Δ eine Transition

$$(q_0,a,\theta\cup\theta'_0,\phi,\zeta,q') \tag{1.5}$$

aus, wobei $\theta'_0\colon R'\to D_0$ beliebig gewählt wird. Diese Transition trifft eine Entscheidung für $\mathfrak{A}$. Analog gibt es in $\mathfrak{A}_\cup$ für jede Transition $(s',a,\theta',\phi,\zeta,q')$ aus Δ' eine Transition

$$(q_0,a,\theta_0\cup\theta',\phi,\zeta,q')\,, \tag{1.6}$$

wobei diesmal $\theta_0\colon R\to D_0$ beliebig gewählt wird. Mit dieser Transition wird eine Entscheidung für $\mathfrak{A}'$ getroffen.

Ansonsten werden die Transitionen von $\mathfrak{A}$ und $\mathfrak{A}'$ übernommen, wobei wie in (1.5) und (1.6) auf dem jeweils anderen Registersatz beliebig operiert wird. Die Endzustandsmenge ergibt sich als Vereinigung der Endzustandsmengen von $\mathfrak{A}$ und $\mathfrak{A}'$.

Im Falle der unendlichen Wörter kann offensichtlich die gleiche Konstruktion benutzt werden. ■

Eine Funktion π, die jedes W-Wort über einem Alphabet B auf ein W-Wort über A abbildet, heißt *Projektion*, wenn es eine Funktion $\pi' : B \to A$ gibt, so daß für jedes W-Wort $u = (b_0, w_0)(b_1, w_1)\ldots$ über B gilt:

$$\pi(u) \quad = \quad (\pi'(b_0), w_0)(\pi'(b_1), w_1)(\pi'(b_2), w_2)\ldots \tag{1.7}$$

Die Funktion π ist offensichtlich durch π' bestimmt und jede Funktion $\pi' : B \to A$ bestimmt durch (1.7) eine Projektion. Deshalb identifizieren wir π und π' meistens.

Wie üblich werden Projektionen auf Sprachen erweitert: Ist L eine W-Sprache über B, so ist die Menge $\{\pi(u) \mid u \in L\}$ die *Projektion von L unter π*.

1.3.2 Lemma *Die Klasse der Σ-erkennbaren W-Sprachen ist effektiv unter Projektion abgeschlossen.*

Beweis. Man ersetze die Beschriftung einer jeden Transition durch ihr Bild unter der Projektion. ■

1.4 Deterministische Automaten und Komplementierung

Zu Beginn des letzten Abschnittes wurde schon erwähnt, daß es im allgemeinen Σ-erkennbare W-Sprachen gibt, deren Komplement nicht Σ-erkennbar ist. Schränkt man sich jedoch auf deterministische Automaten ein, so hat man den Abschluß unter Komplementbildung. Damit ist die Situation mit der bei Kellerautomaten und polynomiell zeitbeschränkten Turingmaschinen vergleichbar (wenngleich noch nicht bewiesen wurde, daß NP nicht unter Komplementbildung abgeschlossen ist). Allerdings muß man im Falle der unendlichen W-Wörter — wie das auch schon von deterministischen Automaten für unendliche diskrete Wörter bekannt ist — von Büchi- zu Muller-Automaten übergehen, d. h., die Akzeptierbedingung ändern.

1.4.1 Definition (Muller-Automat) Ein Σ-*Muller-Automat* ist ein Quintupel der Form $(Q, R, (s, \sigma), \Delta, \mathfrak{F})$, dessen vier erste Komponenten wie bei einem Σ-Automaten sind und dessen letzte Komponente $\mathfrak{F}$ eine Menge von Endzustandsmengen ist, d. h. eine Teilmenge von $\wp(Q)$.

Σ-Muller-Automaten werden nur zum Erkennen von Sprachen unendlicher bewerteter Wörter benutzt. Eine unendliche Berechnung erfüllt die Endbedingung, wenn die Zustände, die unendlich oft in der zugehörigen Transitionsfolge auftreten, eine Menge aus $\mathfrak{F}$ bilden.

1.4.2 Lemma *Eine W-Sprache ist genau dann durch einen Σ-Muller-Automaten erkennbar, wenn sie von einem Σ-Büchi-Automaten erkannt wird.*

Beweis. Übliche Konstruktionen für diskrete Wörter können übernommen werden.

Ist $(Q, R, (s, \sigma), \Delta, F)$ ein Σ-Büchi-Automat, so erkennt der Σ-Muller-Automat

$$(Q, R, (s, \sigma), \Delta, \{P \subseteq Q \mid F \cap P \neq \emptyset\})$$

die gleiche Sprache.

In der anderen Richtung ist die Konstruktion etwas aufwendiger: Der Büchi-Automat simuliert den gegebenen Muller-Automaten, rät zusätzlich am Anfang die Endzustandsmenge, die für das Akzeptieren des Wortes verantwortlich sein könnte, und überprüft deterministisch während der (nichtdeterministischen) Simulation, ob die geratene Menge gerade die Menge der Zustände ist, die in dem gewählten Lauf unendlich oft auftreten. Letzteres ist mit Hilfe der Büchischen Akzeptierbedingung möglich. ■

Zur Definition des Begriffs ‚Determinismus' benutzen wir die Schreibweise

$$(q, \rho) \xrightarrow[u]{\delta},$$

die angibt, daß es eine Konfiguration (q', ρ') mit $(q, \rho) \xrightarrow[u]{\delta} (q', \rho')$ gibt:

1.4.3 Definition (Determinismus) Ein Σ-Automat ist *deterministisch*, wenn es für jede Konfiguration (q, ρ) und jeden bewerteten Buchstaben (a, w) höchstens eine Transition δ mit $(q, \rho) \xrightarrow[(a,w)]{\delta}$ gibt. Ein Σ-Automat heißt *vollständig*, wenn es jeweils mindestens eine Transition mit der angegebenen Eigenschaft gibt.

Sprechen wir im folgenden von einem deterministischen Σ-Automaten, so ist ein deterministischer Σ-Rabin-Scott-Automat bzw. ein deterministischer Σ-Muller-Automat gemeint. Eine W-Sprache heißt *deterministisch Σ-erkennbar*, wenn es einen deterministischen Σ-Automaten gibt, der sie erkennt.

Aus dem Beweis von 1.4.2 ergibt sich sofort:

1.4.4 Bemerkung Jede von einem deterministischen Σ-Büchi-Automaten erkannte Sprache ist deterministisch uhrenerkennbar.

Trivialerweise gilt, daß die Klasse der durch vollständige, deterministische Σ-Automaten erkannten Sprachen gegen Komplementbildung abgeschlossen ist: Um zum Komplement einer Sprache überzugehen, vertausche man in einem Rabin-Scott-Automaten einfach End- mit nicht-Endzuständen, während man in einem Σ-Muller-Automaten das System der Endzustandsmengen komplementiere.

Wir wollen uns klar machen, daß die Voraussetzung der Vollständigkeit nicht notwendig ist:

1.4.5 Lemma *Zu jedem deterministischen Σ-Automaten läßt sich effektiv ein äquivalenter vollständiger, deterministischer Σ-Automat konstruieren.*

Beweis. Sei $\mathfrak{A} = (Q, R, (s, \sigma), \Delta, F)$ bzw. $\mathfrak{A} = (Q, R, (s, \sigma), \Delta, \mathfrak{F})$ der gegebene deterministische Automat.

Nehmen wir an, daß es in $\mathfrak{A}$ für jeden Zustand q und jeden Buchstaben a höchstens m Transitionen mit Quellzustand q und Beschriftung a gebe. Dann soll der neue Automat außer dem Registersatz R genau $m-1$ weitere Kopien R^i mit $0 < i < m$ von R benutzen. Der ursprüngliche Registersatz werde mit R^0 bezeichnet. Die Anfangskonfiguration werde so erweitert, daß für die Kopien gleiche Anfangswerte angenommen werden. Jede Transition $(q, a, \theta, \phi, \zeta, q')$ werde in trivialer Weise auf die Kopien fortgesetzt: Die neue Transitionsrelation enthalte die Transition $(q, a, \theta', \phi, \zeta', q')$, wobei θ' auf R^i arbeite wie θ auf R und ζ' auf R^i wie ζ auf R. Dadurch wird die erkannte Sprache nicht geändert und der Automat bleibt deterministisch. Zusätzlich hat er aber noch die Eigenschaft, daß zu jedem Zeitpunkt jeder Registerinhalt m-mal zur Verfügung steht, nämlich in jedem Registersatz einmal.

Nun wird zur Zustandsmenge ein neuer Zustand q_0 hinzugefügt, eine sogenannte Senke. Und für jede Kombination aus Zustand q (von q_0 verschieden) und Buchstaben a wird die Transition

$$(q, a, \theta, \phi, \zeta, q_0)\,, \tag{1.8}$$

die im folgenden genauer spezifiziert wird, zu der neuen Transitionsrelation hinzugenommen: Für jede Transition $(q, a, \theta', \phi', \zeta', q') \in \Delta$ nehme man einen Registersatz R^i, lasse θ für jedes $i < m$ auf R^i wie θ' auf R operieren und nehme zu ϕ ein Konjunktionsglied ϕ'' hinzu, welches aus ϕ' durch Negieren und Substitution der Register aus R durch die entsprechenden Register aus R^i entstehe.

Damit ist sichergestellt, daß in jeder Konfiguration (q, ρ) mit einem Zustand $q \in Q$ zu jedem bewerteten Buchstaben genau eine Transition ausgeführt werden kann. Die zusätzliche Transition (1.8) kann nämlich nach Konstruktion genau dann schalten, wenn dies keine der anderen Transitionen aus Δ kann.

Für q_0 wird dies dadurch erreicht, daß schließlich für jeden Buchstaben a eine Transition der Form $(q_0, a, \theta, \mathtt{tt}, \emptyset, q_0)$ mit beliebiger Fortschreibung θ hinzugefügt wird.

Die Effektivität der Konstruktion ist trivialerweise gegeben. ■

1.4.6 Korollar *Die Klasse der deterministisch Σ-erkennbaren W-Sprachen endlicher bzw. unendlicher Wörter ist effektiv gegen Komplementbildung abgeschlossen.*

Daraus erhalten wir:

1.4.7 Satz *Die Klasse der deterministisch Σ-erkennbaren W-Sprachen ist effektiv gegen Anwendung der booleschen Operationen abgeschlossen.*

Beweis. Es braucht hier lediglich bemerkt zu werden, daß der im Beweis von 1.3.1 konstruierte Automat $\mathfrak{A}_\cap$ zur Erkennung des Durchschnitts zweier Σ-erkennbarer W-Sprachen deterministisch ist, sofern dies auch die beiden Ausgangsautomaten sind. (Beachte, daß dies nicht für den Automaten $\mathfrak{A}_\cup$ gilt.) ■

Ist die Klasse der Σ-erkennbaren W-Sprachen nicht unter Komplementbildung abgeschlossen, dann erkennen die deterministischen Σ-Automaten weniger Sprachen als die nichtdeterministischen. Die Potenzmengenkonstruktion, die bei endlichen diskreten Wörtern zur Determinisierung benutzt wird, führt oftmals deshalb nicht zum Erfolg, weil mit den erreichbaren Zuständen auch die jeweils erreichbaren Registerbelegungen verfolgt werden müßten und diese im Fall einer unendlichen Wertemenge nicht in endlich vielen Registern abgelegt werden können. Werden dagegen in einem gegebenen Automaten in jedem Schritt einer Berechnung unabhängig von der bislang getroffenen Auswahl an Transitionen die gleichen Fortschreibungen und die gleichen Rücksetzungen angewendet, so kann eine modifizierte Potenzmengenkonstruktion zur Determinisierung benutzt werden.

1.4.8 Definition (Registerdeterminismus) Ein Σ-Automat heißt *registerdeterministisch*, wenn es eine Äquivalenzrelation $\equiv$ auf seiner Zustandsmenge gibt, die folgende Eigenschaften aufweist:

1) Transitionen mit äquivalenten Quellzuständen und gleicher Beschriftung benutzen die gleiche Fortschreibung.

2) Sind $(q, a, \theta, \phi, \zeta, q')$ und $(q'', a, \theta, \phi', \zeta', q''')$ Transitionen mit äquivalenten Quellzuständen und gleicher Beschriftung und gibt es ein $w \in W$ und eine Registerbelegung ρ, so daß $\overset{w}{\rho}$ sowohl ϕ wie auch ϕ' erfüllt, dann gelten $\zeta = \zeta'$ und $q' \equiv q'''$.

Als direkte Konsequenz aus dieser Definition halten wir fest:

1.4.9 Bemerkung Ist $\mathfrak{A}$ ein registerdeterministischer Σ-Automat und sind

$$(q_0, \rho_0) \xrightarrow[(a_0, w_0)]{\delta_0} (q_1, \rho_1) \xrightarrow[(a_1, w_1)]{\delta_1} (q_2, \rho_2) \xrightarrow[(a_2, w_2)]{\delta_2} \dots$$

und

$$(q'_0, \sigma_0) \xrightarrow[(a_0, w_0)]{\delta'_0} (q'_1, \sigma_1) \xrightarrow[(a_1, w_1)]{\delta'_1} (q'_2, \sigma_2) \xrightarrow[(a_2, w_2)]{\delta'_2} \dots$$

Berechnungen von $\mathfrak{A}$ auf einem W-Wort $u = (\underline{a}, \underline{w})$, die beide die Anfangsbedingung erfüllen, dann gelten $\sigma_i = \rho_i$ und $q_i \equiv q'_i$ für jedes $i < |u|$.

Diese Beobachtung nutzen wir nun aus:

1.4.10 Lemma *Zu jedem registerdeterministischen Σ-Automaten kann effektiv ein äquivalenter vollständiger, deterministischer Σ-Automat konstruiert werden.*

Beweis. Wegen 1.4.5 reicht es, einen deterministischen Automaten zu konstruieren.

Wir beginnen mit einem Beweis für endliche Wörter.

Sei $\mathfrak{A} = (Q, R, (s, \sigma), \Delta, F)$ ein registerdeterministischer Σ-Rabin-Scott-Automat mit Äquivalenzrelation $\equiv$ (wie in 1.4.8). Der neue Automat $\mathfrak{A}'$ habe dann die Form

$$\mathfrak{A}' \;=\; (\wp(Q), R, (\{s\}, \sigma), \Delta', \{P \subset Q \mid P \cap F \neq \emptyset\}),$$

wobei die neue Transitionsrelation Δ' wie folgt bestimmt wird: Sei P eine Teilmenge einer Äquivalenzklasse von $\equiv$. Für jeden Buchstaben a werde eine Aufzählung $(q_0, a, \theta, \phi_0, \zeta_0, q_0'), \ldots, (q_{m-1}, a, \theta, \phi_{m-1}, \zeta_{m-1}, q_{m-1}')$ der Transitionen aus Δ mit Quellzustand q_i aus P und Beschriftung a ermittelt. Dann werde für jede nichtleere Teilmenge M von m mit $i \in M$ die Transition

$$\delta' \;=\; (P, a, \theta, \psi_M, \zeta_i, \{q_j' \mid j \in M\})$$

mit

$$\psi_M \;\triangleq\; \bigwedge_{j \in M} \phi_j \wedge \bigwedge_{j \in m \setminus M} \neg \phi_j$$

in Δ' aufgenommen. (Eine leere Indexmenge in der zweiten Konjunktion ergebe `tt`.)

Dann gilt nämlich $(P, \rho) \xrightarrow[(a,w)]{\delta'} (P', \rho')$ in $\mathfrak{A}'$ genau dann, wenn P' die Menge aller Zustände q' ist, für die es ein $q \in P$ und eine Transition δ in $\mathfrak{A}$ gibt, so daß $(q, \rho) \xrightarrow[(a,w)]{\delta} (q', \rho')$ erfüllt ist. Dies sichert, daß $\mathfrak{A}'$ zu $\mathfrak{A}$ äquivalent ist.

Die spezielle Konstruktion der Registertests ψ_M garantiert den Determinismus von $\mathfrak{A}'$, denn bei festem P und unterschiedlichen Mengen M und M' ist $\psi_M \cap \psi_{M'}$ nicht erfüllbar.

Für unendliche Wörter ist die Konstruktion weitaus aufwendiger. Wir skizzieren sie im folgenden. Aus einem Σ-Büchi-Automaten $\mathfrak{A} = (Q, R, (s, \sigma), \Delta, F)$ werde ein diskreter Büchi-Automat $\mathfrak{A}'$ konstruiert, indem jede Transition $(q, a, \theta, \phi, \zeta, q')$ zu $(q, (q, a, \theta, \phi, \zeta, q'), q')$ zusammengefaßt wird. Dann werde eine Determinisierung für $\mathfrak{A}'$ durchgeführt, wie sie z. B. in [Saf88] zu finden ist. Das Ergebnis sei $\mathfrak{A}''$. Nun werde in $\mathfrak{A}''$ jede Transition $(p, (q, a, \theta, \phi, \zeta, q'), p')$ in die Transition $((p, q), a, \theta, \phi, \zeta, (p', q'))$ umgewandelt, wodurch ein Σ-Muller-Automat, etwa $\mathfrak{A}'''$, entsteht, der äquivalent zu $\mathfrak{A}$ ist. Dieser kann wieder nichtdeterministisch sein. Der Nichtdeterminismus kann jedoch eliminiert werden, indem auf der zweiten Komponente des Zustandsraumes von $\mathfrak{A}'''$ die Konstruktion durchgeführt wird, die im ersten Teil des Beweises (für Σ-Rabin-Scott-Automaten) vorgestellt wurde. ■

1.5 Einfache Transitionen

Eine Transition eines Σ-Automaten ist insofern ein kompliziertes Gebilde, als bei ihrer Anwendung jeweils eine Fortschreibung, ein Test und eine Rücksetzung wirken. Es stellt sich die Frage, ob man nicht mit einfacheren Transitionen auskommen kann.

Man könnte sich zum Beispiel die Transition $(q, a, \theta, \phi, \zeta, q')$ als ein Folge von drei Transitionen, etwa

$$(q, a, \theta, q_0), (q_0, \phi, q_1), (q_1, \zeta, q') \tag{1.9}$$

vorstellen, wobei man q_0 und q_1 als neue Stützzustände und für jede Transition eine entsprechend modifizierte Semantik annehmen würde.

Diese Idee wollen wir in diesem Abschnitt verfolgen. Das Ziel ist es, die Äquivalenz beider Modelle, des Modells mit den normalen und des Modells mit den einfachen Transitionen, nachzuweisen.

1.5.1 Definition (einfache Transition) Eine Transition heißt *einfach*, wenn sie eine der drei folgenden Formen annimmt: (q, a, θ, q'), (q, ϕ, q') oder (q, ζ, q'), wobei q und q' für Zustände stehen, a ein Buchstabe sein soll, θ eine Fortschreibung, ϕ ein Registertest und ζ eine Rücksetzung sein mögen. Die Transitionen werden als *Buchstabentransition*, *Testtransition* und *Rücksetztransition* bezeichnet. Test- und Rücksetztransitionen heißen auch ϵ-*Transitionen*.

Ein Σ-*Automat mit einfachen Transitionen* ist ein gewöhnlicher Σ-Automat, dessen Transitionsrelation ausschließlich aus einfachen Transitionen besteht.

Der Begriffsapparat, den wir für Σ-Automaten in Abschnitt 1.2 entwickelt haben, kann auf Σ-Automaten mit einfachen Transitionen sinngemäß übertragen werden. Für die Erweiterung der dem Begriff der Berechnung zugrunde liegenden Einzelschrittrelation wollen wir dies beispielhaft vorführen. Dazu sei (q, ρ) eine Konfiguration. Dann schreiben wir

$$(q, \rho) \xrightarrow[(a,w)]{(q,a,\theta,q')} (q', \rho \overset{w}{\theta})$$

bei einer Buchstabentransition,

$$(q, \rho) \xrightarrow[\epsilon]{(q,\phi,q')} (q', \rho)\,,$$

wenn ρ den Test ϕ erfüllt, und

$$(q, \rho) \xrightarrow[\epsilon]{(q,\zeta,q')} (q', \rho\zeta)$$

bei einer Rücksetztransition.

Aus der anfänglichen Überlegung zum Aufbrechen einer Transition in einfache Transitionen (siehe 1.9) ergibt sich unmittelbar:

1.5.2 Bemerkung Zu jedem Σ-Automaten läßt sich effektiv ein äquivalenter Σ-Automat mit einfachen Transitionen konstruieren.

Die Umkehrung dieser Bemerkung, nämlich daß es zu jedem Σ-Automaten mit einfachen Transitionen einen äquivalenten „normalen“ Σ-Automaten gibt, gilt auch, ihr Beweis ist jedoch schwieriger und langwierig. Ihm ist der Rest des Abschnitts gewidmet.

Damit die Effektivität der noch vorzustellenden Konstruktion gewahrt werden kann, muß eine zusätzliche Annahme über die Berechnungsstruktur gemacht werden:

1.5.3 Definition (effektive Berechnungsstruktur) Eine Berechnungsstruktur $\Sigma = (D, D_0, \mathcal{C}, \mathcal{F})$ heißt *effektiv*, wenn D_0 und $\mathcal{C}$ sowie $\mathcal{F}$ abzählbar sind und effektiv überprüfbar ist, ob für $d_0 \in D_0$ und $C \in \mathcal{C}$ der Test „$d_0 \stackrel{?}{\in} C$“ positiv ausfällt.

Wir wollen folgendes zeigen:

1.5.4 Lemma *1) Zu jedem Σ-Automaten mit einfachen Transitionen gibt es einen äquivalenten Σ-Automaten.*

2) Ist Σ eine effektive Berechnungsstruktur, so kann die Umwandlung gemäß 1) effektiv erfolgen.

Die wesentliche Problematik liegt darin, daß in einem Σ-Automaten in jedem Schritt ein bewerteter Buchstabe gelesen wird, während bei Σ-Automaten mit einfachen Transitionen beliebig viele ϵ-Transitionen zwischen dem Lesen zweier Buchstaben ausgeführt werden können. Bei Automaten für diskrete Wörter kann man ϵ-Transitionen einfach zusammenziehen, wobei für ω-Wörter noch eine Zusatzkonstruktion nötig ist, die die Wahrung der Akzeptierbedingung sichert. Prinzipiell kann man nun bei Σ-Automaten in der gleichen Weise vorgehen, muß sich nur klar machen, daß jeweils eine Folge von Registertests und Rücksetzungen durch einen einzigen Registertest, gefolgt von einer Rücksetzung, ersetzt werden kann. Dann kann nämlich eine Buchstabentransition, gefolgt von einer Folge von Rücksetzungen und Registertests, durch eine herkömmliche Transition ersetzt werden, die zuerst die Fortschreibung anwendet und dann den Ersatztest und die Ersatzrücksetzung anschließt. Außerdem muß man sich überlegen, wie die Registertests und die Rücksetzungen, die vor dem Lesen des ersten bewerteten Buchstabens eines Wortes ausgeführt werden, behandelt werden können.

Das Redukt von Rücksetzungen und Tests. Sei R ein fester Registersatz.

Das *Redukt* einer endlichen Folge $\underline{\kappa}$ von Rücksetzungen und Registertests, das ein Paar, bestehend aus einem Registertest und einer Rücksetzung, ist, wird induktiv über die Länge von $\underline{\kappa}$ definiert.

Ist $\underline{\kappa}$ die leere Folge, so sei das Redukt von $\underline{\kappa}$ das Paar $(\mathtt{tt}, \emptyset)$. Ist $\underline{\kappa} = \underline{\kappa}'\kappa$ und (ϕ, ζ) das Redukt von $\underline{\kappa}'$, so bestimme sich das Redukt (ϕ', ζ') von $\underline{\kappa}$ in Abhängigkeit von dem Typ von κ wie folgt:

Rücksetzung: Ist κ eine Rücksetzung, so sei $\phi' = \phi$ und

$$\zeta'(r) \quad = \quad \begin{cases} \zeta(r)\,, & \text{falls } r \in \mathrm{dom}(\zeta) \setminus \mathrm{dom}(\kappa), \\ \kappa(r)\,, & \text{falls } r \in \mathrm{dom}(\kappa), \\ \text{undefiniert}, & \text{sonst}, \end{cases} \qquad \text{für } r \in R.$$

Registertest: Ist κ ein Registertest, so sei $\zeta' = \zeta$ und ϕ' die Konjunktion aus ϕ und dem Test, der dadurch entsteht, daß in ϕ jede atomare Subformel Cr mit $r \in \mathrm{dom}(\kappa)$ durch den Wahrheitswert von „$\kappa(r) \in C$“ ersetzt wird.

Die beschriebene Konjunktion kann also nur dann effektiv berechnet werden, wenn die Tests der Form „$\kappa(r) \stackrel{?}{\in} C$“ effektiv durchgeführt werden können, was unter der Voraussetzung, daß eine effektive Berechnungsstruktur Σ vorliegt, auch möglich ist.

Das *Redukt einer konsistenten Folge von ϵ-Transitionen* ist das Redukt der korrespondierenden Folge von Registertests und Rücksetzungen.

Die entscheidende Eigenschaft des Redukts einer Folge von ϵ-Transitionen wird in der folgenden Bemerkung festgehalten:

1.5.5 Bemerkung Ist

$$(q_0, \kappa_0, q_1)(q_1, \kappa_1, q_2) \ldots (q_{m-1}, \kappa_{m-1}, q_m) \tag{1.10}$$

eine Folge von ϵ-Transitionen mit Redukt (ϕ, ζ) und $\underline{\rho}$ eine Folge von $m+1$ Registerbelegungen, so gilt genau dann

$$(q_0, \rho_0) \stackrel{(q_0,\kappa_0,q_1)}{\underset{\epsilon}{\longrightarrow}} \ldots \stackrel{(q_{m-1},\kappa_{m-1},q_m)}{\underset{\epsilon}{\longrightarrow}} (q_m, \rho_m)\,,$$

wenn ρ_0 den Test ϕ erfüllt und $\rho_m = \rho_0\zeta$ gilt.

Damit kann in einer Berechnung die Folge (1.10) durch

$$(q_0, \phi, q)(q, \zeta, q_m)$$

ersetzt werden, sofern mit Hinblick auf die Büchische Akzeptierbedingung für unendliche bewertete Wörter das Auftreten von Endzuständen geeignet berücksichtigt wird, indem z. B. der Stützzustand q genau dann zu einem Endzustand gemacht wird, wenn in $\{q_1, \ldots, q_{m-1}\}$ ein Endzustand liegt.

Prinzipiell braucht die Anzahl der Redukte, die sich entlang von konsistenten ϵ-Transitionsfolgen in einem Automaten ergeben, nicht endlich zu sein. Dennoch kann

man sie durch eine entsprechend gewählte Äquivalenzrelation in endlich viele Klassen unterteilen, ohne die in ihnen enthaltene wesentliche Information zu verlieren.

Zwei Redukte (ϕ, ζ) und (ϕ', ζ') heißen *äquivalent*, falls $\zeta = \zeta'$ gilt und ϕ und ϕ' logisch äquivalent sind, wenn man in ihnen unterschiedliche atomare Registertests durch unterschiedliche aussagenlogische Variablen ersetzt. (Man könnte hier auch direkte logische Äquivalenz verlangen, was u. U. eine gröbere Einteilung, d. h. weniger Äquivalenzklassen, liefern würde. Mit Hinblick auf eine effektive Berechnung müßte dann jedoch die starke Voraussetzung, daß die von den Testprädikaten aus $\mathcal{C}$ in D erzeugte boolesche Algebra effektiv sein möge, gemacht werden.)

Zwei endliche konsistente ϵ-Transitionsfolgen $\underline{\delta}$ und $\underline{\delta}'$ heißen *äquivalent*, wenn ihre Redukte äquivalent sind, beide mit dem gleichen Zustand beginnen und auf den gleichen Zustand enden. Bei Büchi-Automaten hat man noch die Zusatzbedingung, daß in δ genau dann ein Endzustand auftrete, wenn in $\underline{\delta}'$ ein Endzustand auftritt.

Wir halten fest:

1.5.6 Lemma *Sei $\mathfrak{A}$ ein Σ-Automat mit einfachen Transitionen.*

1) *Es gibt nur endlich viele Äquivalenzklassen von konsistenten ϵ-Transitionsfolgen in $\mathfrak{A}$.*

2) *Ein Repräsentantensystem der Äquivalenzklassen aller in $\mathfrak{A}$ auftretenden endlichen konsistenten ϵ-Transitionsfolgen kann effektiv berechnet werden.*

Beweis. 1) Jede Äquivalenzklasse mit Repräsentant

$$(q_0, \kappa_0, q_1)(q_1, \kappa_1, q_2) \ldots (q_{m-1}, \kappa_{m-1}, q_m)$$

ist durch q_0, q_m, die Äquivalenzklasse des Reduktes (ϕ, ζ) von $\underline{\kappa}$ und im Falle der Büchi-Automaten die Angabe, ob $\underline{q}$ einen Endzustand enthält, bestimmt. Für die letzte Angabe und die beiden ersten kommen insgesamt $2|Q|^2$ Möglichkeiten in Frage. Ist m die Anzahl der Elemente aus D_0, die in Rücksetztransitionen von $\mathfrak{A}$ auftreten, so gibt es für ζ höchstens $(m+1)^{|R|}$ Möglichkeiten. (Beachte, daß auch der Wert „undefiniert“ auftreten kann.) Ist weiterhin n die Anzahl der in $\mathfrak{A}$ vorkommenden atomaren Registertests, so gibt es für ϕ (bezüglich aussagenlogischer Äquivalenz) höchstens 2^{2^n} Möglichkeiten. Insgesamt ergibt sich als obere Schranke für die Anzahl der Äquivalenzklassen $2|Q|^2 \times (m+1)^{|R|} \times 2^{2^n}$.

2) Bei dem Problem, die Repräsentanten der auftretenden Äquivalenzklassen zu finden, handelt es sich wegen der nach 1) gegebenen endlichen Anzahl der Äquivalenzklassen um eine Instanz des allgemeinen Wegeproblems, die mit dem Kleene-Algorithmus lösbar ist. Man beachte, daß die Berechnung von Redukten sowie der Äquivalenztest für Redukte effektiv durchführbar sind. ■

Beweis von 1.5.4. Zuerst geben wir einen Beweis für einen Σ-Rabin-Scott-Automaten $\mathfrak{A} = (Q, R, (s, \sigma), \Delta, F)$ mit einfachen Transitionen an und gehen daran anschließend auf die für Büchi-Automaten notwendigen Modifikationen ein.

Die Konstruktion besteht aus drei Schritten. Nach der Beschreibung von Schritt 1) wird eine Betrachtung zur Vorbereitung der weiteren Konstruktion angestellt.

Schritt 1) Man berechne ein Repräsentantensystem $\mathcal{R}$ der in $\mathfrak{A}$ auftretenden Äquivalenzklassen von endlichen ϵ-Transitionsfolgen (vgl. 1.5.6) und bestimme die Teilmenge $\mathcal{R}_0$ von $\mathcal{R}$, die die Transitionsfolgen mit Redukt (ϕ, ζ) enthält, die mit s beginnen und für die σ den Test ϕ erfüllt.

Falls $\underline{\delta}$ ein Repräsentant aus $\mathcal{R} \setminus \mathcal{R}_0$ ist, so beginnt keine akzeptierende Berechnung von $\mathfrak{A}$ mit einem Stück, das durch eine zu $\underline{\delta}$ äquivalente Transitionsfolge bestimmt ist. Für die Transitionsfolgen $\underline{\delta}$ aus $\mathcal{R}_0$ ist dies jedoch möglich. Ist (ϕ, ζ) das Redukt eines Elementes $\underline{\delta}$ von $\mathcal{R}_0$, so läßt sich jede Berechnung, die durch die Belegung $\sigma\zeta$ und eine Transitionsfolge, die mit dem letzten Zustand von $\underline{\delta}$ beginnt, bestimmt ist und die Endbedingung erfüllt, durch Vorschreiben der durch $\underline{\delta}$ bestimmten Berechnung zu einer akzeptierenden Berechnung auf dem gleichen Wort fortsetzen. (Beachte, daß $\underline{\delta}$ definitionsgemäß nur aus ϵ-Transitionen besteht.)

Diese Überlegung führt zu der folgenden Definition: Sei $\underline{\delta}$ ein Element von $\mathcal{R}_0$. Dann sei $L_{\underline{\delta}}$ die Sprache der Wörter, für die es eine akzeptierende Berechnung von $\mathfrak{A}$ gibt, die durch eine Transitionsfolge bestimmt ist, die mit einer zu $\underline{\delta}$ äquivalenten Transitionsfolge beginnt und dann mit einer Buchstabentransition fortfährt.

Dann gilt:

$$\mathrm{L}(\mathfrak{A}) \;=\; \bigcup_{\underline{\delta} \in \mathcal{R}_0} L_{\underline{\delta}}\,. \tag{1.11}$$

Schritt 2) Für jedes $\underline{\delta} \in \mathcal{R}_0$ konstruiere man wie folgt aus $\mathfrak{A}$ einen Automaten $\mathfrak{A}_{\underline{\delta}}$, der $L_{\underline{\delta}}$ erkennt, wobei angenommen wird, daß q der letzte Zustand von $\underline{\delta}$ sei und (ϕ, ζ) das Redukt von $\underline{\delta}$: Man wähle $(q, \sigma\zeta)$ als Anfangskonfiguration. Außerdem werde für jede Buchstabentransition (q, a, θ, q') und jede Transitionsfolge $\eta \in \mathcal{R}$ mit erstem Zustand q', letztem Zustand q'' und Redukt (ϕ', ζ') die Transition $(q', a, \theta, \phi', \zeta', q'')$ in die Transitionsrelation von $\mathfrak{A}_{\underline{\delta}}$ aufgenommen.

Schritt 3) Man konstruiere mit Hilfe von 1.3.1 den Automaten, der die Vereinigung der von den Automaten $\mathfrak{A}_{\underline{\delta}}$ erkannten Sprachen erkennt.

Die Korrektheit der Konstruktion ist durch 1.5.5 und (1.11) gesichert. Die Effektivität ist nur an einer Stelle zweifelhaft, wenn nämlich in Schritt 1) jeweils überprüft werden muß, ob σ den Test ϕ erfüllt. Dies ist aber unter der Voraussetzung, daß die Berechnungsstruktur effektiv ist, effektiv möglich.

Für Σ-Büchi-Automaten verfährt man in den Schritten 1) und 3) analog, jedoch muß man beim Zusammenziehen der Transitionsfolgen in Schritt 2) jeweils berücksichtigen, ob in $\underline{\eta}$ ein Endzustand aufgesucht wird oder nicht. Dies führt zu einer etwas

komplizierteren Konstruktion, die in einer Vergrößerung des Zustandsraumes resultiert, aber unmittelbar der Konstruktion nachempfunden werden kann, die üblicherweise zur Elimination von ϵ-Transitionen in Büchi-Automaten auf diskreten Wörtern eingesetzt wird. ■

Bei der Beschreibung konkreter Automaten wird es oftmals nützlich sein, wenn wir in einem Automaten sowohl herkömmliche wie auch einfache Transitionen benutzen können. Manchmal werden sogar *kombinierte Transitionen* von Vorteil sein. Darunter wollen wir Transitionen verstehen, die aus einer herkömmlichen Transition durch Entfernen einzelner Komponenten (der Fortschreibung, des Tests, der Rücksetzung) oder sogar aller entstehen, z. B. (q, ϕ, ζ, q') oder (q, q'). Aus 1.5.2 und 1.5.4 geht hervor, daß auch dieses liberale Automatenmodell zu dem Modell des Σ-Automaten effektiv äquivalent ist.

Kapitel 2

Monadische Logiken auf bewerteten Wörtern

Das Ziel dieses Kapitels ist eine logische Charakterisierung der Σ-erkennbaren W-Sprachen für eine beliebige Wertemenge W und eine beliebige Berechnungsstruktur Σ.

In den Arbeiten [Büc60] und [Elg61] wird gezeigt, daß jede erkennbare Sprache endlicher diskreter Wörter durch einen Satz in einer geeigneten monadischen Logik zweiter Stufe (S1S, ‚monadic second-order logic with one successor') beschrieben werden kann und daß umgekehrt jeder Satz dieser Logik eine Menge von endlichen diskreten Wörtern beschreibt, die erkennbar ist. Vergleichbare Resultate wurden für Mengen von endlichen Bäumen ([TW68], [Don70]) und endlichen Spuren ([Tho90b]) erzielt. Für die erkennbaren Mengen der jeweiligen unendlichen Objekte (ω-Wörter, unendliche Bäume und unendliche Spuren) sind entsprechende Ergebnisse bekannt ([Büc62], [Rab69], [EM93]).

Im Fall der Σ-erkennbaren W-Mengen kann man nicht auf eine Beschreibung durch eine volle Logik hoffen, da eine solche mit einer Formel auch ihr Negat enthielte und damit eine unter Komplementbildung abgeschlossene Klasse von Sprachen beschriebe. Jedoch ist die Klasse der Σ-erkennbaren W-Sprachen im allgemeinen nicht unter Komplementbildung abgeschlossen (vgl. 4.2.5). Auch die Klasse der erkennbaren Bildsprachen, die in [GR92] eingeführt wurde, ist nicht unter Komplementbildung abgeschlossen. Dennoch läßt sie eine stabile logische Charakterisierung zu: In [GRST94] wird gezeigt, daß eine Bildsprache genau dann erkennbar ist, wenn sie durch einen existentiell monadisch quantifizierten Satz definiert wird. Ein ähnliches Ergebnis werden wir für Σ-erkennbare W-Sprachen erzielen.

Zuerst wiederholen wir die klassischen Definitionen und Ergebnisse für diskrete Wörter (Abschnitt 2.1). Dann erläutern wir den allgemeinen logischen Rahmen, in dem wir arbeiten werden (Abschnitt 2.2), und daran anschließend formulieren wir das Hauptergebnis dieses Kapitels, den sogenannten Charakterisierungssatz (Abschnitt 2.3). Der Beweis des Satzes umfaßt die drei nächsten Abschnitte, an die

sich der Beweis einer Verallgemeinerung des Charaktersierungssatzes anschließt (Abschnitt 2.7). Zum Schluß geht es um die Frage, unter welchen Umständen der Leerheitstest für Σ-Automaten entscheidbar und die Projektionen jeder Σ-erkennbaren Sprache regulär ist (Abschnitt 2.8).

2.1 Monadische Logiken zweiter Stufe und Büchis Ergebnisse

Im allgemeinen ist die Syntax einer monadischen Logik zweiter Stufe durch eine herkömmliche Signatur erster Stufe vorgegeben, die Symbole für Funktionen (einschließlich Konstanten) und Relationen auf dem Universum enthält. Die Terme der Sprache sind die gleichen wie die der Sprache der ersten Stufe in der gegebenen Signatur. Die Formeln werden in der üblichen Weise aus atomaren Formeln induktiv durch die Anwendung boolescher Operationen und die Quantifizierung über Element- und Mengenvariablen (!) gebildet, wobei außer den atomaren Formeln der Sprache der ersten Stufe für jede Mengenvariable X und jeden Term t der Ausdruck Xt zugelassen wird und die Bedeutung „t ist Element von X“ erhält. Standardmäßig werden Elementvariablen und (Element-)Terme mit Kleinbuchstaben und Mengenvariablen mit Großbuchstaben bezeichnet.

Eine Formel der Gestalt $\exists X_0 \ldots \exists X_{m-1} \phi_0$, bei der ϕ_0 keine Mengenquantoren enthält, wird üblicherweise *existentiell-monadische Formel* genannt. Demgegenüber sprechen wir von einer beliebigen Formel der monadischen Logik zweiter Stufe in der gegebenen Signatur als einer *monadischen Formel*.

2.1.1 Definition Sei Ω eine Signatur (wie oben beschrieben). Die Menge der monadischen Ω-Formeln wird mit MSOΩ bezeichnet, während $\exists$MSOΩ für die existentiell-monadischen Ω-Formeln steht.

Um über diskrete Wörter über dem Alphabet A sprechen zu können, betrachten wir die Signatur Π erster Stufe, die aus einem zweistelligen Relationssymbol $<$ (in Infixschreibweise) und einem einstelligen *Buchstabenprädikat* P_a für jeden Buchstaben a aus A besteht. Wird die Signatur auf der Grundlage eines mit einem von A verschiedenen Buchstaben bezeichneten Alphabets gebildet, so tritt ein entsprechendes Argument, z. B. wie in $\Pi(B)$, hinzu.

Aus jedem diskreten Wort $u = \underline{a}$ über A gewinnt man in natürlicher Weise eine Π-Struktur, die mit u^Π bezeichnet wird: Die Menge $|u|$ ist das Universum von u^Π, das Symbol $<$ wird durch die natürliche Kleiner-als-Relation auf dem Universum und P_a durch $\{i < |u| \mid a_i = a\}$ interpretiert. Man stelle sich also das Universum als die Indexmenge des Wortes $\underline{a}$ — im eigentlichen Sinne als Folge betrachtet — vor. Der Vorstellung, daß es sich bei einem Wort um ein geometrisches Objekt (räumliche

Aneinanderreihung von Buchstaben) handelt, entspringt die Bezeichnung *Stelle* für ein Element des Universums.

Jede dieser Π-Strukturen wird als Π-*Wortstruktur* oder *diskrete Wortstruktur* bezeichnet.

Grundsätzlich werden Formeln aus MSOΠ nur in Π-Wortstrukturen interpretiert und zwar getrennt nach endlichen bzw. unendlichen Wörtern, wie wir es von den Automaten her kennen: Die Menge aller endlichen Π-Wortmodelle eines Satzes ϕ aus MSOΠ wird mit $\mathrm{Mod}_+(\phi)$, die Menge aller unendlichen Π-Wortmodelle mit $\mathrm{Mod}_\omega(\phi)$ bezeichnet. Falls nur $\mathrm{Mod}(\phi)$ geschrieben wird, so ist je nach Zusammenhang $\mathrm{Mod}_+(\phi)$, $\mathrm{Mod}_\omega(\phi)$ oder beides gemeint. (Vgl. die Festlegung im Anschluß an 1.2.3.)

2.1.2 Definition (definierbare Sprache) Eine Sprache L von diskreten Wörtern heißt *monadisch* Π-*definierbar*, falls es einen monadischen Π-Satz ϕ gibt, für den $\mathrm{Mod}(\phi) = L$ gilt. Sie heißt *existentiell-monadisch* Π-*definierbar*, wenn ϕ existentiell-monadisch gewählt werden kann.

Der fundamentale Zusammenhang zwischen definierbaren und erkennbaren Sprachen diskreter Wörter ist in dem folgenden Satz zusammengefaßt:

2.1.3 Satz ([Büc60][Elg61][Büc62]) *1) Sei L eine Sprache diskreter Wörter. Dann sind die drei folgenden Aussagen äquivalent.*

- *(a) L ist monadisch* Π-*definierbar.*
- *(b) L ist existentiell-monadisch* Π-*definierbar.*
- *(c) L wird durch einen Rabin-Scott- bzw. durch einen Büchi-Automaten erkannt.*

Die jeweiligen Umwandlungen (von Formeln in Formeln, von Formeln in Automaten und zurück) können effektiv erfolgen.

2) Es ist entscheidbar, ob ein gegebener monadischer Π-*Satz ein endliches* Π-*Wortmodell hat und ob er ein unendliches* Π-*Wortmodell hat.*

Da Π-Formeln nur in Π-Wortstrukturen interpretiert werden, definieren wir (logische) Äquivalenz auch nur bezüglich Π-Wortstrukturen: Zwei monadische Π-Formeln heißen *äquivalent*, wenn jede Interpretation in einer Π-Wortstruktur genau dann die eine Formel erfüllt, wenn sie die andere erfüllt. Und ein monadischer Π-Satz ϕ heißt *äquivalent* zu einem Automaten $\mathfrak{A}$, wenn $\mathrm{Mod}(\phi) = \mathrm{L}(\mathfrak{A})$ gilt.

Mit anderen Worten besagt 2.1.3 also, daß man zu jedem diskreten Automaten effektiv einen äquivalenten (existentiell-)monadischen Π-Satz konstruieren kann und umgekehrt.

2.2 Erweiterte monadische Logiken zweiter Stufe

Für die Beschreibung von erkennbaren Sprachen bewerteter Wörter reichen die anfangs des letzten Abschnitts vorgestellten monadischen Logiken zweiter Stufe nicht aus. Sie müssen um Relationen, die zwischen Elementen und Mengen des Universums bestehen können, erweitert werden. Formal lassen wir deshalb in einer *erweiterten Signatur* außer Symbolen für Funktionen und Relationen auf dem Universum eine dritte Art von Symbolen zu, die *erweiterte Symbole* genannt werden. Die Signaturfunktion ordnet jedem der erweiterten Symbole einen Typ (m, n) zu und jedes erweiterte Symbol vom Typ (m, n) wird in einer Struktur mit Universum M durch eine Teilmenge von $\wp(M)^m \times M^n$ interpretiert, d. h., ein solches Symbol steht für eine *erweiterte Relation* mit m Mengen- und n Elementargumenten. Man kann also jedes n-stellige Relationssymbol auch als erweitertes Symbol des Typs $(0, n)$ auffassen.

Zu den Regeln für die Bildung der monadischen Formeln tritt eine weitere: Falls E ein erweitertes Symbol des Typs (m, n) ist, $X_0, \ldots, X_{m-1}$ Mengenvariablen sind und $t_0, \ldots, t_{n-1}$ Terme, dann ist auch

$$E(X_0, \ldots, X_{m-1}, t_0, \ldots, t_{n-1})$$

eine atomare Formel, eine sogenannte *erweitert-atomare* Formel.

Die Bezeichnungen MSOΩ und $\exists$MSOΩ aus 2.1.1 werden für erweiterte Signaturen beibehalten.

2.3 Die Sprache MSO$\Pi\Sigma$ und der Charakterisierungssatz

Sei Σ wiederum eine W-Berechnungsstruktur.

Für die Beschreibung von Σ-erkennbaren Sprachen wird die Signatur Π (aus Abschnitt 2.1) um erweiterte Symbole ergänzt.

2.3.1 Definition (Signatur $\Pi\Sigma$) Es ist $\Pi\Sigma$ die erweiterte Signatur, die außer den Symbolen aus Π für jedes $l \geq 0$, jede mit den Teilmengen von l indizierte Familie $F = \{f_P\}_{P \subseteq l}$ von Schrittfunktionen aus $\mathcal{F}$, jede mit den Teilmengen von l indizierte Familie $H = \{h_P\}_{P \subseteq l}$ von Anfangselementen und jedes Testprädikat C aus $\mathcal{C}$ ein erweitertes Symbol $\mathrm{E}_{C,F,H}$ des Typs $(l+1, 1)$ enthält.

Im folgenden wird beschrieben, wie man aus einem W-Wort $u = (\underline{a}, \underline{w})$ eine $\Pi\Sigma$-Struktur $u^{\Pi\Sigma}$ gewinnt.

Zuerst werden das Universum und die Interpretation der Symbole, die auch in der Signatur Π enthalten sind, derart gewählt, daß das Redukt $u^{\Pi\Sigma} \restriction \Pi$ mit der aus Abschnitt 2.1 bekannten Wortstruktur $\underline{a}^{\Pi}$ übereinstimmt. Mit anderen Worten, das

Universum von $u^{\Pi\Sigma}$ ist $|u|$, das Symbol $<$ steht für die natürliche Kleiner-als-Relation auf dem Universum und es ist $u^{\Pi\Sigma} = \{i < |u| \mid a_i = a\}$.

Für die Interpretation eines erweiterten Symbols $\mathrm{E}_{C,F,H}$ des Typs $(l+1,0)$ in $\Pi\Sigma$ definieren wir zunächst Hilfsfunktionen.

Ist $N \subseteq \mathbf{N}$ eine Menge von natürlichen Zahlen und $m \in \mathbf{N}$, so ist der *Vorgänger von m in N* die Zahl $\max(\{m' \in N \mid m' < m\} \cup \{-1\})$, die mit $\mathrm{vg}(m,N)$ bezeichnet wird.

Sind $F = \{f_P\}_{P\subseteq l}$ und $H = \{h_P\}_{P\subseteq l}$ wie in 2.3.1, so sei

$$\mathrm{E}^u_{F,H}\colon \wp(|u|)^m \times \{(m,n) \in (\{-1\} \cup |u|)^2 \mid m \le n\} \to D$$

die Funktion, die durch Induktion über die Differenz $n-m$ wie folgt definiert ist (wobei $\underline{M}$ für eine l-gliedrige Folge von Teilmengen von $|u|$ stehe):

$$\begin{aligned} \mathrm{E}^u_{F,H}(\underline{M},m,m) &= h_{\{i \mid m \in M_i\}}\,, \\ \mathrm{E}^u_{F,H}(\underline{M},m,n+1) &= f_{\{i \mid n+1 \in M_i\}}(\mathrm{E}^u_{F,H}(\underline{M},m,n), w_{n+1})\,. \end{aligned}$$

(Beachte, daß dadurch immer $\mathrm{E}^u_{F,H}(\underline{M},-1,-1) = h_\emptyset$ gilt.)

D. h., der Wert $\mathrm{E}^u_{F,H}(\underline{M},m,n)$ kann auf die folgende Weise errechnet werden: Man bestimmt zuerst, zu welchen Mengen aus $\underline{M}$ die Stelle m gehört. Daraus ergibt sich dann vermöge H das Speicherelement, mit dem die Berechnung gestartet wird. Dann bestimmt man nacheinander für $i = m+1, \ldots, n$ jeweils die Mengen von $\underline{M}$, zu denen i gehört. Dadurch sind vermöge F Fortschreibungen vorgegeben. Diese wendet man nacheinander auf den Startwert an, wobei jeweils als zweites Argument der Reihe nach einer der Werte $w_m, \ldots, w_n$ genommen wird.

Das erweiterte Symbol $\mathrm{E}_{C,F,H}$ werde in $u^{\Pi\Sigma}$ durch die Menge aller Tripel $(N,\underline{M},m)$ interpretiert, für die $\mathrm{E}^u_{F,H}(\underline{M},\mathrm{vg}(m,N),m) \in C$ gilt. Damit gehört $(N,\underline{M},m)$ genau dann zu der Interpretation von $\mathrm{E}_{C,F,H}$ in $u^{\Pi\Sigma}$, wenn das Speicherelement, das man auf die oben beschriebene Weise auf dem Wege vom Vorgänger von m in N zu m erhält, zu C gehört.

Die Beschreibung von $u^{\Pi\Sigma}$ ist somit vollständig.

Die aus W-Wörtern gewonnenen $\Pi\Sigma$-Strukturen werden als $\Pi\Sigma$-*Wortstrukturen* bezeichnet.

Genauso wie in Abschnitt 2.1 festgelegt wurde, daß Π-Formeln nur in Π-Wortstrukturen interpretiert werden, so sollen $\Pi\Sigma$-Formeln nur in $\Pi\Sigma$-Wortstrukturen interpretiert werden. Demgemäß mögen auch die Begriffe *monadisch $\Pi\Sigma$-definierbar*, *existentiell-monadisch $\Pi\Sigma$-definierbar* und der Begriff der ‚Äquivalenz' sowie die Bezeichnung $\mathrm{Mod}(\phi)$ verstanden werden.

Das fundamentale Ergebnis dieser Arbeit ist eine Beschreibung der Σ-erkennbaren Sprachen durch $\Pi\Sigma$-Formeln:

2.3.2 Charakterisierungssatz *Sei L eine W-Sprache. Dann sind die folgenden Aussagen äquivalent:*

1) L ist existentiell-monadisch $\Pi\Sigma$-definierbar.

2) L ist Σ-erkennbar.

Die jeweiligen Umwandlungen (von Formeln in Automaten und Automaten in Formeln) können effektiv erfolgen.

Der Beweis dieses Satzes ist umfangreich. Der nächste Abschnitt beschäftigt sich mit der Implikation von 2) nach 1), während sich die darauffolgenden Abschnitte 2.5 und 2.6 mit der Implikation von 1) nach 2) auseinandersetzen.

2.4 Logische Beschreibung von akzeptierenden Berechnungen

Wir zeigen die Implikation von 2) nach 1) in 2.3.2, dem Charakterisierungssatz.

2.4.1 Lemma *Zu jedem Σ-Automaten läßt sich effektiv eine äquivalente existentiell-monadische $\Pi\Sigma$-Formel konstruieren.*

Beweis. Wir geben zuerst einen Beweis für W-Sprachen unendlicher Wörter. Anschließend werden wir die für den Fall der endlichen Wörter notwendigen Modifikationen erläutern.

Sei $\mathfrak{A} = (Q, R, (q, \rho), \Delta, F)$ ein Σ-Büchi-Automat mit s Registern $r_0, \ldots, r_{s-1}$ und t Transitionen $\eta_0, \ldots, \eta_{t-1}$ der jeweiligen Form $\eta_j = (q_j, a_j, \theta_j, \phi_j, \zeta_j, q'_j)$.

Wir konstruieren nun eine existentiell-monadische $\Pi\Sigma$-Formel akz mit der Eigenschaft $\mathrm{Mod}_\omega(\mathsf{akz}) = \mathrm{L}(\mathfrak{A})$.

Dazu wird der folgende Ansatz gewählt:

$$\mathsf{akz} \triangleq \exists X_0 \ldots \exists X_{s-1} \exists Y_0 \ldots \exists Y_{t-1} \mathsf{akzBer}\,.$$

Die Formel akzBer soll eine Formel ohne Mengenquantoren werden. Sie soll genau dann in einer Interpretation $(u^{\Pi\Sigma}, \nu)$ mit einer Variablenbelegung ν erfüllt sein, wenn es eine durch eine Transitionsfolge $\underline{\delta}$ bestimmte akzeptierende Berechnung von $\mathfrak{A}$ auf u gibt, so daß für jedes $k \in \mathbf{N}$, jedes $j < t$ und jedes $i < s$ gilt:

$$k \in \nu(Y_j) \quad \text{gdw.} \quad \delta_k = \eta_j\,, \tag{2.1}$$

$$k \in \nu(X_i) \quad \text{gdw.} \quad \delta_k \text{ setzt } r_i \text{ zurück}\,. \tag{2.2}$$

Die Formel akzBer werden wir als eine Konjunktion der Bedingungen, die an eine akzeptierende Berechnung gestellt werden, erhalten. Als Abkürzung werden verwendet:

$$\mathrm{suc}(x, y) \triangleq x < y \wedge \forall z(x < z \rightarrow \neg z < y)\,,$$
$$0 = x \triangleq \neg\exists y(y < x)\,.$$

Zuerst wollen wir die notwendigen Nebenbedingungen, die sich direkt aus der Forderung an die Art der Kodierung ergeben, formulieren.

Die Mengen Y_j mit $j < t$ bilden eine Partition des Universums:

$$\mathsf{Partition} \triangleq \forall x(\bigvee_{j<t} Y_j x \wedge \bigwedge_{j<t}(Y_j x \to \bigwedge_{j'<j} \neg Y_{j'} x)) .$$

Außerdem ergeben sich die Belegungen der Mengen X_i direkt aus den Belegungen der Mengen Y_j:

$$\mathsf{RückPkt} \triangleq \bigwedge_{i<s} \forall x(X_i x \leftrightarrow \bigvee_{j<t:\ \eta_j \text{ setzt } r_i \text{ zurück}} Y_j x) .$$

Damit die kodierte Berechnung akzeptierend ist, müssen Anfangs- und Endbedingung erfüllt sein:

$$\mathsf{AnfBed} \triangleq \bigvee_{j:q_j=q} \forall x(0 = x \to Y_j x) ,$$

$$\mathsf{EndBed} \triangleq \forall x \exists y(x < y \wedge \bigvee_{j:q_j \in F} Y_j y) .$$

Die Transitionsfolge muß konsistent sein und ihre Beschriftung muß mit der Beschriftung des gegebenen Wortes übereinstimmen:

$$\mathsf{konsTransFolge} \triangleq \forall x \forall y(\mathrm{suc}(x,y) \to \bigvee_{j,j':q'_j=q_{j'}} (Y_j x \wedge Y_{j'} y)) ,$$

$$\mathsf{Beschriftung} \triangleq \forall x \bigwedge_{a \in A} (\mathrm{P}_a x \to \bigvee_{j:a_j=a} Y_j x) .$$

Schließlich müssen alle Registertests erfüllt sein. Um dies zu beschreiben, benutzen wir die erweiterten Prädikate aus $\Pi\Sigma$. Wir betrachten ein Register r_i und definieren $F^i = \{f^i_P\}_{P \subseteq t}$ und $H^i = \{h^i_P\}_{P \subseteq t}$ wie folgt:

$$f^i_P = \begin{cases} \theta_j(r_i) , & \text{falls } P = \{j\}, \\ \text{beliebig}, & \text{sonst}, \end{cases} \tag{2.3}$$

$$h^i_P = \begin{cases} \zeta_j(r_i) , & \text{falls } P = \{j\} \text{ und } r_i \in \mathrm{dom}(\zeta_j), \\ \rho(r_i) , & \text{falls } P = \emptyset, \\ \text{beliebig}, & \text{sonst}. \end{cases} \tag{2.4}$$

Dann hat

$$\mathrm{E}^u_{F^i,H^i}(\nu(X_i), \nu(Y_0), \ldots, \nu(Y_{t-1}), \mathrm{vg}(\nu(x), \nu(X_i)), \nu(x))$$

genau den Wert des Registers r_i, den dieses bei dem durch $\underline{\delta}$ bestimmten Lauf von $\mathfrak{A}$ auf u nach der Verarbeitung der ersten $x+1$ Ereignisse enthält (sofern (2.1) und (2.2) erfüllt sind). Deshalb setzen wir für jedes Testprädikat C:

$$\psi_i^C \triangleq \mathrm{E}_{C,F^i,H^i}(X_i, \underline{Y}, x).$$

Außerdem entstehe die Formel ϕ_j' aus ϕ_j, indem jede atomare Formel Cr_i durch ψ_i^C ersetzt werde. Dann wird die Erfüllung der Registertests durch die Formel

$$\mathsf{Test} \triangleq \forall x \bigwedge_{j<t} (Y_j x \to \phi_j')$$

ausgedrückt.

Damit sind alle Bedingungen an die Kodierung einer akzeptierenden Berechnung formuliert und akzBer ergibt sich als Konjuktion dieser. Die gesamte Formel ist Abbildung 2.1 zu entnehmen.

Im Fall der endlichen bewerteten Wörter bzw. Rabin-Scott-Automaten kann die Formel akz prinzipiell übernommen werden. Es muß lediglich der anderen Endbedingung Rechnung getragen werden.

Offensichtlich kann akz aus $\mathfrak{A}$ effektiv bestimmt werden. ■

Für die spätere Verwendung wollen wir noch eine wichtige Eigenschaft der Formel akz festhalten. Dazu sei der *E-Rang* einer $\Pi\Sigma$-Formel ϕ die Anzahl der in ϕ auftretenden erweitert-atomaren $\Pi\Sigma$-Formeln, wobei ein Unterschied in dem benutzten Testprädikat nicht gerechnet wird. (Die Formel $\mathrm{E}_{C,F,H}(X,Y,x) \wedge \mathrm{E}_{C',F,H}(X,Y,x)$ hat z. B. E-Rang Eins.) Der E-Rang einer $\Pi\Sigma$-definierbaren W-Sprache sei der minimale E-Rang aller existentiell-monadischen $\Pi\Sigma$-Formeln, die L definieren.

2.4.2 Bemerkung Ist L eine W-Sprache, die von einem Σ-Automaten mit s Registern erkannt wird, so ist ihr E-Rang kleiner oder gleich s.

2.5 Interpretation in diskreten Strukturen

Für den Beweis der Implikation von 1) nach 2) im Charakterisierungssatz (siehe 2.3.2) benutzen wir den Büchischen Satz. Eine existentiell-monadische $\Pi\Sigma$-Formel ϕ überführen wir in eine in gewissem Sinne äquivalente monadische Π-Formel ϕ^* (die über diskrete Wörter spricht), benutzen dann 2.1.3, um einen zu ϕ^* äquivalenten diskreten Automaten $\mathfrak{A}$ zu konstruieren, und wandeln schließlich den diskreten Automaten $\mathfrak{A}$ in einen Automaten $\hat{\mathfrak{A}}$ um, der auf bewerteten Wörtern arbeitet und zu der ursprünglichen Formel ϕ äquivalent ist. In diesem Abschnitt geht es um den Schritt von ϕ zu ϕ^*, während im nächsten Abschnitt die Konstruktion von $\hat{\mathfrak{A}}$ aus $\mathfrak{A}$ vorgestellt wird.

Es sei $X_0, X_1, \ldots$ ein unendlicher Vorrat an unterschiedlichen Mengenvariablen und x eine feste Elementvariable. Für jede natürliche Zahl m sei $\mathcal{W}_m$ die Menge $\{X_0, \ldots, X_{m-1}\}$.

$$
\begin{aligned}
\mathsf{akz} &\triangleq \exists X_0 \ldots \exists X_{s-1} \exists Y_0 \ldots \exists Y_{t-1} \mathsf{akzBer} \\
\mathsf{akzBer} &\triangleq \mathsf{Partition} \wedge \mathsf{RückPkt} \wedge \mathsf{AnfBed} \wedge \mathsf{EndBed} \\
&\quad \wedge \mathsf{konsTransFolge} \wedge \mathsf{Beschriftung} \wedge \mathsf{Test}
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{Partition} &\triangleq \forall x \Big(\bigvee_{j<t} Y_j x \wedge \bigwedge_{j<t} \big(Y_j x \rightarrow \bigwedge_{j'<j} \neg Y_{j'} x \big) \Big) \\
\mathsf{RückPkt} &\triangleq \bigwedge_{i<s} \forall x \Big(X_i x \leftrightarrow \bigwedge_{j<t:\ \eta_j \text{ setzt } r_i \text{ zurück}} Y_j x \Big) \\
\mathsf{AnfBed} &\triangleq \bigvee_{j:q=q_j} \forall x (0 = x \rightarrow Y_j x) \\
\mathsf{EndBed} &\triangleq \forall x \exists y \Big(x < y \wedge \bigvee_{j:q_j \in F} Y_j y \Big) \\
\mathsf{konsTransFolge} &\triangleq \forall x \forall y \big(\mathrm{suc}(x,y) \rightarrow \bigvee_{j,j':q'_j = q_{j'}} (Y_j x \wedge Y_{j'} y) \big) \\
\mathsf{Beschriftung} &\triangleq \forall x \bigwedge_{a \in A} \big(\mathrm{P}_a x \rightarrow \bigvee_{j:a_j = a} Y_j x \big) \\
\mathsf{Test} &\triangleq \forall x \bigwedge_{j<t} \big(Y_j x \rightarrow \phi'_j \big)
\end{aligned}
$$

mit

$$
\begin{aligned}
\phi'_j &\triangleq \phi_j[\psi_i^C / C r_i] \\
\psi_i^C &\triangleq \mathrm{E}_{C,F^i,H^i}(X_i, \underline{V}, x)
\end{aligned}
$$

$$
f_P^i = \begin{cases} \theta_j(r_i), & \text{falls } P = \{j\}, \\ \text{beliebig}, & \text{sonst}, \end{cases}
$$

$$
h_P^i = \begin{cases} \zeta_j(r_i), & \text{falls } P = \{j\} \text{ und } r_i \in \mathrm{dom}(\zeta_j), \\ \rho(r_i), & \text{falls } P = \emptyset, \\ \text{beliebig}, & \text{sonst}. \end{cases}
$$

Abbildung 2.1: Die Formel akz zur Beschreibung einer akzeptierenden Berechnung

Wir stellen zuerst zwei Vorüberlegungen an.

2.5.1 Lemma *Zu jeder erweitert-atomaren* $\Pi\Sigma$*-Formel* $\mathrm{E}_{C,F,H}(X_i, \underline{Y}, x)$ *mit Mengenvariablen aus* $\mathcal{W}_m$ *gibt es Familien* F' *und* H'*, so daß* $\mathrm{E}_{C,F',H'}(X_i, X_0, \ldots, X_{m-1}, x)$ *und* $\mathrm{E}_{C,F,H}(X_i, \underline{Y}, x)$ *äquivalent sind.*

Beweis. Dies ist eine einfache Folgerung aus den beiden folgenden Behauptungen, die unmittelbar einsichtig sind. Dabei sei E ein erweitertes Symbol vom Typ $(l+1, 1)$ aus $\Pi\Sigma$, weiterhin sei $\underline{Y}$ eine k-gliedrige und es sei $\underline{Y'}$ eine k'-gliedrige Folge von Variablen aus $\mathcal{W}_m$.

Behauptung 1 Die beiden Formeln

$$\mathrm{E}_{C,F,H}(X_i, \underline{Y}, \underline{Y'}, x) \quad \text{und} \quad \mathrm{E}_{C,F',H'}(X_i, \underline{Y}, X, \underline{Y'}, x)$$

sind äquivalent, wenn man $f'_P = f_{P'}$ und $h'_P = h_{P'}$ für jedes $P \subseteq l$ setzt und dabei P' wie folgt wählt:

$$P' \;=\; (P \cap k) \cup \{i - 1 \mid i > k \wedge i \in P\}\,.$$

Behauptung 2 Die beiden Formeln

$$\mathrm{E}_{C,F,H}(X_i, \underline{Y}, X, \underline{Y'}, X', \underline{Y''}, x) \quad \text{und} \quad \mathrm{E}_{C,F',H'}(X_i, \underline{Y}, X', \underline{Y'}, X, \underline{Y''}, x)$$

sind äquivalent, wenn man $f'_P = f_{P'}$ und $h'_P = h_{P'}$ für jedes $P \subseteq l$ setzt und dabei P' wie folgt wählt:

$$P' \;=\; \begin{cases} P \cup \{k\} \setminus \{k + k' + 1\}\,, & \text{falls } k + k' + 1 \in P, \text{ aber } k \notin P, \\ P \cup \{k + k' + 1\} \setminus \{k\}\,, & \text{falls } k \in P, \text{ aber } k + k' + 1 \notin P, \\ P\,, & \text{sonst.} \end{cases}$$

■

Außerdem gilt:

2.5.2 Bemerkung Ist y eine von x verschiedene Variable und $E(\underline{Y}, y)$ eine erweitert-atomare $\Pi\Sigma$-Formel, dann ist $E(\underline{Y}, y)$ äquivalent zu $\exists x(x = y \wedge E(\underline{Y}, x))$.

Also können wir davon ausgehen, daß in jeder existentiell-monadischen $\Pi\Sigma$-Formel mit Variablen aus $\mathcal{W}_m$ jede erweitert-atomare Formel von der Gestalt

$$E(X_i, X_0, \ldots, X_{m-1}, x)$$

ist. Solche Formeln mögen *m-normal* heißen.

Im folgenden sei $\mathcal{T}$ eine endliche Menge von m-normalen erweitert-atomaren $\Pi\Sigma$-Formeln und $B_m(\mathcal{T})$ das Alphabet $A \times \wp(\mathcal{T}) \times \wp(m)$. Weiterhin sei $\exists_m \mathrm{MSO}\Pi\Sigma(\mathcal{T})$ die Bezeichnung für die Menge aller m-normalen $\Pi\Sigma$-Formeln, in denen jede erweitert-atomare Teilformel ein Element aus $\mathcal{T}$ ist.

Es sei $u = (\underline{a}, \underline{w})$ ein W-Wort und $\underline{M}$ eine m-gliedrige Folge von Mengen $M_i \subseteq |u|$. Die $(m, \mathcal{T})$-*Kodierung von u und $\underline{M}$* soll das diskrete Wort

$$v \;=\; (a_0, T_0, N_0)(a_1, T_1, N_1)(a_2, T_2, N_2)\ldots$$

über $B_m(\mathcal{T})$ sein, wobei für $i < |u|$ die Menge N_i durch $N_i = \{k \mid i \in M_k\}$ gegeben sei und für jede Interpretation $(u^{\Pi\Sigma}, \nu)$ mit einer Variablenbelegung ν, die $\nu(X_i) = M_i$ für $i < m$ erfüllt, gelte:

$$(u, \nu) \models \alpha \quad \text{gdw.} \quad \alpha \in T_{\nu(x)} \qquad \text{für } \alpha \in \mathcal{T}\,.$$

Sprechen wir von einem Wort v als einer $(m, \mathcal{T})$-Kodierung eines Wortes u, dann meinen wir, daß es ein Folge $\underline{M}$ gibt, so daß v die $(m, \mathcal{T})$-Kodierung von u und $\underline{M}$ ist.

Wir definieren eine Funktion

$$^*\colon \exists_m \mathrm{MSO}\Pi\Sigma(\mathcal{T}) \to \exists \mathrm{MSO}\Pi(B_m(\mathcal{T}))\,.$$

Ist $\phi \triangleq \exists \underline{Y} \phi_0$ ein Element von $\exists_m \mathrm{MSO}\Pi\Sigma(\mathcal{T})$, so sei ϕ^* die Formel, die man aus ϕ durch Entfernen des Präfixes $\exists \underline{Y}$ und durch Ersetzen

- jeder atomaren Formel $\mathrm{P}_a y$ durch $\displaystyle\bigvee_{(b,T,N):a=b} \mathrm{P}_{(b,T,N)} y$,
- jeder atomaren Formel $X_i y$ durch $\displaystyle\bigvee_{(a,T,N):i\in N} \mathrm{P}_{(a,T,N)} y$,
- und jeder erweitert-atomaren Formel $\alpha \in \mathcal{T}$ durch $\displaystyle\bigvee_{(a,T,N):\alpha\in T} \mathrm{P}_{(a,T,N)} x$.

erhält.

Die Verträglichkeit von $(m, \mathcal{T})$-Kodierungen mit der Übersetzungsfunktion * ergibt sich unmittelbar (per Induktion über den Formelaufbau):

2.5.3 Lemma *Sei $\phi \in \exists_m \mathrm{MSO}\Pi\Sigma(\mathcal{T})$. Dann gilt für jedes W-Wort u: Eine Interpretation $(u^{\Pi\Sigma}, \nu)$ erfüllt genau dann ϕ, wenn $(v^{\Pi''}, \nu) \models \phi^*$ gilt, für die $(m, \mathcal{T})$-Kodierung v des bewerteten Wortes u und der Folge $\nu(X_0), \ldots, \nu(X_{m-1})$ und mit $\Pi'' = \Pi(B_m(\mathcal{T}))$.* ■

Die hier vorgestellte „Kodierung“, die Funktion * und 2.5.3 sind angelehnt an den Interpretierbarkeitsbegriff, wie er in der Modelltheorie der Logik erster Stufe Verwendung findet (siehe z. B. [Hod93]). Hier wurde er den Notwendigkeiten, die bei der Betrachtung existentiell-monadischer Formeln entstehen, angepaßt. In Abschnitt 2.7.2 werden wir diese Idee wieder aufgreifen und um den Aspekt der Biinterpretierbarkeit ergänzen.

2.6 Konstruktion von Σ-Automaten aus diskreten Automaten

Zu einem gegebenen diskreten Rabin-Scott- bzw. Büchi-Automaten $\mathfrak{A}$ über dem Alphabet $B_m(\mathcal{T})$ konstruieren wir (effektiv) einen Σ-Automaten $\hat{\mathfrak{A}}$, der genau die Menge aller W-Wörter akzeptiert, die eine $(m,\mathcal{T})$-Kodierung besitzen, die von $\mathfrak{A}$ akzeptiert wird.

Wir gehen von einem diskreten Automaten $\mathfrak{A}$ über $B_m(\mathcal{T})$ aus, der in der Form $\mathfrak{A} = (Q, s, \Delta, F_0)$ gegeben sei, und beschreiben $\hat{\mathfrak{A}}$ mit $\hat{\mathfrak{A}} = (Q, R, (s,\sigma), \Delta', F_0)$. Die Registermenge R beinhalte für jedes Tripel (i,F,H), für das es ein Testprädikat C gibt, so daß $\mathrm{E}_{C,F,H}(X_i, \underline{X}, x)$ zu $\mathcal{T}$ gehört, ein Register $r_{i,F,H}$. Die Anfangsbelegung sei durch $\rho(r_{i,F,H}) = h_\emptyset$ für jedes $r_{i,F,H} \in R$ definiert. Für jede Transition $\delta = (q,(a,T,N),q')$ aus Δ gebe es in Δ' eine Transition $\hat{\delta} = (q,a,\theta,\phi,\zeta,q')$ mit

$$\begin{aligned}
\theta(r_{i,F,H}) &= f_N && \text{für } r_{i,F,H} \in R,\\
\phi &\triangleq \bigwedge_{\alpha\in T} \alpha \wedge \bigwedge_{\alpha\in\mathcal{T}\setminus T} \neg\alpha\,, &&\\
\zeta(r_{i,F,H}) &= \begin{cases} h_N\,, & \text{falls } i \in N,\\ \text{undefiniert}, & \text{sonst,}\end{cases} && \text{für } r_{i,F,H} \in R.
\end{aligned}$$

Der konstruierte Automat hat die gewünschten Eigenschaften, wie aus dem folgenden Lemma ersichtlich ist:

2.6.1 Lemma *Sei $\mathfrak{A}$ ein diskreter Automat über $B_m(\mathcal{T})$. Dann gilt $u \in \mathrm{L}(\hat{\mathfrak{A}})$ für ein W-Wort u über A genau dann, wenn es eine $(m,\mathcal{T})$-Kodierung von u gibt, die zu $\mathrm{L}(\mathfrak{A})$ gehört.*

Beweis. Ist $(\underline{a},\underline{w})$ ein W-Wort und dürfen wir

$$(q_0,\rho_0) \xrightarrow[(a_0,w_0)]{\hat{\delta}_0} (q_1,\rho_1) \xrightarrow[(a_1,w_1)]{\hat{\delta}_1} (q_2,\rho_2) \xrightarrow[(a_2,w_2)]{\hat{\delta}_2} \dots \tag{2.5}$$

schreiben, wobei (q_0,ρ_0) die Anfangskonfiguration von $\hat{\mathfrak{A}}$ sei, so gilt nach Konstruktion mit den Bezeichnungen $\delta_i = (q_i,(a_i,T_i,N_i),q_{i+1})$ und $M_j = \{k \mid j \in N_k\}$ für jedes $k < |(\underline{a},\underline{w})|$ und jedes in Frage kommende Tripel (i,F,H):

$$\mathrm{E}^u_{F,H}(\underline{M}, \mathrm{vg}(k,M_i), k) = f_{N_k}(\rho_k(r_{i,F,H}), w_k)\,, \tag{2.6}$$

$$\rho_{k+1}(r_{i,F,H}) = \begin{cases} \mathrm{E}^u_{F,H}(\underline{M}, \mathrm{vg}(k,M_i), k)\,, & \text{falls } i \notin N_k\\ h_{N_k}\,, & \text{sonst,}\end{cases} \tag{2.7}$$

was man durch Induktion über k leicht nachweist. Ebenso gilt für jedes $\alpha \in \mathcal{T}$ mit $\alpha = \mathrm{E}_{C,F,H}(X_i, \underline{X}, x)$:

$$\mathrm{E}^u_{C,F,H}(M_i, \underline{M}, k) \quad \text{gdw.} \quad \alpha \in T_k\,. \tag{2.8}$$

Also ist $(\underline{a}, \underline{T}, \underline{N})$ eine Kodierung von $(\underline{a}, \underline{w})$ und $\underline{M}$, für die wir

$$q_0 \xrightarrow[(a_0,f_0,N_0)]{\delta_0} q_1 \xrightarrow[(a_1,f_1,N_1)]{\delta_1} q_2 \xrightarrow[(a_2,f_2,N_2)]{\delta_2} \dots \tag{2.9}$$

in $\mathfrak{A}$ schreiben dürfen. Daraus folgt die eine Implikation des Lemmas.

Dürfen wir umgekehrt (2.9) für eine Kodierung $(\underline{a}, \underline{T}, \underline{N})$ eines W-Wortes $(\underline{a}, \underline{w})$ und einer Folge $\underline{M}$ schreiben, so auch (2.5), wobei wieder (2.6) bis (2.8) gelten. Daraus folgt die umgekehrte Richtung der Behauptung. ■

Nun können wir den Beweis des Charakterisierungssatzes 2.3.2 vollenden.

Beweis der Implikation von 1) nach 2) im Charaktersieriungssatz. Wegen 2.5.1 und 2.5.2 können wir davon ausgehen, daß ein m-normaler ΠΣ-Satz vorliegt. Es sei $\mathcal{T}$ die Menge aller in ϕ auftretenden erweitert-atomaren Formeln. Es sei $\mathfrak{A}$ der nach 2.1.3 (Satz von Büchi und Elgot) zu ϕ^* äquivalente Automat über $B_m(\mathcal{T})$. Dann gilt:

$$\begin{aligned}
& u \in \mathrm{Mod}(\phi) \\
\text{gdw.}\quad & \exists v\ (v \text{ ist } (m,\mathcal{T})\text{-Kodierung von } u \text{ und } v \in \mathrm{Mod}(\phi^*))\,, && \text{(nach 2.5.3)} \\
\text{gdw.}\quad & \exists v\ (v \text{ ist } (m,\mathcal{T})\text{-Kodierung von } u \text{ und } v \in \mathrm{L}(\mathfrak{A}))\,, && \text{(nach 2.1.3)} \\
\text{gdw.}\quad & u \in \mathrm{L}(\hat{\mathfrak{A}})\,, && \text{(nach 2.6.1)}
\end{aligned}$$

womit $\hat{\mathfrak{A}}$ der gesuchte Automat ist.

Die Effektivität der Konstruktion des Automaten $\hat{\mathfrak{A}}$ ergibt sich zum einen aus dem Zusatz in 2.1.3, zum anderen aus der Möglichkeit, die Umwandlung einer existentiell-monadischen ΠΣ-Formel in eine m-normale Formel effektiv durchführen zu können. ■

Die Konstruktion von $\hat{\mathfrak{A}}$ ist derart, daß die Anzahl der Register mit dem E-Rang von ϕ übereinstimmt.

Es sei der *Registerindex* eines Σ-Automaten die Anzahl seiner Register und der Registerindex einer Σ-erkennbaren Sprache der minimale Registerindex aller die Sprache erkennenden Σ-Automaten.

Dann erhalten wir aus 2.4.2 zusammen mit der obigen Beobachtung:

2.6.2 Bemerkung Der Registerindex und der E-Rang einer Σ-erkennbaren (bzw. existentiell-monadisch ΠΣ-definierbaren) Sprache sind gleich.

2.7 Wortstrukturen, erweitert-existentielle Formeln und Mengenterme

Zuweilen ist es günstig, wenn man sich beim tatsächlichen Schreiben von Formeln einen gewissen Komfort erlauben kann. So haben wir z.B. in Abschnitt 2.4 beim Schreiben der Formel akz mehrfachen Gebrauch von der Abkürzung suc gemacht. In diesem Sinne werden in diesem Abschnitt Spracherweiterungen von $\exists\mathrm{MSO}\Pi\Sigma$ studiert, die zwar die Ausdrucksstärke nicht ändern, aber in späteren Kapiteln beim Notieren von Formeln nicht nur Platz sparen, sondern vor allem zur Übersichtlichkeit beitragen werden.

Zum einen wird die Signatur Π (außer um ein eigenes Relationssymbol für suc) um Funktionssymbole und Konstanten erweitert, die einen einfachen Umgang mit der den Wortstrukturen zugrundeliegenden diskreten Ordnungsrelation erlauben (Unterabschnitt 2.7.4). Zum anderen werden zur leichteren Handhabung von Mengen sogenannte ‚Mengenterme' eingeführt (Unterabschnitt 2.7.3). Beide Erweiterungen sind unproblematisch, denn es ist leicht einzusehen, daß sie die Ausdrucksstärke von $\exists\mathrm{MSO}\Pi\Sigma$ nicht vergrößern. Demgegenüber ist die dritte Erweiterung, die wir behandeln, komplizierter: An gewissen Stellen werden in existentiell-monadischen $\Pi\Sigma$-Formeln Allquantoren erlaubt.[1] Um zu zeigen, daß diese Spracherweiterung keine Vergrößerung der Ausdrucksstärke mit sich bringt, greifen wir auf die Implikation von 1) nach 2) in Satz 2.1.3 zurück, d.h., wir machen von der Tatsache Gebrauch, daß in monadischen Π-Formeln (interpretiert in diskreten Wortstrukturen) auf Allquantoren verzichtet werden kann. Aus logischer Sicht handelt es sich bei dem Ergebnis um eine Quantorenelimination (Unterabschnitt 2.7.2).

Alle Erweiterungen werden in (fast) voller Allgemeinheit vorgestellt, weshalb wir mit neuen Begriffsbildungen beginnen.

2.7.1 Wortsignaturen

Unter einer *Wortsignatur* wollen wir eine erweiterte Signatur verstehen, die das zweistellige Symbol $<$ enthält und ansonsten nur erweiterte Symbole mit einem oder keinem Elementargument. (D.h., für jedes erweiterte Symbol gibt es eine Zahl k, so daß es vom Typ $(k,0)$ oder $(k,1)$ ist.) Insbesondere ist für eine Berechnungsstruktur Σ die Signatur $\Pi\Sigma$ eine Wortsignatur, wenn man die Buchstabenprädikate als erweiterte Prädikate des Typs $(0,1)$ auffaßt.

Im Rest des Abschnitts sei Ω eine Wortsignatur, sofern nichts anderes vermerkt ist.

Eine Ω-Struktur heiße *Wortstruktur*, wenn ihr Universum ein Anfangsstück der

[1] Wie schon eingangs dieses Kapitels dargelegt wurde, ist i.a. nicht zu erwarten, daß sogar jede beliebige monadische $\Pi\Sigma$-Formel zu einer existentiell-monadischen $\Pi\Sigma$-Formel äquivalent ist.

natürlichen Zahlen (insbesondere **N** selbst) ist und $<$ mit der natürlichen Ordnungsrelation auf dem Universum übereinstimmt. Man beachte, daß diese Definition nicht mit dem Begriff der ΠΣ-Wortstruktur in Konflikt gerät. Denn jede ΠΣ-Wortstruktur ist auch eine Wortstruktur in dem hier definierten Sinn.

Monadische Ω-Formeln mögen im Rest dieses Abschnitts entweder nur in endlichen oder nur in unendlichen Wortstrukturen interpretiert werden, wie wir es von Π- und ΠΣ-Formeln gewohnt sind.

2.7.2 Erweitert-existentielle Formeln

Wir beginnen mit der Spracherweiterung um Allquantoren.

2.7.1 Definition (erweitert-existentielle Formel) Eine monadische Ω-Formel der Gestalt $\exists X_0 \ldots \exists X_{l-1} \phi_0$, in der jede in einer erweitert-atomaren Teilformel gebundene Mengenvariable von dem Präfix $\exists X_0 \ldots \exists X_{l-1}$ gebunden wird, heißt *erweitert-existentielle Formel.*

Z. B. ist die Formel

$$\exists X_0 \forall X_1 (E(X_0, X_1) \wedge E(X_1, X_2))$$

keine erweitert-existentielle Formel, dagegen aber

$$\exists X_0 \exists X_1 \forall X_2 (E(X_0, X_1) \wedge \exists x_0\, X_2 x_0)\,,$$

wobei angenommen wird, daß E ein erweitertes Symbol vom Typ $(2,0)$ ist.

Wir zeigen nun:

2.7.2 Lemma *Jede erweitert-existentielle Ω-Formel kann effektiv in eine äquivalente existentiell-monadische Ω-Formel umgewandelt werden.*

Beweis. Wir benutzen wieder die Idee der Interpretation von Strukturen zusammen mit Folgen von Teilmengen des Universums in diskreten Wortstrukturen.

O. E. nehmen wir an, daß die Signatur Ω keine erweiterten Symbole des Typs $(k,0)$ besitze. (Hat sie diese doch, so gehen wir zu der Signatur über, die anstelle jedes erweiterten Symbols E des Typs $(k,0)$ ein entsprechendes Symbol E' des Typs $(k,1)$ hat.)

Es sei x wieder eine feste Elementvariable.

Eine erweitert-existentielle Ω-Formel ϕ heiße (l,m)-normal, wenn sie von der Gestalt $\exists X_0 \ldots \exists X_{l-1} \phi_0$ ist und die folgende Eigenschaften besitzt:

- Die freien Mengenvariablen in ϕ sind unter $X_l, \ldots, X_{m-1}$.
- Ist $E(\underline{Y}, x)$ eine erweitert-atomare Teilformel von ϕ, so gilt $x = y$.

- Keine der Variablen aus $\mathcal{W}_m = \{X_0, \ldots, X_{m-1}\}$ wird in ϕ_0 quantifiziert.

Es sei $\mathcal{T}_0$ eine endliche Menge von erweitert-atomaren (l,m)-normalen Formeln, $\mathcal{T} = \mathcal{T}_0 \cup \{X_i x \mid i < m\}$ und $B = \wp(\mathcal{T})$. Die *$\mathcal{T}$-Kodierung* einer Ω-Wortstruktur $\mathfrak{M}$ und einer m-gliedrigen Folge von Teilmengen von M (dem Universum von $\mathfrak{M}$) sei das Wort $\underline{T}$ über dem Alphabet B, das M als Indexmenge hat und dergestalt ist, daß für jede Interpretation $(\mathfrak{M}, \nu)$ gilt:

$$(\mathfrak{M}, \nu) \models \alpha \quad \text{gdw.} \quad \alpha \in T_{\nu(x)} \qquad \text{für } \alpha \in \mathcal{T}.$$

Wir definieren zwei Funktionen, * und $\hat{}$, die eine (l,m)-normale Formel in eine monadische $\Pi(B)$-Formel ohne Variablen aus $\mathcal{W}_m$ bzw. eine existentiell-monadische $\Pi(B)$-Formel ohne Variablen aus $\mathcal{W}_m$ in eine existentiell-monadische Ω-Formel umwandeln.

Es sei ϕ^* die Formel, die aus ϕ durch Entfernen des Präfixes $\exists X_0 \ldots \exists X_{l-1}$ und durch Ersetzen jeder Formel $\alpha \in \mathcal{T}$ durch

$$\bigvee_{T : \alpha \in T} \mathrm{P}_T x$$

entsteht.

Es sei $\hat{\phi}$ die Formel, die aus ϕ durch Vorschreiben des Präfixes $\exists X_0 \ldots \exists X_{l-1}$ und durch Ersetzen jeder atomaren Formel $\mathrm{P}_T y$ durch

$$\exists x (x = y \wedge \bigwedge_{\alpha \in T} \alpha \wedge \bigwedge_{\alpha \in \mathcal{T} \setminus T} \neg \alpha)$$

entsteht.

In Analogie zu 2.5.3 ergibt sich nun:

Behauptung 1) Sei ϕ eine (l,m)-normale Formel. Dann gilt für jede Ω-Wortstruktur $\mathfrak{M}$: Eine Interpretation $(\mathfrak{M}, \nu)$ erfüllt genau dann ϕ, wenn für die $\mathcal{T}$-Kodierung v von u und $\nu(X_0), \ldots, \nu(X_{m-1})$ die Beziehung $(v^{\Pi(B)}, \nu) \models \phi^*$ gilt.

2) Sei ϕ eine $\Pi(B)$-Formel ohne Mengenvariablen aus $\mathcal{W}_m$. Dann gilt für jede Ω-Wortstruktur $\mathfrak{M}$: Eine Interpretation $(\mathfrak{M}, \nu)$ erfüllt genau dann $\hat{\phi}$, wenn für die $\mathcal{T}$-Kodierung v von u und $\nu(X_0), \ldots, \nu(X_{m-1})$ die Beziehung $(v^{\Pi(B)}, \nu) \models \phi$ gilt.

Um nun aus einer beliebigen erweitert-existentiellen Ω-Formel ϕ eine äquivalente existentiell-monadische Ω-Formel zu konstruieren, geht man zuerst zu einer äquivalenten (l,m)-normalen Formel ϕ' über, überführt ϕ'^* mit Hilfe von 2.1.3 in eine äquivalente existentiell-monadische Formel ψ ohne Variablen aus $\mathcal{W}_m$ und konstruiert dazu $\hat{\psi}$. ■

Zwei Dinge seien hier kurz angemerkt. Erstens ist durch die beiden Funktionen * und $\hat{}$ und durch die obige Behauptung eine Art Biinterpretierbarkeit von Ω- und $\Pi(B)$-Wortstrukturen gegeben. Zweitens erscheint die hier vorgenommene Kodierung einfacher als die, die in Abschnitt 2.5 vorgestell wurde: das Alphabet B ist nur eine Menge von atomaren Formeln und nicht, wie $B_m(\mathcal{T})$, ein kartesisches Produkt aus drei Mengen. Die etwas aufwendigere Konstruktion in Abschnitt 2.5 erleichtert die spätere Konstruktion des Automaten $\hat{\mathfrak{A}}$ in Abschnitt 2.6.

2.7.3 Mengenterme

Generell kann man in der Sprache der monadischen Logik in einer beliebigen (erweiterten oder nicht erweiterten) Signatur Ω mehr Freizügigkeit im Umgang mit Mengen erlauben, ohne daß die Ausdrucksstärke geändert wird. An den Stellen, wo normalerweise eine Mengenvariable steht, können auch ‚Mengenterme' zugelassen werden.

Formal führt man in der induktiven Definition der Syntax eine zusätzliche Regel für die Konstruktion von (Mengen-)Termen ein und erweitert die Regel zur Konstruktion atomarer Formeln:

Mengenterm: Ist ϕ eine bislang konstruierte Formel und x eine Elementvariable, so ist

$$\{x \mid \phi\} \tag{2.10}$$

ein *Mengenterm*, dessen freie Variablen die Variablen von ϕ, ausgenommen x, sind.

Atomare Formel: Ist E ein Prädikat vom Typ (m,n), sind $T_0, \ldots, T_{m-1}$ Mengenterme oder Mengenvariablen und sind $t_0, \ldots, t_{n-1}$ herkömmliche Terme, dann ist

$$E(T_0, \ldots, T_{m-1}, t_0, \ldots, t_{n-1})$$

eine erweitert-atomare Formel.

(Man beachte, daß dadurch eine Schachtelung von Formeln und Termen möglich ist: In einer atomaren Formel kann ein Mengenterm auftreten, in dem wieder eine atomare Formel steht, die wiederum einen Mengenterm als Argument hat, usw.)

Die Semantik wird dann in entsprechender Weise definiert. In einer gegebenen Interpretation $(\mathfrak{M}, \nu)$ wird dem Term (2.10) die Menge

$$\{k \in M \mid (\mathfrak{M}, \nu\left[\tfrac{k}{x}\right]) \models \phi\}$$

zugeordnet. (Hierbei steht $\nu\left[\frac{k}{x}\right]$ für die Belegung, die aus ν entsteht, indem der Wert an der Stelle x zu k abgeändert wird.)

Man kann nun zeigen, daß jede monadische Formel mit Mengentermen zu einer Formel ohne Mengenvariablen äquivalent ist. Im allgemeinen gilt dies nicht für erweitert-existentielle oder existentiell-monadische Formeln. Es dürfen nur gewisse Mengenterme zugelassen werden:

2.7.3 Definition (erweitert-existentielle Formel mit Mengentermen)
Eine monadische Ω-Formel mit Mengentermen heißt *erweitert-existentiell*, falls in jedem Mengenterm, der als Argument in einer erweitert-atomaren Teilformel auftritt, keine freie Elementvariable vorkommt und an freien Mengenvariablen nur solche auftreten, die entweder gar nicht oder durch das existentielle Präfix der ganzen Formel gebunden werden.

Wir zeigen nun:

2.7.4 Lemma *Jede erweitert-existentielle Ω-Formel mit Mengentermen kann effektiv in eine äquivalente erweitert-existentielle Formel ohne Mengenterme umgewandelt werden.*

Beweis. Dazu nehmen wir an, daß $\phi \triangleq \exists X_0 \ldots \exists X_{m-1} \phi_0$ eine erweitert-existentielle Formel mit mindestens einem Mengenterm sei. O.E. enthalte ϕ_0 keine Quantifizierungen von Variablen aus $\underline{X}$. Dann gibt es in ϕ auch einen Mengenterm $T = \{x \mid \psi\}$, so daß ψ ohne Mengenterme ist, d.h., T ist ein innerster Mengenterm. Wir schreiben ϕ_0 in der Form $\phi_0 = \phi_0(T)$.

Definitionsgemäß gibt es in ψ außer x keine freie Elementvariable und es werden alle in ϕ gebundenen Mengenvariablen von ψ (erst) durch die Variablen im Präfix $\exists \underline{X}$ gebunden. Dann ist ϕ aber zu der Formel

$$\phi' \quad \triangleq \quad \exists Y \exists \underline{X} (\forall x (Yx \leftrightarrow \psi) \wedge \phi_0(Y))$$

äquivalent, wenn Y eine neue Mengenvariable ist.

In der neuen Formel ϕ' kommt ein Mengenterm weniger als in der ursprünglichen Formel ϕ vor. Durch wiederholte Anwendung des Verfahrens kann also ϕ von Mengentermen befreit werden. ■

Wir führen einige hilfreiche Abkürzungen ein. Für den Term $\{x \mid Xx \vee Yx\}$ schreiben wir einfach $X \cup Y$. In diesem Sinne mögen auch $X \cap Y$, $X \cup Y \cup Z$, usw. verstanden werden. Außerdem wird P_a für $\{x \mid \mathrm{P}_a x\}$ benutzt werden, und es wird $\emptyset$ für $\{x \mid \mathtt{ff}\}$ stehen.

2.7.4 Erweiterte Wortstrukturen

Die Signatur Π wurde in Abschnitt 2.3 absichtlich sparsam gewählt, um in Beweisen nicht unnötige Fallunterscheidungen durchführen zu müssen. Hier zeigen wir nun, daß die üblichen Erweiterungen die Ausdrucksstärke nicht verändern.

Die Signatur Υ enthalte außer dem zweistelligen Symbol $<$ das zweistellige Symbol suc, die Konstante 0 und die einstelligen Funktionssymbole $+1$ und -1 (in Postfixschreibweise).

Im folgenden enthalte die Signatur Ω keine Symbole aus Υ und $\Omega \cup \{<\}$ sei eine Wortsignatur.

Ist nun $\mathfrak{M}$ eine $(\Omega \cup \{<\})$-Wortstruktur, so gewinnt man aus ihr in natürlicher Weise eine $(\Omega \cup \Upsilon)$-Struktur durch die folgenden Festlegungen: Wir setzen $\mathrm{suc}^{\mathfrak{M}} = \{(i, i+1) \mid i+1 < |\mathfrak{M}|\}$ und $0^{\mathfrak{M}} = 0$. Außerdem gelte

$$+1^{\mathfrak{M}}(j) = \begin{cases} j\,, & \text{falls } M \text{ endlich und } j = |M| - 1, \\ j+1\,, & \text{sonst,} \end{cases}$$

$$-1^{\mathfrak{M}}(j) = \begin{cases} 0\,, & \text{falls } j = 0, \\ j-1\,, & \text{sonst,} \end{cases}$$

für $j < |M|$.

Die in dieser Weise entstehenden Strukturen heißen *erweiterte Wortstrukturen.*

2.7.5 Lemma *Zu jeder existentiell-erweiterten $(\Omega \cup \Upsilon)$-Formel ϕ läßt sich effektiv eine existentiell-erweiterte $(\Omega \cup \{<\})$-Formel konstruieren, die genau dann von einer Wortstruktur erfüllt wird, wenn die zugehörige erweiterte Wortstruktur ϕ erfüllt.*

Beweis. Wir beschreiben, wie eine gegebene erweitert-existentielle $(\Omega \cup \Upsilon)$-Formel $\phi \triangleq \exists X_0 \ldots X_{m-1} \phi_0$ schrittweise umgewandelt werden kann. Die Effektivität des Verfahrens wird offensichtlich sein.

Wieder mögen o. E. in ϕ_0 keine Quantifizierungen von Variablen aus $\mathcal{W}_m$ auftreten.

Zuerst werden die Vorkommen der Konstanten 0 eliminiert: Enthält ϕ eine atomare Teilformel $\psi \triangleq \psi(t)$, in der ein konstanter Term der Form $t = t(0)$ auftritt, so kann ψ durch die Formel

$$\exists y(\psi(t(y)) \wedge \neg \exists x\, \mathrm{suc}(x, y))$$

äquivalent ersetzt werden, wenn y eine Variable ist, die in ψ nicht auftritt. Iterierte Anwendung des Verfahrens läßt das Symbol 0 verschwinden.

Sei nun ϕ eine Formel mit einer atomaren Teilformel $\psi = \psi(t)$, in der ein Term t der Form $t = t' + 1$ oder $t = t' - 1$ enthalten ist. Im ersten Fall kann ψ durch

$$\exists y(\mathrm{suc}(t', y) \wedge \psi(y))$$

ersetzt werden. Im anderen Fall nimmt man stattdessen die Formel

$$\exists y(\mathrm{suc}(y, t') \wedge \psi(y)) \vee (\neg \exists y\, \mathrm{suc}(y, t') \wedge \psi(t'))\,.$$

Wiederholte Anwendung des Ersetzungsprozesses liefert eine Formel ohne die Symbole 0, +1 und −1.

Schließlich kann man jede atomare Formel $\mathrm{suc}(x, y)$ durch

$$x < y \wedge \neg \exists z(x < z \wedge z < y)$$

ersetzen und gelangt dadurch zu der gewünschten Formel. ■

Das Konstantensymbol 0 steht für das kleinste Element in einer (erweiterten) Wortstruktur. In endlichen Wortstrukturen hat man auch ein größtes Element. Für dieses führen wir das Symbol max ein. Dann gilt 2.7.5 entsprechend auch für endliche erweiterte Wortstrukturen und Formeln in der Signatur $\Omega \cup \Upsilon \cup \{\mathsf{max}\}$.

Aus 2.7.2, 2.7.4 und 2.7.5 ergibt sich nun:

2.7.6 Korollar *Jede erweitert-existentielle* $(\Omega\cup\Upsilon)$- *bzw.* $(\Omega\cup\Upsilon\cup\{\mathsf{max}\})$-*Formel mit Mengentermen läßt sich effektiv in eine äquivalente existentiell-monadische* $(\Omega\cup\{<\})$-*Formel umwandeln.*

Die Untersuchungen dieses Abschnitts schlagen sich in einer zusätzlichen Äquivalenz in der Formulierung des Charakterisierungssatzes nieder. Der leichteren Lesbarkeit wegen schreiben wir für $\Pi\Sigma \cup \Upsilon$ bzw. $\Pi\Sigma \cup \Upsilon \cup \{\mathsf{max}\}$ einfach $\Pi'\Sigma$.

2.7.7 Erweiterter Charakterisierungssatz *Sei L eine W-Sprache. Dann sind die folgenden Aussagen äquivalent.*

1) *L ist existentiell-monadisch* $\Pi\Sigma$*-definierbar.*

2) *L ist definierbar durch eine erweitert-existentielle* $\Pi'\Sigma$*-Formel mit Mengentermen.*

3) *L ist* Σ*-erkennbar.*

Die jeweiligen Umwandlungen können effektiv erfolgen.

Dementsprechend sind die meisten Ergebnisse, die wir im weiteren Verlauf der Arbeit für existentiell-monadische $\Pi\Sigma$-Formeln erzielen werden, auch für erweitert-existentielle $\Pi'\Sigma$-Formeln mit Mengentermen gültig. Weil wir die Formeln als Abkürzungen für $\exists\mathrm{MSO}\Pi\Sigma$-Formeln verstehen wollen, werden die entsprechenden Ergebnisse nicht mit aufgeführt werden.

Als abkürzende Schreibweisen führen wir ein: **1** für den Term $0+1$, **2** für $0+1+1$, usw., ebenso wie $x + \mathbf{2}$ für $x + 1 + 1$, $x + \mathbf{3}$ für $x + 1 + 1 + 1$, usw. Entsprechendes gelte für -1.

2.8 Erfüllbarkeit und Regularität diskreter Projektionen

Es ist bekannt, daß das Leerheitsproblem, also die Frage danach, ob ein gegebener Automat kein Wort akzeptiert, für diskrete Büchi-Automaten entscheidbar ist ([Büc62]). Dies gilt nicht allgemein für Σ-Automaten (vgl. Abschnitt 4.7, siehe z. B. 4.7.10) —

hier spielt die Stärke bzw. die Schwäche der gegebenen Berechnungsstruktur Σ eine wesentliche Rolle.

Auf der logischen Seite entspricht dem Leerheitsproblem das Erfüllbarkeitsproblem: Besitzt eine gegebene Formel eine Interpretation, in der sie gültig ist?

Da durch den Charakterisierungssatz eine effektive Korrespondenz zwischen Σ-Automaten und existentiell-monadischen Σ-Sätzen gegeben ist, sind das Leerheitsproblem für Σ-Automaten und das Erfüllbarkeitsproblem für existentiell-monadische $\Pi\Sigma$-Formeln gleich schwierig, d. h.:

2.8.1 Satz *Das Erfüllbarkeitsproblem für existentiell-monadische $\Pi\Sigma$-Formeln ist genau dann entscheidbar, wenn das Leerheitsproblem für Σ-Automaten entscheidbar ist.*

Beweis. Nach 2.3.2, 2)$\Rightarrow$1), kann man jeden Σ-Automaten $\mathfrak{A}$ effektiv in einen äquivalenten existentiell-monadischen $\Pi\Sigma$-Satz ϕ umwandeln. Also ist die von $\mathfrak{A}$ erkannte Sprache genau dann leer, wenn ϕ nicht erfüllbar ist. Das Leerheitsproblem kann also auf das Erfüllbarkeitsproblem reduziert werden. In diesem Sinne kann auch durch die auf S. 33 vorgestellte Konstruktion eines Σ-Automaten aus einem existentiell-monadischen $\Pi\Sigma$-Satz benutzt werden, um die Reduzierbarkeit des Erfüllbarkeitsproblems auf das Leerheitsproblem zu zeigen. (Beachte, daß es genügt, die Behauptung für Sätze zu zeigen. Denn trivialerweise kann jede existentiell-monadische Formel $\phi \triangleq \exists \underline{X} \phi_0$ mit freien Variablen $Y_0, \ldots, Y_{m-1}$ und $x_0, \ldots, x_{l-1}$ in den existentiell-monadischen $\Pi\Sigma$-Satz $\exists \underline{X} \exists \underline{Y} \exists \underline{x} \phi_0$ umgewandelt werden, der das gleiche Erfüllbarkeitsproblem hat.) ■

Dieser Satz erlaubt die direkte Übertragung von automatentheoretischen Ergebnissen in die Logik, wovon wir in Kapitel 5 Gebrauch machen werden.

Daneben setzt uns die spezielle Form der benutzten Umwandlung von existentiell-monadischen $\Pi\Sigma$-Formeln in Automaten über die Büchische Logik in die Lage, für die Entscheidbarkeit des Erfüllbarkeitsproblems ein interessantes hinreichendes Kriterium in logischer Formulierung anzugeben. Außerdem erhalten wir ein Kriterium für die Regularität der diskreten Projektionen von Σ-erkennbaren Sprachen.

2.8.2 Satz *1) Gibt es zu jeder natürlichen Zahl m und jeder endlichen Menge $\mathcal{T}'$ von m-normalen erweitert-atomaren $\Pi\Sigma$-Formeln eine endliche Obermenge $\mathcal{T}$ von m-normalen erweitert-atomaren $\Pi\Sigma$-Formeln und einen $\Pi(B_m(\mathcal{T}))$-Satz $\mathsf{Interpretation}_m(\mathcal{T})$, so daß für jedes diskrete Wort v über $B_m(\mathcal{T})$ gilt:*

$$v \models \mathsf{Interpretation}_m(\mathcal{T})$$

gdw. v ist die $(m, \mathcal{T})$-Kodierung eines W-Wortes,

dann ist die diskrete Projektion jeder Σ-erkennbaren Sprache regulär.

2) Sind die Voraussetzungen von Punkt 1) gegeben und lassen sich zusätzlich die Sätze $\mathsf{Interpretation}_m(\mathcal{T})$ effektiv bestimmen, dann

(a) läßt sich zu jedem Σ-Automaten $\mathfrak{A}$ effektiv ein diskreter Automat konstruieren, der die diskrete Projektion von $\mathrm{L}(\mathfrak{A})$ erkennt;

(b) sind das Leerheitsproblem für Σ-Automaten und das Erfüllbarkeitsproblem für existentiell-monadische $\Pi\Sigma$-Formeln entscheidbar.

Mit anderen Worten: Sind die Mengen der Kodierungen von W-Wörtern endlich axiomatisierbar, so ist die diskrete Projektion jeder Σ-erkennbaren Sprache regulär. Sind die Axiome uniform, so sind auch das Leerheitsproblem für Σ-Automaten und das Erfüllbarkeitsproblem für erweitert-existentielle Σ-Formeln entscheidbar.

Beweis. 1) Zu einem gegebenen Σ-Automaten $\mathfrak{A}$ konstruiere man eine äquivalente m-normale existentiell-monadische $\Pi\Sigma$-Formel $\phi \triangleq \exists X_0 \ldots X_{m-1}\phi_0$ (siehe 2.4.1, 2.5.1 und 2.5.2). Es sei $\mathcal{T}'$ die Menge der in ϕ_0 enthaltenen erweitert-atomaren Formeln und $\mathcal{T}$ die Menge aus der Voraussetzung. Dann enthält $\mathrm{Mod}(\chi)$ mit

$$\chi \quad \triangleq \quad \phi^* \wedge \mathsf{Interpretation}_m(\mathcal{T})$$

nach 2.5.3 nur $(m,\mathcal{T})$-Kodierungen von W-Wörtern aus $\mathrm{L}(\mathfrak{A})$ und auch mindestens eine $(m,\mathcal{T})$-Kodierung von jedem solchen Wort. Die natürliche Projektion der Sprache $\mathrm{Mod}(\chi)$ auf A ist deshalb genau die diskrete Projektion von $\mathrm{L}(\mathfrak{A})$. Da $\mathrm{Mod}(\chi)$ nach 2.1.3 (Sätze von Büchi und Elgot) eine reguläre Sprache diskreter Wörter ist, ist auch jede ihrer Projektionen regulär, insbesondere die natürliche Projektion auf A.

2)(a) Wir beschreiben ein effektives Konstruktionsverfahren: Ein gegebener existentiell-monadischer $\Pi\Sigma$-Satz werde zuerst in einen äquivalenten m-normalen $\Pi\Sigma$-Satz ϕ umgewandelt. Dann werde die Menge $\mathcal{T}'$ der erweitert-atomaren Teilformeln aus ϕ bestimmt und χ wie im Beweis von 1) konstruiert. Danach wandle man χ in einen diskreten Automaten $\mathfrak{A}$ gemäß 2.1.3 (Sätze von Büchi und Elgot) um. Schließlich konstruiere man aus $\mathfrak{A}$ einen Automaten, der die natürliche Projektion von $\mathrm{L}(\mathfrak{A})$ auf A erkennt (was trivialerweise möglich ist).

3)(b) Da eine W-Sprache genau dann leer ist, wenn ihre diskrete Projektion leer ist, und das Leerheitsproblem für diskrete Automaten entscheidbar ist, folgt die Entscheidbarkeit des Leerheitsproblems für Σ-Automaten aus 3)(a). Die Entscheidbarkeit des Erfüllbarkeitsproblems für existentiell-monadische $\Pi\Sigma$-Formeln folgt aus der Entscheidbarkeit des Leerheitsproblems mit 2.8.1. ■

Teil II

Automaten und Logiken mit Zeitabhängigkeit

Kapitel 3

Zeitabhängige Automaten und ihre logische Beschreibung

Zur Beschreibung zeitabhängiger Systeme sind in den letzten Jahren eine Reihe unterschiedlicher, im weitesten Sinne als Automaten zu bezeichnende Formalismen entwickelt worden. Das erste, wohl auch einfachste und am besten studierte Modell geht auf Alur und Dill zurück ([AD90]) und ist unter der Bezeichnung ‚timed automaton' bekannt. Dabei handelt es sich um einen klassischen endlichen Automaten, der durch die Hinzunahme einer endlichen Zahl von Uhren an die Belange, die bei der Beschreibung zeitlichen Verhaltens auftreten, angepaßt wurde. Sein Studium hat mittlerweile zu einem neuen Gebiet in der Automatentheorie und der Theorie formaler Sprachen geführt, u. a. auch zu dem Begriff ‚timed regular language' (siehe [AD91] und [AD]).

Zeitgleich wurden in [Lew90] von Harry R. Lewis ‚timed state diagrams' eingeführt. Darauf folgten weitere Modelle: ‚timed transition systems' ([HMP91]), ‚interval automata' ([Alu91]), ‚parallel timer processes' (mit oder ohne ‚memory cells', [Čer92]), ‚action timed graphs' ([NSY91] bzw. [NSY93]) und ‚guarded-command real-time programs' ([HNSY92]).

Alle diese Systeme sind insofern von gleicher Konzeption, als sie mit einem wie auch immer gearteten Mechanismus versehen sind, der es erlaubt, gleichzeitig eine begrenzte Anzahl zeitlicher Abstände verfolgen und messen zu können. Es ist deshalb auch nicht verwunderlich, daß auf den ersten Blick höchst unterschiedliche Systeme dieser Art, wie z. B. ‚timed automata' und ‚timed state diagrams', gleiche Ausdrucksstärke besitzen, was in [Kro93] bewiesen wurde. Wir wollen diese Automaten unter dem Begriff Abstandsautomaten zusammenfassen.

Zur Beschreibung komplizierterer zeitlicher Zusammenhänge, in denen auch abgesehen von zeitlichen Abständen sich mit der Zeit verändernde Größen eine wichtige Rollen spielen, sind Abstandsautomaten zu schwach. An ihre Stelle treten Automaten, die allgemein als ‚hybride Automaten' bekannt sind (siehe [GNRR93]). Diese liegen genauso wie ihre einfachen Vorgänger in unterschiedlichsten Formen vor. Die

wichtigsten Modelle, die auch im Zusammenhang mit der automatischen Verifikation von hybriden Systemen studiert werden, sind ‚constant slope hybrid systems' und ‚integration graphs' (beide [KPSY93]) sowie ‚(linear) hybrid automata' ([ACHH93]).

Die zeitlichen Abhängigkeiten, die das Verhalten eines hybriden Systems bestimmen, werden in hybriden Automaten durch reelle Variablen modelliert, deren Belegungen sich (stückweise) kontinuierlich verändern. Meistens werden — wie auch in den angegebenen Fällen — nur lineare Abhängigkeiten in Betracht gezogen.

Die Vielfalt der Modelle zur Beschreibung zeitlichen Verhaltens, sowohl durch Abstandsautomaten wie auch durch hybride Automaten, vergrößert sich ständig. Da es sich um komplexe Mechanismen handelt, die in den Automaten ablaufen, ergeben sich viele Freiheiten bei der Definition, von denen reichlich Gebrauch gemacht wird. Die Situation erinnert in dieser Hinsicht an Turingmaschinen, von denen zahlreiche Varianten bekannt sind: Einband- und Mehrbandmaschinen, on-line und off-line Versionen, Maschinen mit beidseitig oder einseitig unbeschränkten Bändern, usw. Im Fall der Turingmaschinen ist man in der glücklichen Lage, sich in jedem Einzelfall davon überzeugen zu können, daß die gleiche Sprachklasse definiert wird. Zumindest für Abstandsautomaten gibt es, wie oben angedeutet wurde, Anzeichen dafür, daß sich in ähnlicher Weise ein stabiler Begriff von erkannter Sprache in Form der ‚timed regular languages' (siehe [AD90], [AD]) entwickelt. Für hybride Automaten ist die Situation verschwommener und es ist noch kein Ergebnis bekannt, daß mit dem Äquivalenzsatz aus [Kro93] vergleichbar wäre. Dennoch scheint sich ein allgemeines Verständnis von dem entwickelt zu haben, was unter einem hybriden Automaten verstanden werden sollte.

In diesem Kapitel werden die aus der Literatur bekannten Automatenmodelle für zeitabhängige Systeme in den Rahmen, der durch den Begriff des Σ-Automaten vorgegeben ist, eingeordnet. Dazu definieren wir drei geeignete Berechnungsstrukturen, Σ_{UA}, Σ_{UA^*} und Σ_{HA}, die uns zu ‚Uhrenautomaten' (Abschnitt 3.2), ‚Uhrenautomaten mit Haltemöglichkeit' (Abschnitt 3.3) bzw. ‚hybriden Automaten' (Abschnitt 3.4) führen werden. In Abschnitt 3.6 werden die Inklusionsverhältnisse zwischen den Klassen der jeweils erkannten Sprachen untersucht.

Für Uhrenautomaten ist die Äquivalenz zu ‚timed automata' offensichtlich. Für Σ_{HA}-Automaten werden wir nachweisen, daß sie zu CSHS mit sogenannten ‚single integrator tests' äquivalent sind (Abschnitt 3.5). Das Modell des Uhrenautomaten mit Haltemöglichkeit ist durch den in [Čer92] eingeführten Begriff des ‚parallel timer process with memory cells' motiviert.

Durch das Hauptergebnis des ersten Teils, des Charakterisierungssatzes, werden wir dann in die Lage versetzt, eine logische Beschreibung des Verhaltens der drei Automatenmodelle zu geben (Abschnitt 3.7).

Wir beginnen mit der Einführung des Modells des zeitlichen Verhaltens, das dieser Arbeit zugrunde liegt.

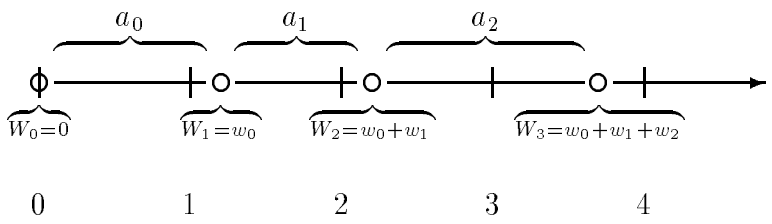


Abbildung 3.1: Graphische Illustration des Begriffes ‚Intervallwort'

3.1 Intervallwörter

Wir spezialisieren uns im gesamten Kapitel auf die Wertemenge **P** der positiven reellen Zahlen.

Jeder **P**-Buchstabe hat definitionsgemäß die Gestalt (a, w), wobei a ein Element des Alphabets und w eine positive reelle Zahl ist. Ein **P**-Buchstabe wird im folgenden als ein *Ereignis* interpretiert. Dabei verstehen wir den Namen des Buchstabens als den *Ereignistyp* und den Wert als die *zeitliche Dauer* des Ereignisses.

In diesem Sinne wird ein **P**-Wort $u = (\underline{a}, \underline{w})$ als eine Folge von Ereignissen interpretiert, die zeitlich direkt aufeinander folgen, mit a_0, a_1, a_2, ... fortlaufend bezeichnet sind und deren Dauern der Folge w_0, w_1, w_2, ... entnommen werden können. Nehmen wir an, daß das erste Ereignis zum Zeitpunkt $W_0 = 0$ beginnt, so beginnt das zweite Ereignis zum Zeitpunkt $W_1 = w_0$, während das erste Ereignis zu diesem Zeitpunkt aufhört. Das zweite Ereignis wiederum endet am Zeitpunkt $W_2 = w_0 + w_1$ und das dritte Ereignis beginnt zu diesem Zeitpunkt, usw. Die Situation ist in Abbildung 3.1 illustriert.

Generell wird $W_0 = 0$ und $W_{i+1} = W_i + w_i$ für jedes $i < |u|$ gesetzt. Jeder dieser Werte wird *beobachtbarer Zeitpunkt* oder auch *Moment* von u genannt. Die zugehörigen Indizes heißen *Positionen* (im Gegensatz zu Stellen).

Die *Gesamtdauer* eines **P**-Wortes $u = (\underline{a}, \underline{w})$ ist die Summe der Dauern aller Einzelereignisse (a_i, w_i) und wird mit $||u||$ bezeichnet: Ist u endlich von der Länge m, so ist $||u|| = W_{m+1}$; ist u unendlich, so ist $||u|| = \lim_{i\to\infty} W_i$.

3.1.1 Definition (Intervallwort) Ein *Intervallwort* ist ein **P**-Wort. Ein unendliches **P**-Wort, dessen Gesamtdauer unendlich ist, wird als *unbeschränktes Intervallwort* bezeichnet.

Mit anderen Worten, ein unbeschränktes Intervallwort hat die Eigenschaft, daß in jedes beschränkte Intervall der positiven reellen Achse jeweils nur endlich viele Momente fallen. Denkt man an die Modellierung von realen Systemen, so würde ein beschränktes unendliches Intervallwort ein chaotisches Verhalten widerspiegeln.

Offenbar ist die Zuordnung zwischen Intervallwörtern und Paaren $(\underline{a}, \underline{W})$ aus einer Folge $\underline{a}$ von Buchstaben und einer gleich langen, streng monoton steigenden Folge $\underline{W}$ mit $W_0 = 0$ eineindeutig und umkehrbar. Eingeschränkt auf unbeschränkte Intervallwörter liefert sie eine Korrespondenz zu den Paaren, bei denen die Folge $\underline{W}$ zusätzlich divergent ist. Die Gesamtheit dieser Paare bildet entsprechend der Terminologie in [AD90] die Menge der ‚timed words' über dem Alphabet A.[1] Damit sind beide Begriffe, ‚unbeschränktes Intervallwort' und ‚timed word', gleichwertig und Ergebnisse der vorliegenden Arbeit und Ergebnisse aus [AD90] und Folgearbeiten werden vergleich- und aufeinander übertragbar. So kann und wird im folgenden eine ‚language of timed words' im Sinne von [AD90] mit einer Sprache unbeschränkter Intervallwörter identifiziert und umgekehrt.

An dieser Stelle ist eine weitere Anmerkung hinsichtlich der Angemessenheit des benutzten Modells zeitlichen Verhaltens, des ‚Intervallwortes', angebracht. Zuweilen (z. B. in [AFH91], [ACHH93] oder in [ACH93]) werden Folgen von gefärbten, endlichen reellen Intervallen zur Beschreibung eines Ereignisverlaufs verwendet. Dabei übernimmt die Färbung die Rolle des Ereignistyps und das endliche reelle Intervall die Beschreibung seiner zeitlichen Ausdehnung. Benutzt man aber Intervalle (und nicht einfach nur Dauerangaben), so ist man gezwungen, auch über Intervallränder zu sprechen: Ist ein Intervall rechtsoffen, so kann das darauffolgende Intervall nur linksabgeschlossen sein, damit nicht ein Zeitpunkt verloren geht; ist umgekehrt ein Intervall rechtsabgeschlossen, so muß das nachfolgende Intervall linksoffen sein, damit kein Zeitpunkt doppelt belegt wird.

Dadurch, daß wir in unserem Modell keine Zuordnung der beobachtbaren Zeitpunkte zu Ereignissen vornehmen, vermeiden wir eine derartige Diskussion um Intervallränder und brauchen keine entsprechenden Konsistenzforderungen aufzustellen. Das heißt nicht, daß die hier vorgestellte Modellierung durch Intervallwörter soweit abstrahierte, daß die aus der Literatur bekannte Modellbildung nicht mehr im einzelnen nachahmbar wäre. Es bleibt dem „Benutzer" überlassen, eine geeignete Kodierung bzw. Interpretation vorzunehmen. So kann man z. B. das gegebene Alphabet A durch das kartesische Produkt $2 \times A \times 2$ ersetzen und erste und letzte Komponente eines komplexen Buchstabens des neuen Alphabets jeweils als „offen" bzw. „geschlossen" interpretieren und die zweite Komponente wie gewohnt als Ereignistyp. Dann muß nur noch dafür Sorge getragen werden, daß die Betrachtungen auf solche Intervallwörter beschränkt werden, die den entsprechenden Konsistenzbedingungen an die Intervallränder genügen. Das ist weiter nicht problematisch, da Forderungen dieser Art lokaler Natur sind und nicht auf die Dauern Bezug nehmen und damit allein durch Zustände (in einem Automaten) kontrolliert werden können.

[1] Das in [AD90] benutzte Modell der ‚timed state sequence' oder ‚timed observation sequence' wird auch in vielen anderen Arbeiten (u. a. schon in [KVdR83] und in [Lew90]) verwendet, allerdings unter anderen Namen, wie z. B. ‚execution' oder ‚timed path'. Vgl. auch die Bemerkungen in der Einleitung.

Formal wird dies in der folgenden Bemerkung festgehalten.

3.1.2 Bemerkung Die Sprache aller Intervallwörter $((\underline{i}, \underline{a}, \underline{j}), \underline{w})$ über dem Alphabet $2 \times A \times 2$, die der Bedingung $j_n + i_{n+1} = 1$ für alle n mit $n + 1 < |u|$ genügen, ist das $\mathbf{P}$-Urbild einer regulären Sprache diskreter Wörter.

In [AFH91] werden sogar Ereignisse der Dauer 0, sogenannte ‚transient states', zugelassen. Da keine zwei solcher Ereignisse aufgrund der Konsistenzbedingungen aufeinander folgen können (das Intervall eines transienten Zustands ist beidseitig geschlossen!), kann man durch eine leichte Modifikation der oben beschriebenen Kodierung auch diesen Aspekt modellieren.

Schließlich werden wie z. B. in [ACHH93] in manchen Arbeiten auch Verhaltensmodelle mit unendlicher Dauer, aber einer endlichen Anzahl von Ereignissen in Betracht gezogen, indem für das letzte Ereignis eine unendliche Dauer zugelassen wird. Dies läßt sich im Rahmen des hier benutzten Begriffsapparates durch einen Übergang zu $\mathbf{P}_\infty$-bewerteten Wörtern modellieren.

3.2 Uhrenautomaten

In diesem Abschnitt entwickeln wir in der Terminologie der Σ-Automaten das Automatenmodell, das von Alur und Dill als ‚timed automaton' in [AD90] eingeführt wurde und das Paradebeispiel für einen Abstandsautomaten ist. Wir wollen diesem Modell den anschaulichen Namen ‚Uhrenautomat' geben.

Wir beginnen mit der Definition der Berechnungsstruktur Σ_{UA}.

Der Speicherbereich von Σ_{UA} ist die Menge $\mathbf{P}_0$ der nichtnegativen reellen Zahlen und Null das einzige Anfangselement.

Als Testprädikate werden Teilmengen von $\mathbf{P}_0$ benutzt, die durch eine Kombination aus einer der *Vergleichsrelationen* $<$, $\leq$, $>$, $\geq$, $=$, $\neq$ und einer natürlichen Zahl beschrieben werden, die auch *Vergleichswert* genannt wird. So steht z. B. < 4 für das reelle Intervall $[0, 4)$, während $\neq 3$ für $[0, 3) \cup (3, \infty)$ steht. Die Menge der Teilmengen von $\mathbf{P}_0$, die auf diese Weise zustande kommen, werde mit $\mathcal{I}$ bezeichnet.

Die Menge der Schrittfunktionen von Σ_{UA} ist einelementig. Sie enthält nur die Additionsfunktion $\mathrm{add}\colon \mathbf{P}_0 \times \mathbf{P} \to \mathbf{P}_0$ mit $\mathrm{add}(x, y) = x + y$.

3.2.1 Definition (Berechnungsstruktur für Uhrenautomaten) Die Berechnungsstruktur Σ_{UA} ist definiert durch:

$$\Sigma_{\mathrm{UA}} \;=\; (\mathbf{P}_0, \{0\}, \{\mathrm{add}\}, \mathcal{I})\,.$$

Ist $\mathfrak{A} = (Q, R, (s, \sigma), \Delta, F)$ ein Σ_{UA}-Automat, so sind in ihm offensichtlich einige Angaben überflüssig. Da Σ_{UA} nur ein Anfangselement, nämlich Null, besitzt, ist

σ identisch Null. Außerdem gibt es in Σ_{UA} nur eine Schrittfunktion, so daß bei festem Registersatz nur eine Fortschreibung möglich ist. Schließlich kommt bei jeder Rücksetzung nur ein Wert in Frage, auf den zurückgesetzt werden kann.

Das führt uns zu der folgenden vereinfachenden Definition:

3.2.2 Definition (Uhrenautomat) Ein *Uhrenautomat* $\mathfrak{A}$ ist ein Quintupel

$$\mathfrak{A} \;=\; (Q, R, s, \Delta, F)\,,$$

bei dem Q, R und F wie bei einem Σ_{UA}-Automaten sind, $s \in Q$ *Anfangszustand* genannt wird und Δ aus Transitionen der Form (q, a, ϕ, ζ, q') besteht. Er wird mit dem Σ_{UA}-Automaten $(Q, R, (s, \sigma), \Delta', F)$ identifiziert, für dessen Anfangsbelegung $\sigma(r) = 0$ für alle $r \in R$ gilt und dessen Transitionsrelation Δ' für jede Transition (q, a, ϕ, ζ, q') aus Δ die Transition $(q, a, \theta, \phi, \zeta, q')$ enthält, wobei $\theta(r) = \mathrm{add}$ für jedes $r \in R$ gewählt wird.

Sinngemäß mögen die Begriffe *deterministischer Uhrenautomat*, *Uhrenautomat mit einfachen Transitionen* und *Uhrenautomat mit kombinierten Transitionen* verstanden werden.

Eine Sprache von Intervallwörtern wird *uhrenerkennbar* genannt, wenn sie von einem Uhrenautomaten erkannt wird. Die Klasse aller uhrenerkennbaren Sprachen wird mit **NTL** (‚non-deterministic timed languages') bezeichnet.

Mit dem Begriff ‚Uhrenautomat' hat es folgende Bewandnis: Legt man für **P**-Wörter die „Ereignis-Dauer"-Interpretation zugrunde, wie wir sie im letzten Abschnitt kennengelernt haben, so kann man jedes Register eines Σ_{UA}-Automaten als eine (Stopp-)Uhr auffassen, die jeweils um die Dauer des aktuellen Ereignisses vorläuft, die auf 0 zurückgesetzt und mit natürlichen Zahlen verglichen werden kann. Deshalb werden wir auch in Zukunft von Registern in Uhrenautomaten als *Uhren* sprechen. Dementsprechend werden Registerbelegungen auch als *Uhrenstände* bezeichnet.

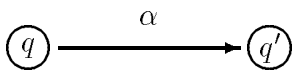
Bevor wir zum ersten konkreten Beispiel eines Register- bzw. Uhrenautomaten kommen, wollen wir uns auf die Art der graphischen Darstellung einigen. Dazu ein Einschub:

Die graphische Darstellung von Register- und Uhrenautomaten. Grundsätzlich ist zu bemerken, daß für die graphische Darstellung von Σ-Automaten im allgemeinen und Uhrenautomaten im speziellen die bekannten Darstellungen von endlichen diskreten Automaten (siehe [HU79]) Vorbild sind.[2]

[2]Damit folgen wir hier der Mehrheit der Arbeiten, die sich mit Uhrenautomaten und ähnlichen Systemen beschäftigen. Es sollte jedoch erwähnt werden, daß gerade bei der Darstellung von hybriden Automaten auch der modulare Darstellungsformalismus der ‚statecharts' benutzt wird [PK92], der ursprünglich für diskrete endliche Automaten eingeführt wurde [Har87]. Die für CSHS in [KPSY93] benutzte Darstellungsform wird in Abschnitt 3.5 eingeführt.

Ein Zustand q wird durch einen mit q beschrifteten Kreis oder ein mit q beschriftetes Oval

dargestellt, eine Transition mit Quellzustand q und Zielzustand q' durch einen beschrifteten Pfeil zwischen den Zuständen:



Um die Beschriftung α des Pfeils beschreiben zu können, nehmen wir an, daß $R = \{r_0, \ldots, r_{m-1}\}$ die Registermenge des Automaten und $(q, a, \theta, \phi, \zeta, q')$ die darzustellende Transition sei. Weiterhin wollen wir annehmen, daß die Rücksetzung ζ durch $\zeta = \{(r_{i_0}, d_0), \ldots, (r_{i_{s-1}}, d_{i_{s-1}})\}$ gegeben sei. Dann ergibt sich die Beschriftung α wie folgt:

$$\alpha \quad = \quad a\colon r_0\theta(r_0), \ldots, r_{m-1}\theta(r_{m-1});\ \phi?;\ r_{i_0} \leftarrow d_0, \ldots, r_{i_{s-1}} \leftarrow d_{i_{s-1}}\ .$$

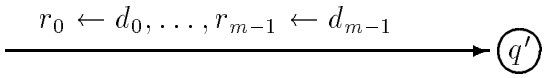
Wird anstelle des Buchstabens a der Großbuchstabe A geschrieben, dann vertritt er alle Buchstaben des Alphabets, d. h., es gibt dann für jeden Buchstaben des Alphabets eine Transition mit dem gleichen Quellzustand, dem gleichen Zielzustand, der angegebenen Fortschreibung, dem angegebenen Registertest und der angegebenen Rücksetzung.

Die Beschriftung für eine kombinierte Transition (im Sinne der Erläuterung am Ende von Abschnitt 1.5) ergibt sich in natürlicher Weise.

Aus Platzgründen wird die Beschriftung eines Pfeils zuweilen auf den Platz oberhalb und unterhalb der Linie verteilt.

Endzustände sind durch doppelte Kreise markiert:

während auf den Anfgangszustand ein Pfeil ohne Quelle zeigt, der mit einer der Anfangsbelegung entsprechenden Rücksetzung versehen ist:



Hat die Berechnungsstruktur nur ein Anfangselement, wie im Falle der Uhrenautomaten, so kann die Angabe der Anfangsbelegung entfallen. □

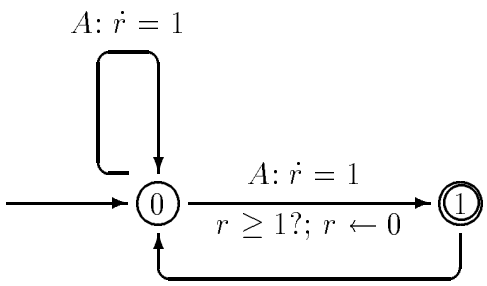


Abbildung 3.2: Ein Uhrenautomat, der die Menge aller unbeschränkten Intervallwörter erkennt

Wir fahren mit der Untersuchung von Uhrenautomaten fort und wenden uns einem ersten Beispiel zu. Es sei vorher noch angemerkt, daß die Anwendung der Schrittfunktion add auf ein Register r bei der graphischen Darstellung von Uhrenautomaten in der Form $\dot{r} = 1$ notiert wird, was seinen Grund darin hat, daß die partielle Ableitung von add nach der zweiten Komponente die konstante Funktion Eins ist: $\frac{\partial \mathbf{add}(x,y)}{\partial y} = 1$.

3.2.3 Beispiel Formal beschreibt man den Automaten aus Abbildung 3.2 durch das Quintupel $(\{0,1\}, \{r\}, 0, \Delta, \{1\})$ mit

$$\Delta \;=\; \{(0,a,0) \mid a \in A\} \cup \{(0,a,r \geq 1, \{(r,0)\}, 1) \mid a \in A\} \cup \{(1,0)\}\,.$$

Man beachte, daß hier von kombinierten Transitionen Gebrauch gemacht wird.

Offensichtlich erkennt dieser Automat, als Büchi-Automat aufgefaßt, die Menge aller unbeschränkten Intervallwörter über dem gegebenen Alphabet A. Einen äquivalenten deterministischen Büchi-Automaten erhält man, indem man die Schleife mit dem Test $r < 1$ versieht und aus der unteren Transition eine normale Transition macht. Ein äquivalenter deterministischer Muller-Uhrenautomat ist in Abbildung 3.3 zu sehen.

Aus diesem Beispiel geht hervor, daß die Menge aller unbeschränkten Intervallwörter deterministisch uhrenerkennbar ist. Daraus folgt unter Einbeziehung der Abschlußeigenschaften der Klasse der Σ_{UA}-erkennbaren Sprachen:

3.2.4 Bemerkung Ist $\mathfrak{A}$ ein Büchi-Uhrenautomat oder ein deterministischer Muller-Uhrenautomat, so ist die Sprache der von ihm akzeptierten unbeschränkten Intervallwörter uhrenerkennbar bzw. deterministisch uhrenerkennbar.

Eine Sprache unbeschränkter Intervallwörter ist also schon dann uhrenerkennbar, wenn es eine uhrenerkennbare Sprache unendlicher Intervallwörter gibt, deren unbeschränkter Anteil mit der vorgegebenen Sprache übereinstimmt.

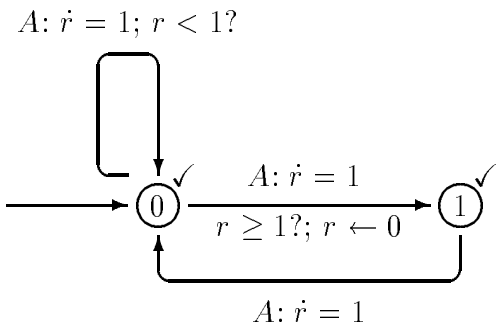


Abbildung 3.3: Ein deterministischer Uhrenautomat, der die Menge aller unbeschränkten Intervallwörter erkennt

Die beiden Häkchen an den Zuständen sollen andeuten, daß beide Zustände zu ein und derselben Endzustandsmenge gehören. D. h., das System der Endzustandsmengen soll hier $\{\{0, 1\}\}$ sein.

Der in [AD90] definierte Begriff des ‚timed Büchi automaton' stimmt mit dem Begriff des Büchi-Uhrenautomaten überein, jedoch wird jeder ‚timed Büchi automaton' in [AD90] über unbeschränkten Intervallwörtern interpretiert. Wegen 3.2.4 können wir demnach schließen:

3.2.5 Satz (timed automata) *Eine Sprache von unbeschränkten Intervallwörtern ist genau dann uhrenerkennbar, wenn sie im Sinne von [AD90] von einem ‚timed Büchi automaton' erkannt wird, d. h., wenn sie im Sinne von [AD] eine ‚timed regular language' ist.*

Damit ist eine Unterscheidung zwischen von Uhrenautomaten und von ‚timed Büchi automata' erkannten Sprachen hinfällig; im weiteren Verlauf der Arbeit wird deshalb nur noch von uhrenerkennbaren Sprachen die Rede sein.

Wir wollen uns noch eine weitere Beispielsprache ansehen.

3.2.6 Beispiel Der in Abbildung 3.4 dargestellte Uhrenautomat erkennt die Sprache aller Intervallwörter, die ausschließlich aus Ereignissen der Dauer Eins bestehen.

Hier wird das Akzeptieren, da es nur einen einzigen Zustand gibt und dieser auch Endzustand ist, nicht direkt durch die Akzeptierbedingung abgefangen, sondern dadurch, daß es überhaupt eine Berechnung gibt.

3.3 Uhrenautomaten mit Haltemöglichkeit

Durch Uhren, wie wir sie im letzten Abschnitt kennengelernt haben, kann man lediglich den zeitlichen Abstand zweier Momente bestimmen. Darüberhinaus ist man

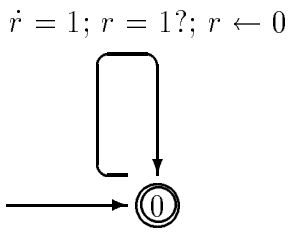


Abbildung 3.4: Ein Uhrenautomat, der die Intervallwörter erkennt, die aus Ereignissen mit Einheitsdauer bestehen

manchmal aber auch daran interessiert, die Gesamtdauer einer Reihe von Ereignissen zu bestimmen, die nicht unmittelbar aufeinander folgen. Dazu müßte man Uhren anhalten und später weiterlaufen lassen können. Das soll in diesem Abschnitt modelliert werden.

Die Idee von Uhren, die zwischenzeitlich angehalten werden dürfen, verfolgt auch Kārlis Čerāns in seiner Arbeit [Čer92], wo er von ‚parallel timer processes with memory cells' spricht. Die ‚memory cells' sind dort die Uhren, die man anhalten kann.

Zu Beginn dieses Kapitels hatten wir eine Unterscheidung zwischen einfachen Abstandsautomaten und komplizierten hybriden Automaten getroffen. Die Automaten, die wir im folgenden vorstellen werden, sind zwischen diesen beiden Modellen anzusiedeln.

Es sei $\mathrm{id}\colon \mathbf{P}_0 \times \mathbf{P} \to \mathbf{P}_0$ mit $\mathrm{id}(x, y) = x$ die *identische Schrittfunktion.*

3.3.1 Definition (Berechnungsstruktur für Uhrenautomaten mit Haltemöglichkeit) Die Berechnungsstruktur Σ_{UA^*} ist definiert durch:

$$\Sigma_{\mathrm{UA}^*} \quad = \quad (\mathbf{P}_0, \{0\}, \{\mathrm{add}, \mathrm{id}\}, \mathcal{I})\,.$$

Wie im vorigen Abschnitt bei Uhrenautomaten ist die Angabe der Anfangsbelegung überflüssig. Jedoch gibt es im allgemeinen mehrere Möglichkeiten, eine Fortschreibung zu bilden.

3.3.2 Definition (Uhrenautomat mit Haltemöglichkeit) Ein *Uhrenautomat mit Haltemöglichkeit* ist ein Quintupel

$$\mathfrak{A} \quad = \quad (Q, R, s, \Delta, F)\,,$$

bei dem Q, R, Δ und F wie bei einem Σ_{UA^*}-Automaten sind und $s \in Q$ *Anfangszustand* genannt wird. Er wird mit dem Σ_{UA^*}-Automaten $(Q, R, (s, \sigma), \Delta, F)$ identifiziert, für dessen Anfangsbelegung $\sigma(r) = 0$ für alle $r \in R$ gilt.

Die Klasse aller durch Uhrenautomaten mit Haltemöglichkeit erkennbaren Sprachen wird mit $\mathbf{NTL}^*$ bezeichnet.

Bei der graphischen Darstellung von Uhrenautomaten mit Haltemöglichkeit steht $\dot{r} = 1$ wieder für die Anwendung der Schrittfunktion add. In diesem Sinne könnten wir die Anwendung der Funktion id auf das Register r als $\dot{r} = 0$ notieren. Der Einfachheit halber vereinbaren wir aber, daß id immer dann angewendet wird, wenn keine Schrittfunktion angegeben ist. (Beachte, daß diese Vereinbarung auch mit der Semantik der ϵ-Transitionen nicht in Konflikt gerät.)

Um die Funktionsweise von Uhrenautomaten mit Haltemöglichkeit zu verstehen, wollen wir uns an einem Beispiel überlegen, wie die Menge der echten Teiler von Eins durch einen Uhrenautomaten mit Haltemöglichkeit erkannt werden kann. Es ist eine unäre Sprache L endlicher Intervallwörter gesucht, die durch einen Uhrenautomaten mit Haltemöglichkeit erkannt wird und die folgende Eigenschaft besitzt:

(*) Für jedes $n > 1$ gibt es ein Wort $\underline{w}$ in L, das mit $\frac{1}{n}$ beginnt, d. h., für das $w_0 = \frac{1}{n}$ gilt. Und falls $\underline{w}$ zu L gehört, so soll es eine natürliche Zahl n mit $n > 1$ geben, so daß $w_0 = \frac{1}{n}$ gilt.

3.3.3 Beispiel (Teiler von Eins) Sei $n > 1$ und $\delta = \frac{1}{n}$. Wir betrachten das unäre Wort u_n, das die Gestalt $u_n = v_0 v_1 \ldots v_{n-1}$ hat, wobei $v_0 = \delta$ und $v_{n-1} = \delta, 1-\delta$ sind und zusätzlich

$$v_i \quad = \quad \delta, i\delta, (n-i-1)\delta, \delta, i\delta, (n-i-1)\delta$$

für jedes i mit $0 < i < n-1$ gilt.

Dann erfüllt die Sprache $\{u_n \mid n > 1\}$ die Bedingung (*). Außerdem wird sie von dem in Abbildung 3.5 dargestellten Uhrenautomaten mit Haltemöglichkeit erkannt, was wir kurz erläutern wollen.

Die Transition von 0 nach 1 liest v_0, die beiden Transitionen zwischen 1 und 8 lesen v_{n-1}. Jedes Wort v_i mit $0 < i < n-1$ wird durch die große Schleife (von 1 nach 1) gelesen.

Der Automat ist derart konstruiert, daß in Zustand 1 alle Register bis auf r_0 und s_0 auf Null stehen. Das Register r_0 enthält dann δ und s_0 den Wert $i\delta$, wenn bislang $v_0 \ldots v_{i-1}$ gelesen wurde. Nach dem Durchlaufen des rechten Astes enthält r_2 den Wert δ und s_2 den Wert $(i+1)\delta$. Beide werden im linken Ast wieder in die Register r_0 und s_0 kopiert.

3.4 Hybride Automaten

Allgemein versteht man unter einem hybriden System ein zeitabhängiges System, dessen Beschreibungsgrößen reelle Wert annehmen und sich — abgesehen von isolierten Sprungstellen — kontinuierlich mit der Zeit ändern, was oft durch Angabe entsprechender Differentialgleichungen beschrieben wird.

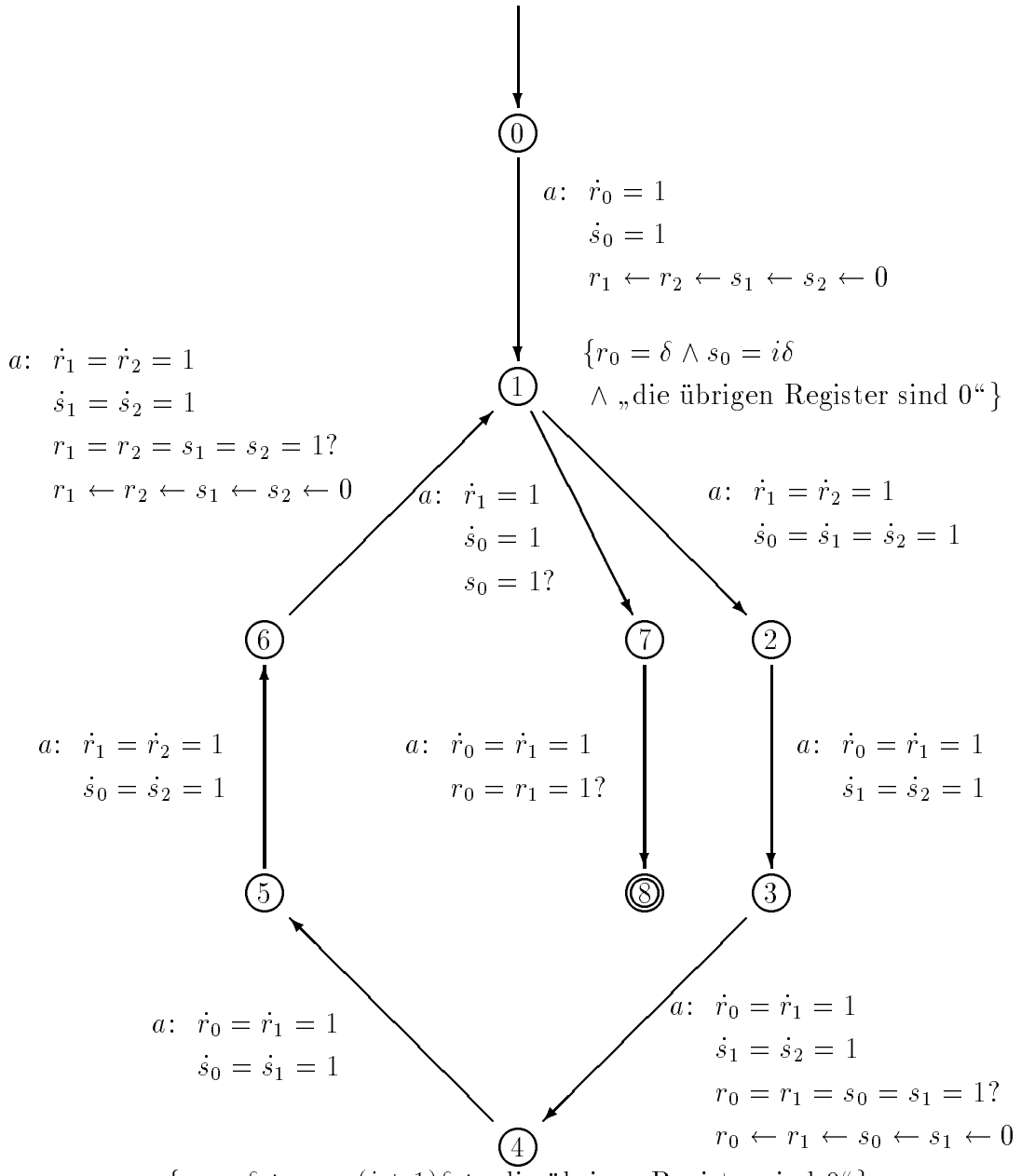


Abbildung 3.5: Ein Uhrenautomat mit Haltemöglichkeit, der die Menge der „Teiler von Eins“ erkennt

Die exakte Beschreibung der von dem Automaten erkannten Sprache und eine Erläuterung seiner Funktionsweise sind in 3.3.3 zu finden.

Bei einfachen Modellen hybrider Systeme beschränkt man sich auf konstante Differentialgleichungen, d. h., die Größen ändern sich linear mit der Zeit. Eine weitere Einschränkung auf ganzzahlige (oder rationale) Steigungsverhältnisse führt zu hybriden Automaten, wie sie z. B. in [KPSY93] und [ACHH93] zu finden sind.

Für die Entwicklung eines Modells für hybride Systeme im Rahmen des Konzeptes der Σ-Automaten definieren wir eine neue Berechnungsstruktur, die die Bezeichnung Σ_{HA} erhält.

Der Speicherbereich von Σ_{HA} sind die reellen Zahlen $\mathbf{R}$, das einzige Anfangselement ist Null. Als Testprädikate sind die reellen Intervalle (also die Teilmengen von $\mathbf{R}$) zugelassen, die sich wie in Abschnitt 3.2 als $\sim d$ schreiben lassen, wobei hier allerdings für d alle ganzen Zahlen eingesetzt werden dürfen. Die Menge dieser Testprädikate werde mit $\mathcal{J}$ bezeichnet. Für jedes $k \in \mathbf{Z}$ ist die Funktion add_k mit $\mathrm{add}_k(x, y) = x + ky$ eine Schrittfunktion von Σ_{HA}.

3.4.1 Definition (Berechnungsstruktur für hybride Automaten) Die Berechnungsstruktur Σ_{HA} ist gegeben durch:

$$\Sigma_{\mathrm{HA}} \;=\; (\mathbf{P}_0, \{0\}, \{\mathrm{add}_k \mid k \in \mathbf{Z}\}, \mathcal{J})\,.$$

Wie bei Uhrenautomaten (mit oder ohne Haltemöglichkeit) ist die Angabe der Anfangsbelegung nicht notwendig.

3.4.2 Definition (hybrider Automat) Ein *hybrider Automat* ist ein Quintupel

$$\mathfrak{A} \;=\; (Q, R, s, \Delta, F)\,,$$

bei dem Q, R, Δ und F wie bei einem Σ_{HA}-Automaten sind und $s \in Q$ *Anfangszustand* genannt wird. Er wird mit dem Σ_{HA}-Automaten $(Q, R, (s, \sigma), \Delta, F)$ identifiziert, für dessen Anfangsbelegung $\sigma(r) = 0$ für alle $r \in R$ gilt.

Die Klasse aller durch hybride Automaten erkannten Sprachen wird mit **NHL** (‚non-deterministic hybrid languages') bezeichnet.

Bei der graphischen Darstellung von hybriden Automaten wird die Anwendung einer Schrittfunktion add_k auf ein Register r im Einklang mit den Schreibweisen aus den beiden vorangehenden Abschnitten durch $\dot{r} = k$ notiert.

3.4.3 Beispiel Der in Abbildung 3.6 dargestellte hybride Automat erkennt die Sprache $\{(\mathrm{a}, 1)^n(\mathrm{b}, 1)^n \mid n > 0\}$, deren diskrete Projektion die kontextfreie, nicht reguläre Sprache $\{\mathrm{a}^n\mathrm{b}^n \mid n > 0\}$ ist.

Im nächsten Abschnitt soll nun gezeigt werden, wie sich das hier vorgestellte Modell eines hybriden Automaten zu dem Modell aus [KPSY93] verhält. Wir werden

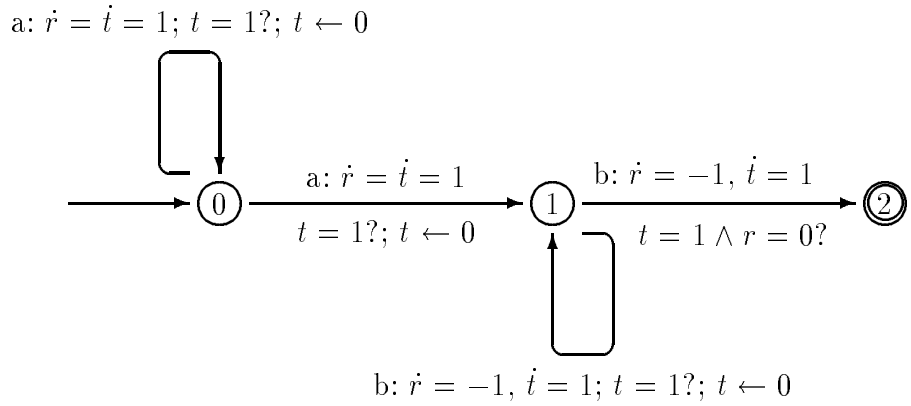


Abbildung 3.6: Ein hybrider Automat, der eine Sprache erkennt, deren diskrete Projektion nicht regulär ist

Der dargestellte hybride Automat erkennt die Sprache $\{(\mathrm{a},1)^n(\mathrm{b},1)^n \mid n > 0\}$, deren diskrete Projektion die kontextfreie, aber nicht reguläre Sprache $\{\mathrm{a}^n\mathrm{b}^n \mid n > 0\}$ ist.

zeigen können, daß beide Modelle im wesentlichen — was noch genau erläutert wird — gleichwertig sind.

Bevor wir dazu übergehen, soll noch bemerkt werden, daß ein anderes „einfaches“ Modell hybrider Systeme, das Modell des ‚fault-tolerant hybrid automaton‘, für das in [Ho93] ein semientscheidbares Verifikationsverfahren vorgestellt wurde, dagegen nicht durch das Konzept des Σ-Automaten zu fassen ist. Selbst dann nicht, wenn man zu anderen Berechnungsstrukturen übergeht. Wesentliches Merkmal der von Ho beschriebenen Systeme ist nämlich ein Nichtdeterminismus in der Auswahl der Fortschreibung, die entlang einer Transition aus unendlich — sogar überabzählbar — vielen getroffen werden kann. Die Klasse der Funktionen, aus denen ausgewählt werden kann, wird jeweils implizit durch die Angabe einer Differential(un)gleichung, etwa in der Form $0.9 \leq \dot{x} \leq 1$, beschrieben, so daß die Systeme selbst effektiv darstellbar sind.

3.5 ‚Constant slope hybrid systems‘

Das in [KPSY93] vorgestellte Modell des ‚constant slope hybrid system‘ (CSHS) weicht in seiner formalen Definition stark von dem hier vorgestellten Modell des hybriden Automaten ab. Unter anderem liegt dies daran, daß Ereignisse nicht von Transitionen gelesen werden, sondern geschehen, während sich der Automat in einem Zustand befindet. Transitionen werden dagegen in einem CSHS nur in den Momenten, die

zwischen zwei Ereignissen liegen, durchschritten. Wir geben eine Definition von CSHS in der bislang für Registerautomaten entwickelten Terminologie.

CSHS werden nur über endlichen Zustandsverläufen interpretiert. Deshalb betrachten wir in diesem Abschnitt ausschließlich endliche Intervallwörter und Rabin-Scott-Automaten. Bei diesen gehen wir o. B. d. A. davon aus, daß in den Anfangszustand keine Transition hineinführt und aus den Endzuständen keine Transitionen herausführen. Ein Zustand, der weder Anfangs- noch Endzustand ist, wird *innerer* Zustand genannt. Eine *Schleife* ist eine Transition mit gleichem Quell- wie Zielzustand.

3.5.1 Definition (Berechnungsstruktur für CSHS) Die Berechnungsstruktur Σ_{CSHS} ist gegeben durch:

$$\Sigma_{\text{CSHS}} \quad = \quad (\mathbf{R}, \mathbf{Z}, \{\text{add}_k \mid k \in \mathbf{Z}\}, \mathcal{J}) \,.$$

3.5.2 Definition (CSHS) Ein CSHS ist ein Σ_{CSHS}-Automat mit kombinierten Transitionen und folgenden Eigenschaften:

1) Jede Transition ist eine kombinierte ϵ-Transition der Gestalt (q, ϕ, ζ, q'), wobei ϕ ein Registertest und ζ eine Rücksetzung ist, oder eine Ereignistransition.

2) Von jedem inneren Zustand geht genau eine Ereignistransition aus, und jede Ereignistransition ist eine Schleife.

3) Registertests haben die Form

$$a_0 r_0 + \ldots + a_{s-1} r_{s-1} \sim d \,, \tag{3.1}$$

wenn $\{r_0, \ldots, r_{s-1}\}$ die Registermenge ist und wobei $a_0, \ldots, a_{s-1}$ ganze Zahlen sind. Sie werden als *lineare Tests* bezeichnet und mit der natürlichen Semantik versehen. (Beachte, daß diese Tests in eigentlichen Σ_{CSHS}-Automaten nicht erlaubt sind.)

Außerdem wird die folgende semantische Forderung gestellt:

4) In jeder Berechnung darf keine Schleife, die aus einer Ereignistransition gebildet wird, zweimal hinereinander durchlaufen werden.

Ein Test wie in (3.1) heißt *einfach*[3], wenn es ein i mit $i < s$ gibt, so daß $a_i = 1$ und $a_j = 0$ für jedes j mit $i \neq j$ gilt, d. h., wenn er von der Form $r_i \sim d$ ist. Einfache Tests sind also Σ_{CSHS}-Registertests im üblichen Sinne.

[3] In der Sprechweise von [KPSY93] handelt es sich um sogenannte ‚single integrator tests'.

Bedingung 4) ist wesentlich. Durch sie wird die Menge der in Frage kommenden Berechnungen eingeschränkt. Die Menge der Berechnungen eines Σ_{CSHS}-Rabin-Scott-Automaten, der den Bedingungen 1) bis 3) genügt, ist im allgemeinen echt größer als die Menge der Berechnungen, die der gleiche Automat erkennt, wenn er als CSHS interpretiert wird.

Bei der graphischen Darstellung von CSHS wird die Beschriftung der jeweils eindeutigen Schleife, die an jedem inneren Zustand angebracht ist, in den Zustand geschrieben. Anfangs- und Endzustände sind kleine, schwarz ausgefüllte Kreise.

3.5.3 Beispiel (Gasbrenner) Wir wollen das schon in [CHR91] vorkommende Beispiel des Gasbrenners betrachten, das auch in [KPSY93] zu finden ist.

Man stelle sich einen Gasbrenner vor, der entweder kein Gas ausströmen läßt oder aber leck ist. Eine solche lecke Phase soll höchstens eine Zeiteinheit andauern und nach einer solchen Phase soll frühestens 30 Zeiteinheiten später erneut eine solche Phase beginnen.

Als sicher wird der Gasbrenner erachtet, wenn er zu jedem Zeitpunkt höchstens fünf Prozent der Zeit, die seit dem Beginn des Beobachtungszeitraums vergangen ist, in der lecken Phase verbracht hat, sofern der betrachtete Zeitpunkt mindestens 60 Zeiteinheiten vom Beginn der Zeitrechnung entfernt ist.[4]

Das in Abbildung 3.7 dargestellte CSHS soll den Gasbrenner modellieren. Das System erkennt alle Phasenverläufe, die in einem unsicheren Zustand enden.

Nun wollen wir den Zusammenhang zwischen CSHS und hybriden Automaten untersuchen.

Zuerst überlegen wir uns, daß die vergrößerte Menge von Anfangselementen, die in CSHS zugelassen sind, nicht notwendig ist.

3.5.4 Lemma *Jedes* CSHS *kann effektiv in ein äquivalentes* CSHS *umgewandelt werden, in dem nur noch Null als Rücksetzwert auftritt. Sind die in einem gegebenen* CSHS *auftretenden Registertests einfach, so auch in dem konstruierten.*

Beweis. Anstelle einer Rücksetzung auf den Wert d führt man eine Rücksetzung auf Null durch, merkt sich den Wert d in der endlichen Kontrolle und bei einem nachfolgenden Registertest verrechnet man den Wert richtig.

Formal ersetze man den Zustandsraum Q eines gegebenen CSHS durch das kartesische Produkt $Q \times M^R$, wobei M die Menge der ganzen Zahlen sei, die in den Rücksetzungen und der Anfangsbelegung des Automaten vorkommen, und R für die Registermenge stehe. Dann nehme man für jede Ereignsitransition (q, a, θ, q) und für

[4] In [CHR91] werden beliebige Teilintervalle des Beobachtungszeitraums, die mindestens die Länge 60 haben, betrachtet. Wir behandeln hier die einfachere Bedingung, die auch dem in [KPSY93] beschriebenen CSHS zugrunde liegt.

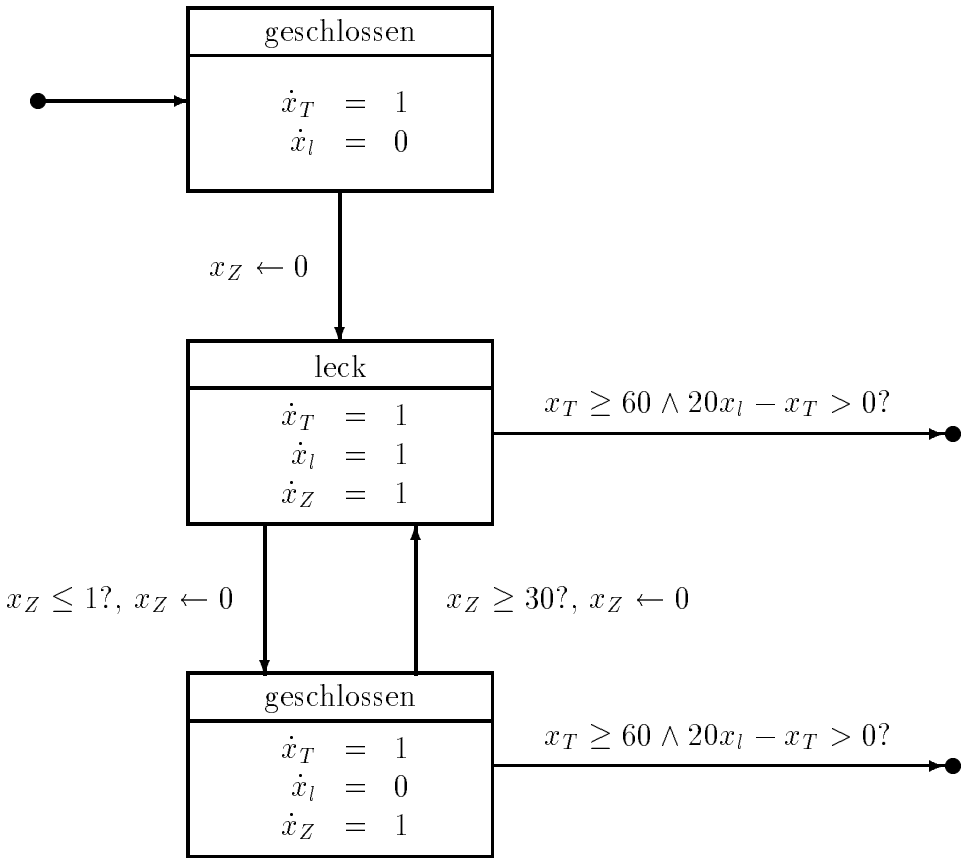


Abbildung 3.7: Ein CSHS zur Modellierung des Gasbrenners

In x_T wird die Systemzeit gemessen. Das Register x_l enthält die Gesamtzeit, die das System im Zustand „leck“ verbracht hat, während x_Z jeweils mißt, wie lange sich das System schon in seinem aktuellen Zustand befindet.

jede Funktion $f: R \to M$ die Ereignistransition $((q,f), a, \theta, (q,f))$ in die Transitionstabelle des neuen Automaten auf. Für jede kombinierte ϵ-Transition (q, ϕ, ζ, q') und jede Funktion $f: R \to M$ gehöre die Transition $((q,f), \phi', \zeta', (q', f\zeta))$ zu der neuen Transitionsrelation, wobei ζ' durch

$$\zeta'(r) \quad = \quad \begin{cases} 0\,, & \text{falls } \zeta(r) \text{ definiert ist,} \\ \text{undefiniert}, & \text{sonst,} \end{cases} \qquad \text{für } r \in R,$$

definiert und ϕ' von der Gestalt

$$a_0 r_0 + \ldots + a_{s-1} r_{s-1} \sim d + f(r_0) + \ldots f(r_{s-1})$$

sei, wenn ϕ die Form (3.1) hat. (Man beachte, daß die Elemente aus M^R auch als Registerbelegungen aufgefaßt werden können und der Ausdruck $f\zeta$ deshalb wohldefiniert ist.)

Ist (s, σ) die Anfangskonfiguration des Σ-Automaten, so sei $((s,\sigma), \{(r,0) \mid r \in R\})$ die Anfangskonfiguration des neuen Automaten. Die Endzustandsmenge des neuen Automaten sei $F \times M^R$, wenn F die Endzustandsmenge des gegebenen Automaten ist. ■

Die gleiche Konstruktion ist auch bei herkömmlichen Σ_{CSHS}-Automaten erfolgreich:

3.5.5 Bemerkung Jeder Σ_{CSHS}-Automat kann effektiv in einen äquivalenten hybriden Automaten umgewandelt werden.

Nun zeigen wir, daß CSHS mit einfachen Tests und hybride Automaten gleichwertig sind.

3.5.6 Lemma *Jeder hybride Automat läßt sich effektiv in ein äquivalentes* CSHS *umwandeln, in dem nur einfache Tests auftreten. Und umgekehrt läßt sich jedes* CSHS *mit einfachen Tests effektiv in einen hybriden Automaten umwandeln.*

Beweis. ($\Rightarrow$) O.E. nehmen wir an, daß ein hybrider Automat mit einfachen Transitionen gegeben ist.

Da jede Konjunktion von Registertests durch Hintereinanderausführen realisiert und jede Disjunktion durch Nichtdeterminismus aufgebrochen werden kann, können wir überdies davon ausgehen, daß jeder Registertest atomar oder die Negation eines atomaren Registertests ist. (Dies gilt allgemein für Σ-Automaten mit einfachen Transitionen.) Da aber das Komplement (bzgl. $\mathbf{R}$) eines jeden Elementes von $\mathcal{J}$ wieder zu $\mathcal{J}$ gehört, kann schließlich jeder Registertest als atomar angenommen werden. Und jeder atomare Registertest ist auch ein einfacher Test.

Eine Testtransition (q, ϕ, q') kann auch als kombinierte ϵ-Transition in der Form $(q, \phi, \emptyset, q')$ geschrieben werden, genau wie jede Rücksetztransition (q, ζ, q') trivialerweise durch $(q, \mathtt{tt}, \zeta, q')$ ersetzt werden kann. Damit ist die Forderung 3.5.2.1) erfüllt.

Um 3.5.2.2) zu erfüllen, ersetzt man zuerst jede Ereignistransition (q, a, θ, q') durch drei Transitionen:

$$(q, \mathtt{tt}, \emptyset, q_0)(q_0, a, \theta, q_0)(q_0, \mathtt{tt}, \emptyset, q') \,,$$

wobei q_0 ein jeweils neuer Stützzustand sei. Die Stützzustände erfüllen Bedingung 3.5.2.2) dann offensichlich. Von den inneren Zuständen des ursprünglichen Automaten gehen keine Buchstabentransitionen mehr aus. Deshalb wird an diese eine „Leerschleife" gelegt, die nicht durchlaufen werden soll. Um dies zu garantieren, wird ein neues Register r eingeführt, auf das in Leerschleifen die Schrittfunktion add_1 angewendet wird und in allen anderen add_0. Dann muß zum Schluß nur getestet werden, ob der Wert von r gleich Null ist, um festzustellen, ob auch keine Leerschleife benutzt wurde. Dafür wird jeder Endzustand zu einem inneren Zustand des neuen Automaten und erhält eine Leerschleife. Außerdem wird ein neuer Endzustand eingeführt, in den von den alten Endzuständen eine kombinierte ϵ-Transition mit Test $r = 0$ führt.

Damit die semantische Bedingung 3.5.2.4) nicht zu einer Einschränkung führt, verdoppelt man alle inneren Zustände, so daß ein mehrfaches Durchlaufen der gleichen Ereignistransitition durch abwechselndes Durchlaufen zweier Ereignistransitionen simuliert werden kann. D. h., man ersetzt die Schleife (q, a, θ, q) einfach durch die Transitionen

$$(q, a, \theta, q),\ (q', a, \theta, q'),\ (q, \mathtt{tt}, \emptyset, q') \text{ und } (q', \mathtt{tt}, \emptyset, q)$$

mit einem jeweiligen neuen Zustand q'.

($\Leftarrow$) Nach 3.5.4 können wir davon ausgehen, daß in einem gegebenen CSHS nur Null als Rücksetzwert auftritt. Das mehrfache Durchlaufen einer Ereignistransition (vgl. 3.5.2.4)) kann in der endlichen Kontrolle überwacht und verhindert werden. ■

3.5.7 Korollar (CSHS) *Eine Sprache von endlichen Intervallwörtern wird genau dann von einem hybriden Automaten erkannt, wenn sie von einem* CSHS *mit einfachen Tests im Sinne von [KPSY93] erkannt wird.*

Es stellt sich die Frage, ob CSHS mit beliebigen linearen Tests ausdrucksstärker als hybride Automaten sind. Beiläufig erwähnen die Autoren von [KPSY93], dies sei nicht der Fall. Ihrer Meinung nach kann man sich nämlich auf einfache Tests beschränken [KPSY93, S. 188], was angesichts des folgenden Beispiels fraglich ist.

3.5.8 Beispiel Das in Abbildung 3.8 dargestellte CSHS erkennt die Sprache aller unären endlichen Intervallwörter $\underline{w} = w_0 \dots w_{2n-1}$, die

$$\sum_{i=0}^{j} w_{2i} + w_{2j+1} \;=\; 1 \qquad \text{für } j < n \tag{3.2}$$

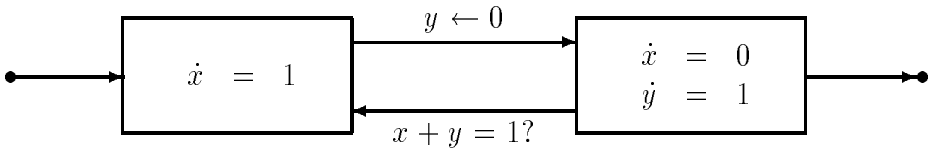


Abbildung 3.8: Illustration zu einer Behauptung in [KPSY93]

Das dargestellte CSHS erkennt die Sprache aus 3.5.8, die belegen soll, daß CSHS nicht mit einfachen Tests auskommen können.

erfüllen.

Das informelle Argument, daß den Autor zu der Vermutung veranlaßt, daß die angegebene Sprache nicht durch einen hybriden Automaten erkannt werden kann, ist wie folgt: Die Teilsummen, die auf der linken Seite von (3.2) stehen, können offensichtlich nicht auseinander hervorgehen, wenn nicht gerade die Werte von w_{2j+1} glücklich ausfallen. Also müßte zur Überprüfung von (3.2) für jedes $j < n$ ein anderes Register zuständig sein. Es gibt in einem festen Automaten aber nur endlich viele Register.

Der lineare Test $20x_l - x_T > 0$, der in dem CSHS aus 3.5.3 vorkommt, ist nicht einfach. Dennoch kann er durch einen einfachen Test ersetzt werden. Man benutzt ein Register, das ständig den Wert $20x_l - x_T$ vorhält, und testet dieses Register anstelle des ganzen Ausdrucks. Entscheidend für den Erfolg dieser Konstruktion ist die Tatsache, daß die in dem Test beteiligten Register x_l und x_T nie zurückgesetzt werden.[5]

Hat man diese Veränderung durchgeführt, so erhält man ein CSHS mit einfachen Tests, zu dem es gemäß 3.5.6 einen äquivalenten hybriden Automaten geben müßte. Ein solcher ist in Abbildung 3.7 zu finden.

3.6 Inklusionsbeziehungen

Da die Berechnungsstruktur Σ_{UA} eine Unterstruktur von Σ_{UA^*} ist, wird jede uhrenerkennbare Sprache auch von einem Uhrenautomaten mit Haltemöglichkeit erkannt. Jeden Uhrenautomaten mit Haltemöglichkeit kann man aber auch als hybriden Automaten auffassen, wenn man die Schrittfunktionen id und add mit add_0 bzw. add_1 identifiziert.

Daraus ergeben sich die folgenden Inklusionsverhältnisse:

3.6.1 Bemerkung Es gilt

$$\mathbf{NTL} \subseteq \mathbf{NTL}^* \subseteq \mathbf{NHL}\,.$$

[5]Das hier beschriebene Verfahren stammt aus [KPSY93] und soll allgemein anwendbar sein. Wie man sich leicht überzeugt, führt es aber in 3.5.8 nicht zum Erfolg.

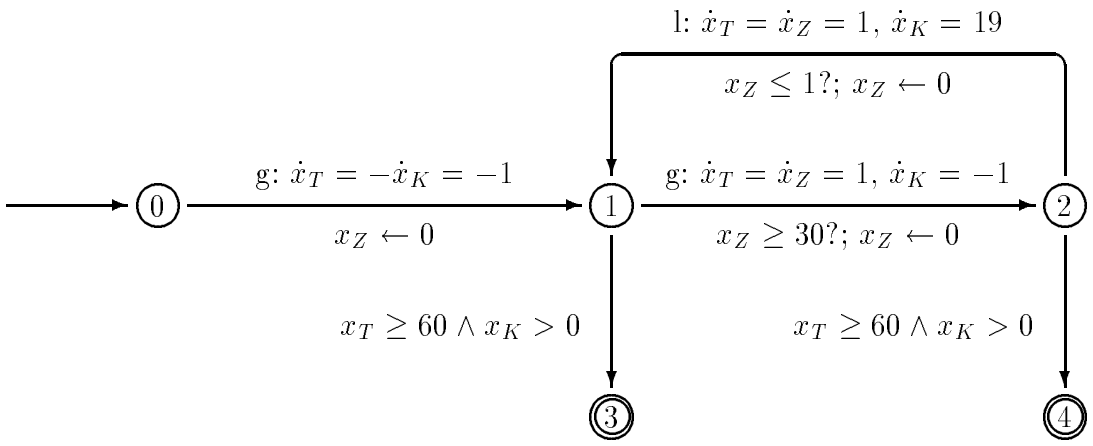


Abbildung 3.9: Ein hybrider Automat zur Modellierung des Gasbrenners

Der dargestellte hybride Automat erkennt die gleiche Sprache wie das CSHS in Abbildung 3.7. Der lineare Test $20x_l - x_T > 0$ ist hier durch $x_K > 0$ ersetzt worden, wobei dafür gesorgt wird, daß x_K jeweils den Wert des Ausdrucks $20x_l - x_T$ beinhaltet. Das Register x_l selbst ist dann überflüssig. In der Abbildung steht g für „geschlossen“ und l für „leck“.

Es soll bewiesen werden, daß die Inklusionen echt sind.

Beide Beweise nutzen die Tatsache, daß die Menge $\mathcal{I}$ (bzw. $\mathcal{J}$) der verwendeten Testprädikate eine gewisse „Grobheit“ besitzt.

Auf $\mathbf{P}_0$ definieren wir die Äquivalenzrelation $\equiv$, die zwei Zahlen x und y als äquivalent betrachtet, wenn für jede natürliche Zahl k die beiden folgenden Bedingungen erfüllt sind: $x \leq k$ gdw. $y \leq k$ und $x \geq k$ gdw. $y \geq k$. Registerbelegungen ρ und ρ' sind äquivalent, in Zeichen $\rho \equiv \rho'$, wenn $\rho(r) \equiv \rho'(r)$ für jedes Register r gilt.

Mit diesen Bezeichnungen läßt sich nun die „Grobheit“ der Testprädikate ausdrücken.

3.6.2 Bemerkung Ist ϕ ein Registertest und sind ρ und ρ' Registerbelegungen mit $\rho \equiv \rho'$, so erfüllt ρ den Test ϕ genau dann, wenn ρ' den Test ϕ erfüllt.

3.6.3 Lemma *1) Es gilt* $\{w_0w_1w_2 \mid w_0 + w_2 = 1\} \in \mathbf{NTL}^* \setminus \mathbf{NTL}$.

2) Es gilt $\{w_0w_1 \mid w_0 = w_1\} \in \mathbf{NHL} \setminus \mathbf{NTL}^*$.

Beweis. 1) Ein Uhrenautomat mit Haltemöglichkeit, der die angegebene Sprache erkennt, ist Abbildung 3.10 zu entnehmen.

Nehmen wir an, es gäbe einen Uhrenautomaten, der die Sprache erkennen würde.

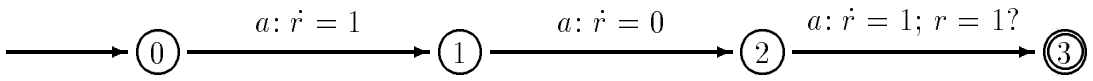


Abbildung 3.10: Ein Uhrenautomat mit Haltemöglichkeit, der $\mathbf{NTL} \subsetneqq \mathbf{NTL}^*$ belegt

Dargestellt ist ein Uhrenautomat mit Haltemöglichkeit, der die Sprache $\{w_0w_1w_2 \mid w_0 + w_2 = 1\}$ erkennt, die nicht uhrenerkennbar ist.

Dann würde der Automat das Wort 0.2,0.5,0.8 [6] akzeptieren und es gäbe eine akzeptierende Berechnung der Form

$$(q_0, \rho_0) \xrightarrow[(a,0.2)]{\delta_0} (q_1, \rho_1) \xrightarrow[(a,0.5)]{\delta_1} (q_2, \rho_2) \xrightarrow[(a,0.8)]{\delta_2} (q_3, \rho_3)\,.$$

Wir wollen zeigen, daß dann durch $\underline{\delta}$ auch eine akzeptierende Berechnung für das Wort 0.2,0.5,0.9 gegeben wäre.

Um das zu zeigen, nehmen wir weiter an, daß δ_2 von der Form $(q_2, a, \phi, \zeta, q_3)$ wäre. Wir betrachten die Registerbelegungen ρ und ρ' mit $\rho(r) = \rho_2(r) + 0.8$ und $\rho'(r) = \rho(r) + 0.9$. Es würde dann ρ den Test ϕ erfüllen. Für jede Uhr r könnte $\rho_2(r)$ nur die Werte 0, 0.5 und 0.7 annehmen, also ρ die Werte 0.8, 1.3 und 1.5, entsprechend ρ' die Werte 0.9, 1.4 und 1.6. Damit würde nach 3.6.2 auch ρ' den Test ϕ erfüllen und wir dürften

$$(q_0, \rho_0) \xrightarrow[(a,0.2)]{\delta_0} (q_1, \rho_1) \xrightarrow[(a,0.5)]{\delta_1} (q_2, \rho_2) \xrightarrow[(a,0.9)]{\delta_2} (q_3, \rho'\zeta)$$

schreiben, was implizieren würde, daß 0.2,0.5,0.9 zu der Sprache gehören würde, was nicht der Fall ist. Widerspruch!

2) Einen hybriden Automaten, der die angegebene Sprache erkennt, findet man in Abbildung 3.11.

Mit ähnlichen Argumenten wie im Beweis von 1), über die Fallunterscheidung nach den möglichen Registerwerten und unter Benutzung von 3.6.2, zeigt man, daß jeder Uhrenautomat mit Haltemöglichkeit, der 0.2,0.2 akzeptieren würde, auch 0.2,0.3 akzeptieren müßte. Widerspruch! ■

3.6.4 Korollar (Inklusionsbeziehungen) *Es gilt*

$$\mathbf{NTL} \subsetneqq \mathbf{NTL}^* \subsetneqq \mathbf{NHL}\,.$$

Damit sind die Inklusionsverhältnisse zwischen den Sprachfamilien der uhrenerkennbaren Sprachen, der Sprachen, die durch Uhrenautomaten mit Haltemöglichkeit erkannt werden, und solchen, die durch hybride Automaten erkannt werden, geklärt.

[6] In dieser Arbeit wird das im Deutschen gebräuchliche Dezimalkomma durch den Dezimalpunkt ersetzt, damit Folgen wie die angegebene lesbar bleiben.

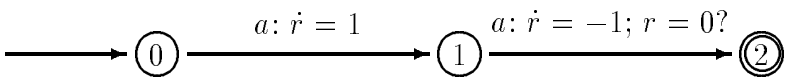


Abbildung 3.11: Ein hybrider Automat, der $\mathbf{NTL}^* \subsetneqq \mathbf{NHL}$ belegt

Dargestellt ist ein hybrider Automat, der die Sprache $\{w_0 w_1 \mid w_0 = w_1\}$ erkennt, die weder von einem Uhrenautomaten noch von einem Uhrenautomaten mit Haltemöglichkeit erkannt wird.

3.7 Logische Beschreibung durch Abstands-, Dauer- und Integralformeln

Die Einordnung der bislang vorgestellten Automatenmodelle zur Beschreibung zeitabhängiger Systeme in den Rahmen, der durch den Begriff des Σ-Automaten vorgegeben ist, ermöglicht eine Anwendung des Hauptergebnisses aus dem ersten Teil der Arbeit, des Charakterisierungssatzes. Wir gewinnen kanonische Logiken für die Beschreibung des Verhaltens von Uhrenautomaten (mit oder ohne Haltemöglichkeit) und hybriden Automaten: Eine Sprache von Intervallwörtern ist genau dann uhrenerkennbar, wenn sie durch einen existentiell-monadischen $\Pi\Sigma_{\mathrm{UA}}$-Satz definiert wird, sie wird genau dann von einem Uhrenautomaten mit Haltemöglichkeit erkannt, wenn sie durch einen existentiell-monadischen $\Pi\Sigma_{\mathrm{UA}^*}$-Satz definierbar ist, und sie wird genau dann von einem hybriden Automaten erkannt, wenn sie durch einen existentiellmonadischen $\Pi\Sigma_{\mathrm{HA}}$-Satz definiert wird.

Anstelle der in den kanonischen Signaturen $\Pi\Sigma_{\mathrm{UA}}$, $\Pi\Sigma_{\mathrm{UA}^*}$ und $\Pi\Sigma_{\mathrm{HA}}$ vorkommenden erweiterten Prädikate wollen wir andere, in gewissem Sinne äquivalente erweiterte Prädikate verwenden, die für die spätere Anwendung von Vorteil sind. Insbesondere werden die Formeln durch Benutzung der neuen erweiterten Prädikate lesbarer.

3.7.1 Abstandsformeln und Uhrenautomaten

Wir beginnen mit einer Untersuchung von $\Pi\Sigma_{\mathrm{UA}}$.

Definitionsgemäß enthält $\Pi\Sigma_{\mathrm{UA}}$ für jede natürliche Zahl l, für jede mit den Teilmengen von l indizierte Familie F von Schrittfunktionen aus $\{\mathrm{add}\}$, jede mit den Teilmengen von l indizierte Familie von Elementen aus $\{0\}$ und jedes Element $I \in \mathcal{I}$ ein erweitertes Symbol $\mathrm{E}_{I,F,H}$. Dessen Bedeutung in einem Intervallwort $u = (\underline{a}, \underline{w})$ war in Abschnitt 2.3 vermöge der Funktion $\mathrm{E}^u_{F,H}$ erklärt worden. Für $\mathrm{E}^u_{F,H}$ ergibt sich für jede l-gliedrige Folge $\underline{M}$ von Mengen $M_i \subseteq |u|$ und m, n mit $m \leq n$ und $m, n \in |u| \cup \{-1\}$:

$$\mathrm{E}^u_{F,H}(M_0, \ldots, M_{l-1}, m, n) \;=\; \sum_{i=m+1}^{n} w_i \;=\; W_{n+1} - W_{m+1}\,, \tag{3.3}$$

d. h., $\mathrm{E}^u_{F,H}(M_0, \ldots, M_{l-1}, m, n)$ ist der zeitliche Abstand des Zeitpunktes an Position $m+1$ von dem Zeitpunkt an Position $n+1$. Dafür wollen wir einfach $\mathrm{d}^u(m+1, n)$ schreiben. Insbesondere stellen wir fest, daß die Mengen $M_0, \ldots, M_{l-1}$ keinerlei Bedeutung haben. Ist das Intervall I durch $\sim c$ gegeben, dann gilt wegen (3.3):

$$(u, \nu) \models \mathrm{E}_{I,F,H}(X, \underline{X}, x) \quad \text{gdw.} \quad \mathrm{d}^u(\mathrm{vg}(\nu(x), \nu(X)) + 1, \nu(x)) \sim c\,. \tag{3.4}$$

Von $\Pi\Sigma_{\mathrm{UA}}$ gehen wir nun zu der Signatur $\Pi\overleftarrow{\mathrm{d}}$ über.

3.7.1 Definition (Signatur $\Pi\overleftarrow{\mathrm{d}}$) Die Signatur $\Pi\overleftarrow{\mathrm{d}}$ entsteht aus Π durch Hinzunahme eines erweiterten Symbols $\overleftarrow{\mathrm{d}}_{\sim c}$ des Typs $(1,1)$ für jede natürliche Zahl c und jede Vergleichsrelation $\sim$.

Aus jedem Intervallwort $u = (\underline{a}, \underline{w})$ erhalten wir eine $\Pi\overleftarrow{\mathrm{d}}$-Struktur $u^{\Pi\overleftarrow{\mathrm{d}}}$, indem wir wie üblich $u^{\Pi\overleftarrow{\mathrm{d}}} \restriction \Pi = \underline{a}^{\Pi}$ setzen und außerdem festlegen, daß (M, m) genau dann zu der Interpretation von $\overleftarrow{\mathrm{d}}_{\sim c}$ in $u^{\Pi\overleftarrow{\mathrm{d}}}$ gehören soll, wenn $\mathrm{d}^u(\mathrm{vg}(m, M) + 1, m) \sim c$ gilt. Für $\overleftarrow{\mathrm{d}}_{\sim c}(X, x)$ schreiben wir deshalb auch:

$$\mathrm{d}(X, x) \quad \sim \quad c \tag{3.5}$$

und bezeichnen solche Formeln als *relative Abstandsformeln*. Um auszudrücken, daß die Mengenvariable in (3.5) in der ersten Komponente auftritt, sprechen wir auch von *Vergangenheitsformeln*.

Monadische $\Pi\overleftarrow{\mathrm{d}}$-Formeln mögen nur über Wortstrukturen interpretiert werden, die auf die angegebene Weise aus Intervallwörtern entstehen.

Dann ergibt sich aus dem (allgemeinen) Charakterisierungssatz 2.3.2 und (3.4) zusammen mit 3.2.5:

3.7.2 Spezieller Charakterisierungssatz für Uhrenautomaten *Für eine Sprache L von Intervallwörtern sind die folgenden Aussagen äquivalent:*

1) L ist uhrenerkennbar.

2) L ist definierbar durch einen existentiell-monadischen $\Pi\overleftarrow{\mathrm{d}}$-Satz.

Ist L eine Sprache von unbeschränkten Intervallwörtern, so tritt die folgende Äquivalenz hinzu:

3) L ist eine ‚timed regular language' im Sinne von [AD]. ■

Die jeweiligen Umwandlungen von Formeln in Automaten und umgekehrt können effektiv erfolgen.

Hinsichtlich des Registerindexes einer uhrenerkennbaren Sprache treffen wir die folgende Festlegung: Der *d-Rang* einer $\Pi\overleftarrow{\mathrm{d}}$-Formel sei die Anzahl der in der Formel vorkommenden erweitert-atomaren Formeln, wobei unterschiedliche Vergleichswerte und unterschiedliche Vergleichsrelationen nicht zählen. Der d-Rang einer uhrenerkennbaren Sprache sei der minimale d-Rang eines die Sprache definierenden existentiell-monadischen $\Pi\overleftarrow{\mathrm{d}}$-Satzes.

Dann gewinnen wir aus 2.6.2:

3.7.3 Bemerkung Ist L eine uhrenerkennbare Sprache, so stimmen d-Rang und Registerindex überein.

Zum Abschluß dieses Unterabschnitts wollen wir eine konkrete existentiell-monadische $\Pi\overleftarrow{\mathrm{d}}$-Formel angeben.

3.7.4 Beispiel (3.2.3 fortgef.) Die Sprache aller unbeschränkten Intervallwörter ist duch die existentiell-monadische $\Pi\overleftarrow{\mathrm{d}}$-Formel

$$\exists X \forall x \exists y (x < y \wedge Xy \wedge \mathrm{d}(X, y) > 1)$$

definiert.

Die Formel fordert die Existenz einer unendlichen Menge, in der immer wieder zwei aufeinanderfolgende Elemente einen Abstand haben, der größer als Eins ist.

3.7.2 Integralformeln und hybride Automaten

Wir beginnen mit der Definition einer neuen Signatur.

3.7.5 Definition (Signatur $\Pi\int$) Die Signatur $\Pi\int$ entsteht aus Π durch Hinzunahme eines erweiterten Symbols $\mathrm{I}_{\underline{c},\sim d}$ des Typs $(l+1, 1)$ für jede positive natürliche Zahl l, jede l-gliedrige Folge $\underline{c}$ von ganzen Zahlen, jede Vergleichsrelation $\sim$ und jede ganze Zahl d.

Mit jedem Intervallwort u wird genau eine $\Pi\int$-Struktur $u^{\Pi\int}$ assoziiert. Dabei ist das Universum von $u^{\Pi\int}$ wie in Abschnitt 2.3 die Menge $|u|$ und die Symbole aus Π erhalten die gleiche Interpretation, d. h., das Π-Redukt $u^{\Pi\int} \restriction \Pi$ stimmt mit $\underline{a}^{\Pi}$ überein. Es ist also nur noch zu beschreiben, wie die erweiterten Symbole in $u^{\Pi\int}$ interpretiert werden.

Für jede Wahl einer l-gliedrigen Folge $\underline{c}$ ganzer Zahlen wird eine Funktion

$$\mathrm{I}^u_{\underline{c}} \colon \wp(|u| \cup \{-1\})^l \times \{(m,n) \in (|u| \cup \{-1\})^2 \mid m \leq n\}$$

definiert. Ist $\underline{M}$ eine l-gliedrige Folge von Mengen $M_i \subseteq |u|$ und ist χ_i die charakteristische Funktion von M_i für jedes $i < l$, so wird

$$\mathrm{I}^u_{\underline{c}}(\underline{M}, m, n) \quad = \quad \sum_{i=m+1}^{n} w_i \sum_{j<l} c_j \chi_j(i)$$

gesetzt. Als einfache und anschauliche Schreibweise für $\mathrm{I}^{u}_{\underline{c}}(\underline{M}, m, n)$ wollen wir

$$\int_m^n c_0 M_0 + \ldots + c_{l-1} M_{l-1} du$$

benutzen.

Die Interpretation des Symbols $\mathrm{I}_{\underline{c},\sim d}$ in $u^{\Pi\int}$ enthalte alle Tripel (N, M, n), für die

$$\int_{\mathrm{vg}(n,N)}^{n} c_0 M_0 + \ldots + c_{l-1} M_{l-1} du \quad \sim \quad d$$

gilt. Für die erweitert-atomare Formel $\mathrm{I}_{\underline{c},\sim d}(X, X_0, \ldots, X_{l-1}, x)$ schreiben wir deshalb auch

$$\int_X^x c_0 X_0 + \ldots + c_{l-1} X_{l-1} \quad \sim \quad d\,.$$

Wir nennen diese Formeln auch *relativierte Integralformeln.*

Es ergibt sich der folgende fundamentale Zusammenhang:

3.7.6 Spezieller Charakterisierungssatz für hybride Automaten *Für eine Sprache L von Intervallwörtern sind die folgenden Aussagen äquivalent:*

1) L wird von einem hybriden Automaten erkannt.

2) L wird durch einen existentiell-monadischen $\Pi\int$-Satz definiert.

Ist L eine Sprache von endlichen Intervallwörtern, so tritt die folgende äquivalente Aussage hinzu:

3) L wird von einem CSHS *mit einfachen Tests erkannt.*

Die jeweiligen Umwandlungen von Formeln in Automaten und umgekehrt können effektiv erfolgen.

Beweis. 1)⇒2) Wegen des Charakterisierungssatzes reicht es, zu zeigen, daß jeder existentiell-monadische $\Pi\Sigma_{\mathrm{HA}}$-Satz effektiv in einen äquivalenten existentiell-monadischen $\Pi\int$-Satz umgewandelt werden kann.

Dazu nehmen wir an, daß $\phi \triangleq \exists X_0 \ldots \exists X_{m-1} \phi_0$ ein existentiell-monadischer m-normaler $\Pi\Sigma_{\mathrm{HA}}$-Satz sei. Wir führen für jede Teilmenge $P \subseteq m$ eine zusätzliche Mengenvariable Z_P ein, die in einer Interpretation $(u^{\Pi\int}, \nu)$ mit der Menge

$$\bigcap_{i \in P} \nu(X_i) \cap \bigcap_{i \in m \setminus P} \sim \nu(X_i)$$

belegt werden soll. Das sichert die Formel

$$\alpha \triangleq \forall x \bigwedge_{P\subseteq m} (\bigwedge_{i\in P} X_i x \leftrightarrow Z_P x \wedge \bigwedge_{i\in m\setminus P} X_i x \leftrightarrow \neg Z_P x)\,.$$

Den zu ϕ äquivalenten $\Pi\int$-Satz wählen wir wie folgt:

$$\exists X_0 \ldots \exists X_{m-1} \exists \underline{Z}(\alpha \wedge \phi_0')\,.$$

Dabei entstehe ϕ_0' aus ϕ_0 durch Substitution der erweitert-atomaren Formeln nach folgendem Verfahren: Jede Formel $E_{C,F,H}(X_i, \underline{X}, x)$ mit $C = \{r \in \mathbf{R} \mid r \sim d\}$ und $f_P = \mathrm{add}_{c_P}$ für jedes $P \subseteq m$ werde durch

$$\int_{X_i}^{x} \sum_{P\subseteq m} c_P Z_P \quad \sim \quad d$$

ersetzt.

2)⇒1) Offensichtlich sind die erweitert-atomaren Formeln

$$\int_{X}^{x} c_0 X_0 + \ldots + c_{l-1} X_{l-1} \quad \sim \quad d$$

und

$$E_{\{r\in R\mid r\sim d\},F,H}(X, \underline{X}, x)$$

äquivalent, wenn man $f_P = \sum_{i\in P} c_i$ und $h_P = 0$ für jedes $P \subseteq l$ wählt. Deshalb läßt sich jeder existentiell-monadische $\Pi\overleftarrow{\mathrm{d}}$-Satz effektiv in einen existentiell-monadischen $\Pi\Sigma_{\mathrm{HA}}$-Satz umwandeln. Daraus ergibt sich mit dem (allgemeinen) Charakterisierungssatz die fragliche Implikation.

1)⇔3) Dies ist die Aussage von 3.5.7. ■

Auch für den Fall der hybriden Automaten soll beispielhaft eine schon bekannte, von einem hybriden Automaten erkennbare Sprache durch eine $\Pi\int$-Formel beschrieben werden.

3.7.7 Beispiel (3.4.3 fortgef.) Die Sprache $\{(\mathrm{a},1)^n(\mathrm{b},1)^n \mid n > 0\}$ wird durch die nachfolgende existentiell-monadische $\Pi'\int$-Formel mit Mengentermen beschrieben:

$$\exists x \forall y((y < x \to \mathrm{P_a} y) \wedge (x < y \to \mathrm{P_b} y))$$
$$\wedge\ \forall x \int_{\{x\mid \mathtt{tt}\}}^{x} 1\{x \mid \mathtt{tt}\} = 1 \wedge \int_{\emptyset}^{\max} 1\mathrm{P_a} + (-1)\mathrm{P_b} = 0\,.$$

Das erste Konjunktionsglied fordert die Existenz einer Stelle, links von der nur a auftritt und rechts von der nur b auftritt. Damit ist für die richtige Reihenfolge der

Buchstaben Sorge getragen. Das zweite Konjunktionsglied sorgt dafür, daß alle Ereignisse von Einheitsdauer sind, während das dritte sichert, daß die Anzahl der vorkommenden a-Ereignisse gleich der Anzahl der vorkommenden b-Ereignisse ist, indem die Gesamtdauer der a-Ereignisse von der Gesamtdauer der b-Ereignisse abgezogen und das Ergebnis auf Null überprüft wird.

3.7.3 Dauerformeln und Uhrenautomaten mit Haltemöglichkeit

Aus dem allgemeinen Charakterisierungssatz 2.3.2 wissen wir, daß eine Sprache von Intervallwörtern genau dann von einem Uhrenautomaten mit Haltemöglichkeit erkannt wird, wenn sie durch einen existentiell-monadischen $\Pi\Sigma_{\mathrm{UA}^*}$-Satz definiert wird.

Bei genauerer Betrachtung des Beweises von 3.7.6 stellt man fest: Jede existentiell-monadische $\Pi\Sigma_{\mathrm{UA}^*}$-Formel läßt sich effektiv in eine äquivalente existentiell-monadische $\Pi\int$-Formel umwandeln, in der nur Integralformeln der Gestalt

$$\int_X^x 1Y \quad \sim \quad c \tag{3.6}$$

vorkommen. Umgekehrt läßt sich auch zu jeder existentiell-monadischen $\Pi\int$-Formel mit Integralformeln wie in (3.6) effektiv ein äquivalenter existentiell-monadischer $\Pi\Sigma_{\mathrm{UA}^*}$-Satz finden.

Formel (3.6) drückt aus: „Die Gesamtdauer der Ereignisse, die zwischen dem letzten Ereignis vor x, das zu X gehört, und x liegen und zu Y gehören, erfüllt die Beziehung $\sim c$.“ Deshalb schreiben wir für (3.6) auch

$$\mathrm{dur}(X, Y, x) \quad \sim \quad c\,.$$

Diese Formeln wollen wir *relative Dauerformeln* nennen.

Die angestellten Überlegungen führen zu der folgenden Definition.

3.7.8 Definition (Signatur Πdur) Die Signatur Πdur entsteht aus Π durch Hinzunahme eines erweiterten Symbols $\mathrm{D}_{\sim c}$ des Typs $(2,1)$ für jede Vergleichsrelation $\sim$ und jede natürliche Zahl c.

Wie in den vorangehenden beiden Unterabschnitten möge für jedes Intervallwort u eine entsprechende Πdur-Wortstruktur $u^{\Pi\mathrm{dur}}$ definiert werden, wobei $\mathrm{D}_{\sim c}(X, Y, x)$ die Bedeutung von (3.6) erhalten soll. Dann gilt:

3.7.9 Spezieller Charakterisierungssatz für Uhrenautomaten mit Haltemöglichkeit *Eine Sprache von Intervallwörtern wird genau dann von einem Uhrenautomaten mit Haltemöglichkeit erkannt, wenn sie durch einen existentiell-monadischen* Πdur*-Satz definiert wird.* ■

Da $\mathrm{d}(X,x) \sim c$ äquivalent zu $\mathrm{dur}(X,\{x \mid \mathtt{tt}\},x) \sim c$ und $\mathrm{dur}(X,Y,x) \sim c$ äquivalent zu $\int_X^x 1\{x \mid \mathtt{tt}\} \sim c$ ist, erlauben wir in $\Pi\mathrm{dur}$-Formeln auch relative Abstandsformeln und in $\Pi\int$-Formeln sowohl relative Abstandsformeln wie auch relative Dauerformeln.

Zum Abschluß des Kapitels werden wir ein schwierigeres Beispiel betrachten und zu den „Teilern von Eins“ zurückkehren.

3.7.10 Beispiel (3.3.3 fortgef.) Hier wollen wir eine zu dem Automaten aus Abbildung 3.5 äquivalente existentiell-monadische $\Pi\mathrm{dur}$-Formel angeben. Sie hat die Gestalt:

$$\exists X_0 \ldots \exists X_7(\mathsf{Zstd}(X_0,\ldots,X_7) \wedge \mathsf{Test}(X_0,\ldots,X_7))\,.$$

Die Formel Zstd kodiert die Zustände, die in einem akzeptierenden Lauf angenommen werden. Die Formel Test überprüft, ob die Registertests erfüllt sind.

Die Variable X_i soll mit den Stellen belegt werden, an denen ein Ereignis steht, das der Automat liest, wenn er in Zustand i ist. Dies wird durch die folgende Formel erreicht:

$$\begin{aligned}\mathsf{Zstd} \;\triangleq\; & X_0 0 \wedge X_1 1 \wedge X_7 \mathsf{max} \wedge \forall x(\bigwedge_{i<j<8} \neg X_i x \wedge X_j x) \\ & \wedge \forall x(x+1 < \mathsf{max} \rightarrow \bigwedge_{i:0<i<7} (X_i x \leftrightarrow X_{[i+1]}(x+1)))\,.\end{aligned}$$

Hierbei steht $[i+1]$ für $i+1$, außer im Falle $i=6$, dann sei $[i+1]$ gleich Eins.

Für die Konstruktion von Test führen wir eine Fallunterscheidung nach dem erreichten Zustand durch, wobei es zum Teil eine Rolle spielt, ob ein Zustand zum ersten Mal aufgesucht wird:

$$\begin{aligned}&\mathsf{Test} \;\triangleq \\ &\forall x\ ((X_3 x \wedge x = \mathbf{3} \rightarrow \mathrm{d}(X_0,x) = 1 \wedge \mathrm{dur}(X_3, X_0 \cup X_2 \cup X_3, x) = 1) \\ &\quad \wedge\ (X_3 x \wedge x > \mathbf{3} \rightarrow \mathrm{d}(X_6,x) = 1 \wedge \mathrm{dur}(X_3, X_4 \cup X_2 \cup X_3, x) = 1 \\ &\qquad\qquad \wedge\ \mathrm{dur}(X_3, X_4 \cup X_5 \cup X_1 \cup X_3, x) = 1) \\ &\quad \wedge\ (X_6 x \rightarrow \mathrm{d}(X_3,x) = 1 \wedge \mathrm{dur}(X_0 \cup X_6, X_1 \cup X_5 \cup X_6, x) = 1 \\ &\qquad\qquad \wedge\ \mathrm{dur}(X_0 \cup X_6, X_2 \cup X_5 \cup X_6, x) = 1) \\ &\quad \wedge\ (X_1 x \wedge X_7(x+1) \rightarrow \mathrm{d}(X_0 \cup X_{,}, x)1 = 1) \\ &\quad \wedge\ (X_7 x \wedge x = \mathbf{2} \rightarrow \mathrm{dur}(X_0, X_0 \cup X_7, x) = 1 \wedge \mathrm{dur}(X_6, X_1 \cup X_7, x) = 1) \\ &\quad \wedge\ (X_7 x \wedge x > \mathbf{2} \rightarrow \mathrm{dur}(X_4, X_4 \cup X_7, x) = 1 \wedge \mathrm{dur}(X_6, X_1 \cup X_7, x) = 1))\,.\end{aligned}$$

Kapitel 4

Uhrenautomaten

In diesem Kapitel wird das Modell des Uhrenautomaten eingehend studiert. Klassische automatentheoretische Fragen stehen in der ersten Hälfte im Vordergrund, während im zweiten Teil Aspekte der in der Einleitung vorgestellten, von Alur und Henzinger formulierten Frage nach einer „maximalen“, gegen die booleschen Operationen abgeschlossenen und effektiv (und effizient) handhabbaren Klasse von Sprachen von Intervallwörtern (siehe [AH91a]) studiert werden.

Zuerst wiederholen wir die grundlegenden, uhrenerkennbare Sprachen betreffenden Sachverhalte, die schon in der einführenden Arbeit [AD90] von Alur und Dill zu finden sind:

1) Die diskrete Projektion jeder uhrenerkennbaren Sprache ist regulär.[1]

2) Das Leerheitsproblem für Uhrenautomaten ist entscheidbar (aber PSPACE-vollständig).

3) Das Inklusionsproblem, das Äquivalenzproblem und das Universalitätsproblem sind nicht entscheidbar.

4) Die Klasse der deterministisch uhrenerkennbaren Sprachen ist echt in der Klasse der uhrenerkennbaren Sprachen enthalten.

Aus der Entscheidbarkeit des Leerheitsproblems und der Unentscheidbarkeit des Universalitätsproblems kann man sofort schließen, daß es kein effektives Verfahren gibt, daß zu einem Uhrenautomaten $\mathfrak{A}$ einen neuen Uhrenautomaten $\mathfrak{A}'$ konstruiert, der das Komplement von $L(\mathfrak{A})$ erkennt. Damit ist allerdings noch nicht gesagt, daß es auch eine uhrenerkennbare Sprache gibt, deren Komplement nicht uhrenerkennbar ist. Eine entsprechende Behauptung wird von Alur und Dill in [AD90] und [Alu91] durch Angabe einer Beispielsprache aufgestellt und begründet, jedoch geben sie keinen

[1] Hier sei schon darauf hingewiesen, daß wir innerhalb des logischen Rahmens im nächsten Kapitel einen Alternativbeweis für dieses und das unter Punkt 2) genannte Ergebnis geben werden.

formalen Beweis. In Abschnitt 4.2 entwickeln wir deshalb ein Iterationslemma für Uhrenautomaten, das auch von eigenem Interesse ist, letztlich aber benutzt wird, um die Behauptung von Alur und Dill zu beweisen.

In Abschnitt 4.3 wollen wir den Registerindex, der im Falle der Uhrenautomaten als Uhrenindex bezeichnet wird, als das Maß der Komplexität einer uhrenerkennbaren Sprache betrachten. Wir beweisen, daß es Sprachen beliebiger Komplexität gibt, d. h., daß es zu jeder natürlichen Zahl n eine uhrenerkennbare Sprache gibt, so daß jeder Uhrenautomat, der die Sprache erkennt, mindestens n Uhren benötigt. Das Komplexitätsmaß des Uhrenindexes teilt **NTL** also in eine unendliche Hierarchie ein. Außerdem werden wir zeigen, daß der Uhrenindex insofern zur effektiven Minimierung von Uhrenautomaten ein unbrauchbares Maß ist, als er nicht berechenbar ist.

Die oben aufgezählten „klassischen“ Ergebnisse auf dem Gebiet der Uhrenautomaten zeigen deutlich, daß die Klasse **NTL** in ihren sprachen- bzw. automatentheoretischen Eigenschaften komplizierter als die Klasse der von endlichen Automaten erkannten diskreten Sprachen ist. Denn die regulären Sprachen sind effektiv unter der Anwendung boolescher Operationen abgeschlossen, das deterministische und das nichtdeterministische Automatenmodell sind gleichwertig und der Leerheitstest ist in quadratischer Zeit durchführbar. Darüberhinaus gibt es im Fall der endlichen Wörter zu jeder regulären Sprache einen eindeutigen (zustands-)minimalen deterministischen Automaten, der in Zeit $\mathcal{O}(n \log n)$ aus einem beliebigen deterministischen Automaten bestimmt werden kann.

Aus dieser Beobachtung resultierte die in der Einleitung erwähnte Frage von Alur und Henzinger nach einer „maximalen“ Klasse von Sprachen von Intervallwörtern, die gegenüber booleschen Kombinationen abgeschlossen ist und ein elementar entscheidbares Leerheitsproblem hat. Eine einfache aber unbefriedigende Antwort ist die folgende: Man nehme die Klasse aller uhrenerkennbaren Sprachen, deren Komplement auch uhrenerkennbar ist. Wir wollen diese Klasse den üblichen Konventionen folgend mit $\mathbf{NTL} \cap \text{co-}\mathbf{NTL}$ bezeichen. Die Antwort ist insofern unbefriedigend, als bislang kein effektives Verfahren bekannt ist, das entscheiden würde, ob eine durch einen Uhrenautomaten gegebene Sprache zu $\mathbf{NTL} \cap \text{co-}\mathbf{NTL}$ gehört. Genauso wenig sind Formalismen — wie z. B. ein geeignetes Automatenmodell — bekannt, die diese Klasse beschreiben würden.

Auf der Suche nach einer geeigneten Teilklasse von **NTL** wurde von Alur und Henzinger in [AH92] ein Modell eines Zweiwege-Uhrenautomaten analysiert. Durch eine hinzugenommene semantische Einschränkung konnte durch diesen Automatentyp tatsächlich eine relevante, unter den booleschen Operationen abgeschlossene Teilklasse ($\mathbf{2DTL}_\omega$) ausgesondert werden. In Abschnitt 4.4 werden wir diese Klasse vorstellen und beweisen, daß die semantische Bedingung aus [AH92] unentscheidbar ist, was zeigt, daß die Definition der Klasse nicht effektiv ist.

In Abschnitt 4.5 betrachten wir mit Hinsicht auf die Fragestellung von Alur und Henzinger die Klasse der Sprachen, die die von *eindeutigen* Uhrenautomaten erkann-

ten Sprachen enthält und die wir mit **UTL** bezeichnen wollen. Sie wird in dieser Arbeit zum ersten Mal behandelt. Es ist offen, ob **UTL** gegen Komplementbildung abgeschlossen ist. Wir können jedoch zeigen, daß es entscheidbar ist, ob ein gegebener Uhrenautomat eindeutig ist, daß die Inklusion $\mathbf{2DTL}_\omega \subseteq \mathbf{UTL}$ gilt und daß die uhrenerkennbare Sprache, die wir benutzen, um den Nichtabschluß von **NTL** unter Komplementbildung zu belegen, nicht zu **UTL** gehört. Aus letzterem folgt dann auch $\mathbf{UTL} \subsetneqq \mathbf{NTL}$.

Den Abschluß des Kapitels (Abschnitt 4.7) bilden Betrachtungen, die ebenfalls von dem von Alur und Henzinger ausgesprochenen Wunsch geleitet sind, einen ansprechenden Begriff von zeitabhängigem Verhalten zu finden. Doch anstatt das Modell des Uhrenautomaten einzuschränken, schlagen wir vor, Erkennbarkeit relativ zu Teilsprachen der Sprache aller Intervallwörter über einem gegebenen Alphabet zu studieren. Die einfachste Einschränkung führt uns zum Begriff der beschränkten Variation und zum Erfolg in dem Sinne, daß wir eine Klasse von Sprachen erhalten, die gegen boolesche Kombinationen abgeschlossen ist und effektiv behandelt werden kann. Zum Schluß des Abschnitts 4.7 werden wir dann noch zeigen, daß die besagte Einschränkung bei stärkeren Automatenmodellen nicht zum Erfolg führt.

Der Vollständigkeit halber sollte hier noch erwähnt werden, daß über die oben genannten Ergebnisse hinausreichende Untersuchungen auf dem Gebiet der Uhrenautomaten zwei Themen betreffen: probabilistische Varianten des Automatenmodells (siehe [ACD91] und [ADC91]) und Minimierungsfragen. Letztere werden in [YL93] und in [ACD+92] behandelt und beschränken sich ausschließlich auf das Problem der Zustandsminimierung.

Wir beginnen mit technischen Vorbereitungen.

4.1 Äquivalente Folgen von Uhrenständen

Zuerst führen wir eine neue Schreibweise für die Anwendung von Fortschreibungen in Uhrenautomaten ein.

Sei R eine Uhrenmenge und $\eta\colon R \to \mathbf{P}_0$ die Funktion, die jede Uhr auf Null abbilde.

Ist $\rho\colon R \to \mathbf{P}_0$ ein Uhrenstand und $w \in \mathbf{P}$, so schreiben wir $\rho + w$ für den Uhrenstand ρ' mit $\rho'(r) = \rho(r) + w$ für jedes $r \in R$. Damit kann die Anwendung der in Σ_{UA} einzig möglichen Fortschreibung, der Funktion θ, die jede Uhr auf add abbildet, vereinfacht geschrieben werden: Es gilt $\rho \overset{w}{\theta} = \rho + w$.

Wir wiederholen noch einmal die Definition der in Abschnitt 3.6 auf der Menge aller Uhrenstände über einem festen Registersatz eingeführten Äquivalenzrelation $\equiv$ und vergröbern sie geeignet. Für nichtnegative reelle Zahlen r und s schreiben wir $r \equiv s$, falls für jede natürliche Zahl l die beiden folgenden Bedingungen erfüllt sind:

$$r \le l \text{ gdw. } s \le l \quad \text{und} \quad r \ge l \text{ gdw. } s \ge l\,. \tag{4.1}$$

Sind ρ und ρ' Uhrenstände, dann wird $\rho \equiv \rho'$ und $\rho \equiv_k \rho'$ geschrieben, wenn $\rho(r) \equiv \rho'(r)$ für jedes $r \in r$ gilt.

Gilt die angegebene Bedingung (4.1) für alle l, die kleiner oder gleich einer vorgegebenen natürlichen Zahl k sind, so schreiben wir $r \equiv_k s$ bzw. $\rho(r) \equiv_k \rho'(r)$. Wir sprechen auch von der *k-Klasse* einer Zahl oder einer Registerbelegung.

4.1.1 Bemerkung 1) Ist ϕ ein Uhrentest mit größtem Vergleichswert l, gilt $l \leq k$ und gilt $\rho \equiv_k \rho'$ für zwei Registerbelegungen ρ und ρ', dann erfüllt ρ den Test ϕ genau dann, wenn ρ' den Test ϕ erfüllt.

2) Ist R eine Uhrenmenge der Mächtigkeit s, dann gibt es $2^s(k+1)^s$ verschiedene $\equiv_k$-Klassen von Uhrenständen.

Wir wollen uns überlegen, unter welchen Voraussetzungen eine Transitionsfolge, die zusammen mit einer Registerbelegung eine Berechnung auf einem Wort bestimmt, auch eine Registerbelegung auf einem anderen Wort u' mit der gleichen Beschriftung (und gleichen Länge) definiert.

Sei dazu $u = (\underline{a}, \underline{w})$ eine Intervallwort, $\underline{\delta}$ eine konsistente Transitionsfolge mit $\delta_i = (q_i, a_i, \phi, \zeta_i, q_{i+1})$ und ρ ein Uhrenstand. Die Transitionsfolge und der Uhrenstand *bestimmen zwei Folgen $\underline{\rho}$ und $\underline{\rho}'$ von Uhrenständen* durch die Regeln $\rho_0 = \rho$ und $\rho_i' = \rho_i + w_i$ sowie $\rho_{i+1} = \rho_i'\zeta_i$ für $i < |u|$. (Damit sind die Längen von $\underline{\rho}$ und $\underline{\rho}'$ durch $|\underline{\rho}| = 1 + |u|$ und $|\underline{\rho}'| = |u|$ gegeben.)

Die Bedingungen, die wir in Abschnitt 1.2 an eine Berechnung gestellt haben, erlauben uns nun, genau dann

$$(q_0, \rho_0) \xrightarrow[(a_0, w_0)]{\delta_0} (q_1, \rho_1) \xrightarrow[(a_1, w_1)]{\delta_1} (q_2, \rho_2) \xrightarrow[(a_2, w_2)]{\delta_2} \dots$$

zu schreiben, wenn ρ_i' den Test ϕ_i für jedes $i < |u|$ erfüllt.

Sei nun $u' = (\underline{a}, \underline{w}')$ ein weiteres Wort mit gleicher Beschriftung (und damit auch mit gleicher Länge) wie u. Dann bestimmt $\underline{\delta}$ zusammen mit ρ in gleicher Weise zwei Folgen $\underline{\sigma}$ und $\underline{\sigma}'$ von Uhrenständen (mit $\sigma_0 = \rho$).

Ist k eine obere Schranke für die Vergleichswerte, die in Uhrentests von $\underline{\delta}$ auftreten, dann bestimmt nach 4.1.1 die Transitionsfolge $\underline{\delta}$ zusammen mit ρ eine Berechnung auf u', wenn $\rho_i' \equiv_k \sigma_i'$ für jedes $i < |u|$ gilt.

Dies führt zu der folgenden Definition:

4.1.2 Definition (äquivalente Folgen von Uhrenständen) Seien $\underline{\rho}$ und $\underline{\sigma}$ Folgen von Uhrenständen über der gleichen Uhrenmenge und sei k eine natürliche Zahl. Die Folgen heißen *äquivalent*, wenn sie gleiche Länge haben und $\rho_i \equiv \sigma_i$ für jedes $i < |\underline{\rho}|$ gilt. Sie heißen *k-äquivalent*, wenn jeweils $\rho_i \equiv_k \sigma_i$ gilt. Es wird $\underline{\rho} \equiv \underline{\sigma}$ bzw. $\underline{\rho} \equiv_k \underline{\sigma}$ geschrieben.

Der *maximale Vergleichswert* eines Uhrenautomaten sei das Maximum aus Null und den Vergleichswerten, die in den Uhrentests der Transitionen des Automaten auftreten.

Aus der obigen Betrachtung erhalten wir nun das folgende Kriterium:

4.1.3 Bemerkung Sei $\mathfrak{A}$ ein Uhrenautomat mit maximalem Vergleichswert l und sei $k \geq l$. Sei $\underline{\delta}$ eine konsistente Transitionsfolge und seien u und u' Intervallwörter mit gleicher Beschriftung wie $\underline{\delta}$.

Sind $\underline{\rho}$ und $\underline{\rho'}$ sowie $\underline{\sigma}$ und $\underline{\sigma'}$ die durch einen beliebigen Uhrenstand und die Transitionsfolge $\underline{\delta}$ auf u bzw. u' bestimmten Folgen von Uhrenständen und gilt $\rho' \equiv_k \sigma'$, so ist durch $\underline{\delta}$ genau dann eine Berechnung auf u gegeben, wenn durch $\underline{\delta}$ eine Berechnung auf u' gegeben ist. In diesem Fall sind entweder beide Berechnungen akzeptierend oder keine von beiden ist es.

4.2 Beschleunigungen von Intervallwörtern

Iterationslemmata sind aus der Theorie der formalen Sprachen diskreter Wörter u.a. für reguläre, lineare und kontextfreie Sprachen bekannt (siehe z. B. [HU79]). Sie werden in der Regel benutzt, um von konkreten Sprachen zu zeigen, daß sie nicht zu der betreffenden Sprachklasse gehören. Außerdem verleihen sie eine gewisse Einsicht in die Schwächen des betrachteten Automatenmodells (bzw. des Grammatiktyps), denn typischerweise erlauben sie, von einem gegebenen Automaten und einem akzeptierten Wort auf die Existenz weiterer akzeptierter Wörter zu schließen, die durch „Iterieren“ aus dem gegebenen entstehen und die der Automat von dem gegebenen Wort nicht unterscheiden kann.

Entscheidend für das jeweilige Iterationslemma ist nun, unter welchen Voraussetzungen ein Wort auf welche Weise iteriert werden kann. Im Falle der endlichen diskreten Automaten z. B. kann jedes Wort iteriert werden, das ein gewisse Mindestlänge überschreitet. Ist dies der Fall, dann besteht die mögliche Iteration in der Wiederholung eines Teilwortes, für das gewisse Längengarantien gegeben werden.

Im Falle der Uhrenautomaten ist die Situation vergleichbar. Wir werden nämlich zeigen, daß ein Intervallwort iteriert werden kann (wir werden den Vorgang ‚beschleunigen‘ nennen), wenn in ein kurzes Intervall viele Momente des Wortes fallen. Dann besteht die Iteration in einer Wiederholung eines Teilwortes, für das eine Mindestlänge garantiert wird. Allerdings müssen bei der Wiederholung gewisse Dauerbeschränkungen eingehalten werden: Im wesentlichen muß die Gesamtdauer der Wiederholungen mit der Dauer des ausgewählten Teilwortes übereinstimmen.

4.2.1 Das Beschleunigungslemma

Falls nicht anders vermerkt, seien Segmente eines Wortes in diesem Abschnitt immer endlich. Dem Segment eines Intervallwortes wird als Dauer die Dauer des assoziierten Teilwortes zugeordnet.

Eine *Stauchung* eines Intervallwortes $u = (\underline{a}, \underline{w})$ ist ein Intervallwort $v = (\underline{a}, \underline{w'})$ mit der Eigenschaft $w'_i \leq w_i$ für alle $i < |u|$. Eine *Deformation* von u ist ein Intervallwort mit gleicher Beschriftung.

4.2.1 Definition (Beschleunigung) Sei e eine positive natürliche Zahl. Eine *Beschleunigung* mit *Exponenten* e eines Intervallwortes u an aufeinanderfolgenden, nichtleeren Segmenten (i, j) und (j, k) ist ein Wort

$$u(0, i)vv_0 \ldots v_{e-2}u(j, *)\,,$$

wobei v eine Stauchung von $u(i, j)$ und jeweils v_i eine Deformation von $u(j, k)u(i, j)$ ist und zusätzlich

$$||vv_0 \ldots v_{e-2}|| \quad = \quad ||u(i, j)|| \tag{4.2}$$

gilt.

Anschaulich gesprochen entsteht die Beschleunigung durch das Ersetzen beider Segmente durch sich abwechselnde deformierte Kopien, wobei die erste Wiederholung des ersten Segmentes eine Stauchung und die letzte Wiederholung des zweiten Segmentes exakt sein muß.

Ist L eine Sprache von Intervallwörtern und u ein Intervallwort in L, so sagen wir, daß u an den aufeinanderfolgenden Segmenten β und γ bzgl. L *beschleunigt* werden kann, falls alle Beschleunigungen von u an β und γ zu L gehören.

4.2.2 Beschleunigungslemma *Zu jedem Uhrenautomaten $\mathfrak{A}$ und jeder positiven natürlichen Zahl t gibt es eine Zahl t' mit der Eigenschaft, daß für jedes Intervallwort $u \in \mathrm{L}(\mathfrak{A})$ gilt: Jedes Segment mit Länge $> t'$ und Dauer kleiner als Eins enthält zwei aufeinanderfolgende Segmente, von denen das erste eine Mindestlänge von t hat, an denen u bzgl.* $\mathrm{L}(\mathfrak{A})$ *beschleunigt werden kann.*

Beweis. Wir stellen zuerst eine kombinatorische Überlegung an. Dazu betrachten wir diskrete endliche Wörter über einem Alphabet B und markieren sie mit Elementen aus einer endlichen, nichtleeren Menge M. Formal verstehen wir unter einem solchen markierten Wort ein Paar (u, μ), bestehend aus einem endlichen diskreten Wort u über B zusammen mit einer Markierungsfunktion $\mu\colon |u| \to \wp(M)$.

Zwei aufeinanderfolgende Segmente (i, j) und (j, k) eines markierten endlichen Wortes (u, μ) mit $u = \underline{a}$ heißen *gut*, falls a_i und a_{k-1} gleich sind und falls jede Marke, die im Segment (i, j) auftritt, auch im Segment (j, k) vorkommt.

Behauptung 1 Sei t eine positive natürliche Zahl. Dann gibt es in jedem markierten Wort der Länge $|B||M|t+1$ mindestens zwei aufeinanderfolgende gute Segmente, von denen das erste mindestens die Länge t hat.

Beweis. Sei (u,μ) ein markiertes endliches Wort mit $u = \underline{a}$. Wir legen für jeden Buchstaben $a \in B$ eine Zahl p_a fest: $p_a = \max(\{-1\} \cup \{i \mid u_i = a\})$. Außerdem definieren wir die Funktion $f\colon |u| \to B \times \wp(M)$ mit den Komponentenfunktionen f_0 und f_1 wie folgt:

$$f_0(i) = a_i\,,$$

$$f_1(i) = \begin{cases} \mu(i) \cup \ldots \cup \mu(p_{a_i})\,, & \text{falls } i \le p_{a_i}, \\ \emptyset\,, & \text{sonst.} \end{cases}$$

Offensichtlich gibt es in u zwei aufeinander folgende gute Segmente, von denen das erste mindestens die Länge t hat, wenn es ein i mit folgender Eigenschaft gibt:

(+) Es gibt ein j mit $j - i \ge t$ und $f(i) = f(j)$.

Dann sind nämlich die Segmente (i,j) und $(j, p_{a_i}+1)$ gute Segmente.

Wir stellen folgendes fest: Aus $i < j$ und $f_0(i) = f_0(j)$ folgt $f_1(j) \subseteq f_1(i)$. Deshalb gibt es in jedem Wort mit einer Mindestlänge von $|B||M|t+1$ eine Position i mit Eigenschaft (+). □

Im folgenden setzen wir voraus, daß $\mathfrak{A}$ ein Uhrenautomat mit einer nichtleeren Registermenge R, Zustandsmenge Q und maximalem Vergleichswert m sei. Weiterhin sei u ein Intervallwort aus $\mathrm{L}(\mathfrak{A})$ und (c,c') ein Segment von u mit $c' - c \ge t_0$ und mit Dauer kleiner als $\frac{1}{2}$. Sei außerdem eine akzeptierende Berechnung von $\mathfrak{A}$ auf u durch die konsistente Transitionsfolge $\underline{\delta}$ gegeben. Es seien $\underline{\rho}$ und $\underline{\rho'}$ die zugehörigen Folgen von Uhrenständen, und es sei $\underline{q}$ die zugehörige Folge von Zuständen.

Zunächst halten wir folgendes fest: Für jede Uhr r ändern sich die m-Klassen (Äquivalenzklassen bezüglich $\equiv_m$) der Werte $\rho'_{c-1}(r)$, …, $\rho'_{c'-1}(r)$ an höchstens drei Stellen. Denn nachdem eine Uhr zurückgesetzt wurde, kann sie in einem Segment mit Dauer kleiner als Eins den Wert Eins nicht mehr erreichen. Und bevor sie in einem Segment mit Dauer kleiner als Eins zurückgesetzt wird, kann sie höchstens eine ganzzahlige Grenze berühren.

Ist also $t_0 \ge t_1 4^{|R|}$ (sehr grobe Abschätzung), so gibt es ein Segment (d,d'), das in (c,c') enthalten ist, dessen Länge mindestens t_1 ist und in dem für jede Uhr r gilt:

(*) Die m-Klassen der Werte von r in (d,d') sind alle gleich. D. h., alle Uhrenstände $\rho'_i(r)$ mit $d - 1 \le i \le d' - 1$ sind m-äquivalent.

Wir wollen Behauptung 1 auf das aktuelle Segment (d, d') anwenden. Dazu sei B das kartesische Produkt aus der Zustandsmenge Q und den m-Klassen der Uhrenstände und $M = R$. Wir betrachten das markierte Wort $(\underline{v}, \mu)$ mit

$$\begin{aligned} v_i &= (q_{d+i}, \rho'_{d+i-1} \equiv_k) && \text{für } i < d' - d, \\ \mu(0) &= \emptyset\,, \\ \mu(i) &= \{r \mid \rho_d(i) = 0\} && \text{für } 0 < i < d - d'. \end{aligned}$$

Aus Behauptung 1 können wir nun schließen, daß es gute Segmente (g, g') und (g', g'') gibt, wobei $g' - g \geq t + 1$ ist, wenn wir

$$t_1 \geq |Q|(2m + 2)|R|(t + 1) + 1 \tag{4.3}$$

wählen. Wir setzen $i = d + g$, $j = d + g' - 1$ und $k = d + g'' - 1$. Dann heißt ‚gut' zu sein:

(**) Es ist $q_i = q_k$, und, ist r eine Uhr und n eine Position mit $i < n \leq j$ und $\rho_n(r) = 0$, so gibt es auch eine Position n' mit $j < n' \leq k$ und $\rho_{n'}(r) = 0$. D. h., wird eine Uhr an einer Position des Segmentes $(i + 1, j)$ zurückgesetzt, so auch an einer Position des Segmentes $(j + 1, k)$.

Behauptung 2 Die konsistente Folge $\underline{\delta}' = \underline{\delta}(0, i)\underline{\delta}(i, k)^e\underline{\delta}(k, *)$ bestimmt eine akzeptierende Berechnung jeder Beschleunigung u' mit Exponenten e an (i, j) und (j, k).

Beweis. Sei u' eine Beschleunigung mit Exponenten e an (i, j) und (j, k). Außerdem seien $\underline{\eta}$ und $\underline{\eta}'$ die durch $\underline{\delta}'$ bestimmten Folgen von Uhrenständen.

Zuerst halten wir fest, daß $\underline{\delta}'$ und u' nach Konstruktion die gleiche Beschriftung haben. Da $\underline{\delta}$ eine akzeptierende Berechnung auf u bestimmt und der erste und letzte Zustand von $\underline{\delta}(i, k)$ übereinstimmen, ist $\underline{\delta}$ eine akzeptierende konsistente Transitionsfolge und wir brauchen nur noch zu überprüfen, ob die Uhrentests erfüllt sind.

Da sowohl u und u' wie auch $\underline{\delta}$ und $\underline{\delta}'$ auf dem Segment $(0, i)$ übereinstimmen, sind auch die Uhrenstände die gleichen und folglich werden auf diesem Stück auch alle Uhrentests erfüllt.

Wegen 4.1.1 reicht es, für jede Uhr r die folgenden Beziehungen nachzuweisen:

$$\rho'_{i+n-1}(r) \equiv_m \eta'_{i+n'(k-i)+n-1}(r) \qquad \text{für } n' < e \text{ und } n \leq k - i \tag{4.4}$$

sowie

$$\rho'_{k+n-1}(r) = \eta'_{i+e(k-i)+n-1}(r) \qquad \text{für } k + n < |u|\,. \tag{4.5}$$

Dazu führen wir eine Fallunterscheidung danach durch, ob für eine gegebene Uhr r in dem Segment $(i + 1, j)$ von u zurückgesetzt wird.

1. Fall, r wird nicht in dem Segment $(i+1,j)$ von u zurückgesetzt. Dann wird r wegen $(**)$ auch nicht im Segment $(j+1,k)$ zurückgesetzt. Aus (4.2) folgt dann sofort (4.5). Außerdem erhalten wir aus (4.5) die Ungleichungskette

$$\rho'_{i-1}(r) \;=\; \eta'_{i-1}(r) \;\leq\; \eta'_n(r) \;\leq\; \eta'_{i+e(k-i)-1}(r) \;=\; \rho'_{k-1}(r)\,.$$

für jedes n mit $i-1 \leq n \leq i+e(k-i)-1$. Da wegen $(*)$ die Beziehung $\rho'_n(r) \equiv_m \rho'_{n'}(r)$ für alle n und n' mit $i-1 \leq n \leq n' \leq k-1$ gilt, ist auch $\eta'_n(r) \equiv_m \rho'_{n'}(r)$ für alle n und n' mit $i-1 \leq n \leq n' \leq i+e(k-i)-1$ erfüllt, woraus (4.4) folgt.

2. Fall, r wird in dem Segment $(i+1,j)$ von u zurückgesetzt. Dann wird r wegen $(**)$ auch im Segment $(j+1,k)$ zurückgesetzt. Daraus können wir dann sofort auf $\eta'_{i+e(k-i)-1}(r) = \rho'_{k-1}(r)$ schließen, woraus (4.5) folgt. Wegen $(*)$ und weil r zurückgesetzt wird, gilt für n mit $i-1 \leq n \leq k-1$ die Ungleichung $0 < \rho'_n(r) < 1$ und deshalb aufgrund der Bedingung, die an eine Stauchung gestellt wird, auch $0 < \eta'_n(r) \leq \rho_n(r) < 1$ für $i-1 \leq n \leq k-1$. Da r in (i,k) zurückgesetzt wird, gilt wegen $W_k - W_i < \frac{1}{2}$ auch $\eta'_{k-1}(r) < \frac{1}{2}$. Davon ausgehend kann nun für $n' = 1,\ldots,(e-1)$ induktiv $0 < \eta'_{i+n'(k-i)+n-1}(r) < \frac{1}{2}$ für n und n' mit $n' < e$ und $n \leq k-i$ gezeigt werden, woraus (4.4) folgt. □

Wählen wir also t' größer als

$$2 \times 4^{|R|}(|Q|(2m+2)|R|(t+1)+1)\,,$$

so stimmt die Behauptung. ■

4.2.2 Nichtabschluß unter Komplementbildung

Wir beginnen mit dem Beispiel, das in [AD90] und [Alu91] zu finden ist:

4.2.3 Beispiel Es sei $L_\sim$ die Sprache aller unbeschränkten Intervallwörter $\underline{w}$, für die es zwei Positionen i und j mit $0 < i < j$ und $W_j - W_i = 1$ gibt. In Abbildung 3.4 ist ein Automat dargestellt, der die Uhrenerkennbarkeit von $L_\sim$ belegt.

Das Komplement dieser Sprache enthält ein unäres unbeschränktes Intervallwort $\underline{w}$ genau dann, wenn für jede Position i mit $i > 0$ gilt: $W_i + 1 \notin \{W_{i+1}, W_{i+2}, \ldots\}$.

Wir zeigen nun:

4.2.4 Lemma *Das Komplement der Sprache $L_\sim$ aus 4.2.3 ist nicht uhrenerkennbar.*

Beweis. Nehmen wir an, das Komplement von $L_\sim$ würde von einem Uhrenautomaten $\mathfrak{A}$ erkannt und t' wäre die aus dem Beschleunigungslemma kommende Zahl (für $t=1$). Wir setzen $q = t'+3$ sowie $\delta = 1\frac{1}{q}$ und betrachten das Wort $\underline{w}$ mit $w_i = \frac{\delta}{q}$. Dieses Wort gehört zu dem Komplement von $L_\sim$, denn: Gäbe es i und j mit $\frac{j\delta}{q} - \frac{i\delta}{q} = 1$, so

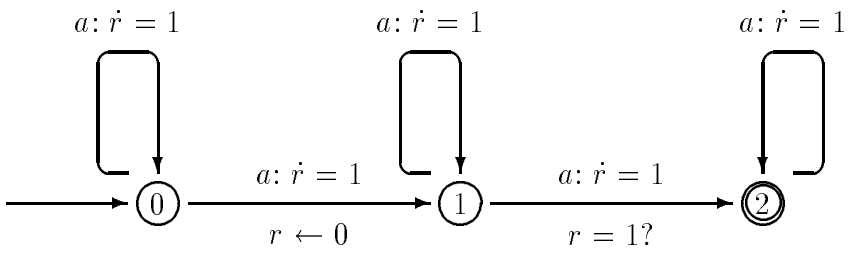


Abbildung 4.1: Ein Uhrenautomat aus [AD90], der den Nichtabschluß von **NTL** unter Komplementbildung belegt

Der Uhrenautomat aus [AD90] und [Alu91], der die Sprache $L_\sim$ erkennt, deren Komplement nicht uhrenerkennbar ist.

würde auch $(j-i)(q+1) = q^2$ gelten. Diese Gleichung ist aber nicht durch natürliche Zahlen zu erfüllen.

Das Segment $(1+q, 2q)$ hat mindestens die Länge t' und eine Dauer, die kleiner als Eins ist. Damit ist das Beschleunigungslemma anwendbar, etwa an den Segmenten (i,j) und (j,k) mit $i \geq 1+q$. Es gehört aber jede Beschleunigung mit Exponenten größer als Eins, in der die Stauchung v (vgl. (4.2)) mit einem Ereignis der Dauer $\frac{1}{q}$ beginnt, nicht zu dem Komplement von $L_\sim$, weil $W_i + \frac{1}{q} - W_{i-q} = 1$ gilt. ■

Die Beispielsprache $L_\sim$ und der vorstehende Beweis lassen sich unmittelbar auf endliche Intervallwörter übertragen. Deshalb können wir festhalten:

4.2.5 Korollar *Die Klasse der uhrenerkennbaren Sprachen endlicher, unendlicher bzw. unbeschränkter Intervallwörter ist nicht gegen Komplementbildung abgeschlossen.*

4.3 Uhrenindex

Die Komplexität einer uhrenerkennbaren Sprache kann man danach messen, wieviel Uhren in einem sie erkennenden Automaten nötig sind. Dies ist auch in Hinblick auf die Durchführung des Leerheitstests eine vernünftige Größe, da die Anzahl der Uhren in die Laufzeit-Komplexität des Tests exponentiell eingeht (siehe [AD90]).

Es wird im folgenden zuerst gezeigt, daß das Problem, die minimale Anzahl der zum Erkennen einer Sprache benötigten Uhren zu bestimmen, unentscheidbar ist. Insbesondere kann es also für Uhrenautomaten kein effektives Minimierungsverfahren geben, das sich an der Anzahl der benutzten Uhren orientiert. Danach wird bewiesen, daß die „Uhrenhierarchie“ unendlich ist, d. h., es wird zu jeder natürlichen Zahl eine uhrenerkennbare Sprache vorgestellt, die nur von Uhrenautomaten erkannt werden kann, die mindestens soviele Uhren besitzen, wie die Zahl vorgibt.

Vorab geben wir die für diesen Abschnitt fundamentale Definition:

4.3.1 Definition (Uhrenindex) Ist $\mathfrak{A} = (Q, R, s, \Delta, F)$ ein Uhrenautomat, so ist $|R|$ sein *Uhrenindex*, in Zeichen $\mathbf{U}(\mathfrak{A})$. Ist L eine uhrenerkennbare Sprache, so ist

$$\mathbf{U}(L) \;=\; \max\{\mathbf{U}(\mathfrak{A}) \mid \mathfrak{A} \text{ Uhrenautomat mit } \mathrm{L}(\mathfrak{A}) = L\}$$

der *Uhrenindex* von L.

Damit ist der Uhrenindex eines Uhrenautomaten gleich dem Registerindex des korrespondierenden Σ_{UA}-Automaten und der Uhrenindex einer uhrenerkennbaren Sprache der Registerindex dieser Sprache, betrachtet als Σ_{UA}-erkennbare Sprache.

4.3.1 Nichtberechenbarkeit des Uhrenindexes

4.3.2 Satz *Der Uhrenindex von uhrenerkennbaren Sprachen ist nicht berechenbar. (Dies gilt sowohl für Sprachen endlicher als auch für Sprachen unendlicher oder unbeschränkter Intervallwörter. Vorausgesetzt wird in allen Fällen ein mindestens dreielementiges Alphabet.)*

Mit anderen Worten: Die Funktion $\mathfrak{A} \mapsto \mathbf{U}(\mathrm{L}(\mathfrak{A}))$, die jedem Uhrenautomaten den Uhrenindex der von ihm erkannten Sprache zuordnet, ist nicht berechenbar.

Beweis. Wir reduzieren das Universalitätsproblem für Uhrenautomaten, was in jedem der drei Fälle unentscheidbar ist, sofern die genannte Voraussetzung über die Alphabetgröße zutrifft (siehe [AD]), auf das gegebene Problem. Dabei nutzen wir die Tatsache aus, daß man zu jedem Uhrenautomaten $\mathfrak{A}$ effektiv einen diskreten Automaten konstruieren kann, der die diskrete Projektion von $\mathrm{L}(\mathfrak{A})$ erkennt (siehe [AD90], vgl. 5.1.6.2)) und daß das Universalitätsproblem für diskrete Rabin-Scott- und Büchi-Automaten entscheidbar ist (siehe z. B. [HU79] bzw. [Tho90a]).

Wir stellen die Reduktion für endliche Wörter vor. In den beiden anderen Fällen verfährt man in gleicher Weise.

Nehmen wir an, der Uhrenindex einer Sprache (die oben angegebene Funktion) wäre berechenbar und $\mathfrak{A}$ wäre ein Rabin-Scott-Uhrenautomat über A, für den wir überprüfen wollten, ob $\mathrm{L}(\mathfrak{A})$ alle endlichen Intervallwörter über A enthält. Dies könnten wir dann folgendermaßen tun.

Zuerst würden wir $i = \mathbf{U}(\mathfrak{A})$ bestimmen. Falls $i > 0$ wäre, so könnte $\mathrm{L}(\mathfrak{A})$ nicht alle endlichen Intervallwörter über A enthalten, da der Uhrenindex der Sprache aller endlichen Intervallwörter offensichtlich Null ist. Wäre $i = 0$, so müßte mit jedem Wort u auch jedes Wort u' mit gleicher Beschriftung in $\mathrm{L}(\mathfrak{A})$ liegen, d. h., es würde $\mathrm{L}(\mathfrak{A})$ mit dem W-Urbild der diskreten Projektion von $\mathrm{L}(\mathfrak{A})$ übereinstimmen. Also würde in diesem Fall $\mathrm{L}(\mathfrak{A})$ genau dann alle endlichen Intervallwörter enthalten, wenn die diskrete Projektion von $\mathrm{L}(\mathfrak{A})$ mit A^+ übereinstimmen würde. Um das zu entscheiden, würde

man zu einem diskreten Automaten für die diskrete Projektion von $\mathrm{L}(\mathfrak{A})$ übergehen (siehe 5.1.6) und den dort entscheidbaren Universalitätstest durchführen. ■

4.3.3 Korollar (Nichtberechenbarkeit von minimalen Automaten) *Es gibt keine berechenbare Funktion, die zu jedem Uhrenautomaten $\mathfrak{A}$ einen äquivalenten Uhrenautomaten $\mathfrak{A}'$ mit $\mathbf{U}(\mathfrak{A}') = \mathbf{U}(\mathrm{L}(\mathfrak{A}))$ konstruiert.*

Damit ist das Minimierungsproblem für Uhrenautomaten unentscheidbar, wenn man den Uhrenindex als Maß für die Komplexität einer uhrenerkennbaren Sprache heranzieht.

4.3.2 Der Hierarchiesatz

In diesem Unterabschnitt werden grundsätzlich Sprachen unendlicher Intervallwörter betrachtet. Alle Definitionen, Beispiele, Lemmas usw. übertragen sich unmittelbar auf Sprachen endlicher Intervallwörter. Es muß lediglich jeweils eine Zusatzüberlegung für das Wortende angestellt werden.

Um den Hierarchiesatz einfach formulieren zu können, führen wir neue Bezeichnungen ein.

4.3.4 Definition (Uhrenhierarchie) Für jede natürliche Zahl m ist $\mathbf{NTL}^m$ die Menge aller uhrenerkennbaren Sprachen mit Uhrenindex kleiner als oder gleich m.

Wir wollen zeigen, daß $\mathbf{NTL}^0, \mathbf{NTL}^1, \mathbf{NTL}^2, \ldots$ eine echte, unendliche Hierarchie bildet:

4.3.5 Uhrenhierarchiesatz *Es gilt*

$$\mathbf{NTL}^0 \subsetneqq \mathbf{NTL}^1 \subsetneqq \mathbf{NTL}^2 \subsetneqq \ldots \tag{4.6}$$

Zur Vorbereitung des Beweises beginnen wir mit der Beschreibung der „Zeugen“ für die Echtheit der Kette in (4.6).

4.3.6 Beispiel Die unäre Sprache L_m enthalte alle unendlichen Intervallwörter $\underline{w}$, die folgende Eigenschaft besitzen: Für jedes $j \in \mathbf{N}$ gilt $W_{j+m} - W_j = 1$.

In Abbildung 4.2 ist ein Automat dargestellt, der L_m erkennt und mit $\mathfrak{A}^m$ bezeichnet wird. Er benutzt m Uhren, woraus $\mathbf{U}(L_m) \leq m$ folgt.

4.3.7 Lemma *Für jedes $m \geq 0$ gilt*

$$\mathbf{U}(L_m) \;=\; m\,.$$

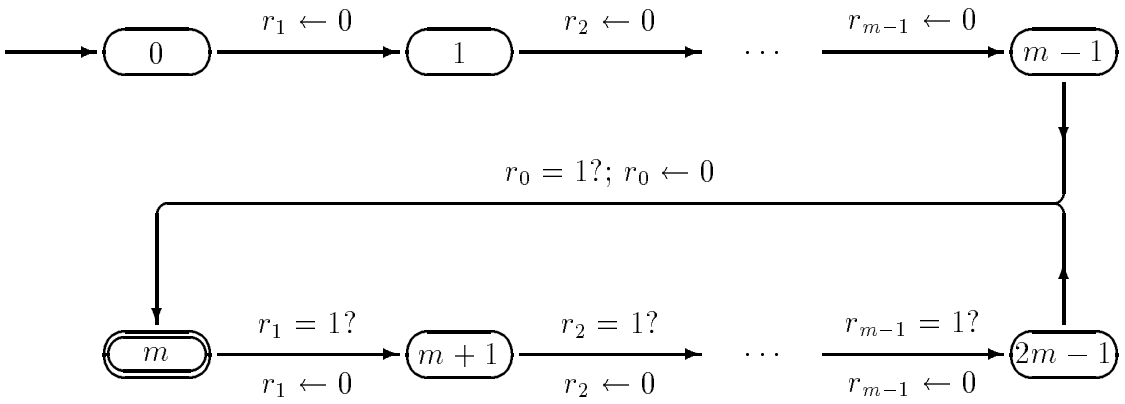


Abbildung 4.2: Der Uhrenautomat $\mathfrak{A}^m$, der $\mathbf{NTL}^{m-1} \subsetneqq \mathbf{NTL}^m$ belegt

Dargestellt ist der Automat $\mathfrak{A}^m$, der die Sprache L_m aus 4.3.6 erkennt, für die $L_m \in \mathbf{NTL}^m \setminus \mathbf{NTL}^{m-1}$ gilt und die damit die Echtheit der Uhrenhierarchie belegt. Jeder Pfeil ist mit der zusätzlichen Beschriftung a: $\dot{r}_0 = \ldots = \dot{r}_{m-1} = 1$ zu versehen. Diese wurde der Übersichtlichkeit halber weggelassen.

Beweis. Aus Beispiel 4.3.6 erhalten wir die Abschätzung $\mathbf{U}(L_m) \leq m$ für jedes $m \in \mathbf{N}$. Für die Abschätzung in die andere Richtung führen wir die Annahme des Gegenteils zum Widerspruch.

Nehmen wir im Widerspruch zur Behautpung an, es würde $\mathbf{U}(L_m) < m$ gelten. Dann gäbe es einen Uhrenautomaten $\mathfrak{A}$ mit $m-1$ Uhren, der L_m erkennen würde. Wir betrachten das Wort $\underline{w}$ mit $w_i = \frac{1}{m}$ für jedes $i \in \mathbf{N}$. Es gehört offensichtlich zu L_m, womit es eine akzeptierende Berechnung von $\mathfrak{A}$ auf $\underline{w}$ geben müßte. Nehmen wir an, diese Berechnung wäre durch die Transitionsfolge $\underline{\delta}$ gegeben und es gehörten die Folgen $\underline{\rho}$ und $\underline{\rho}'$ von Uhrenständen dazu.

Das Wort $\underline{w}$ enthält im Intervall $[0,1)$ genau m Positionen, nämlich $0, \ldots, m-1$. Deshalb tritt einer der beiden folgenden Fälle ein:

1) Für jede Uhr r gibt es eine Position j mit $0 < j < m$ und $\rho_j(r) = 0$.

2) Es gibt eine Position j mit $0 < j < m$ und der Eigenschaft: Für jede Uhr r mit $\rho_j(r) = 0$ gibt es eine Position k mit $j < k < m$, für die $\rho_k(r) = 0$ gilt.

Nach diesen beiden Fällen führen wir eine Fallunterscheidung durch.

Fall 1) Wir betrachten $\underline{w}' = \frac{1}{2m}, \frac{1}{m}, \frac{1}{m}, \ldots$ und behaupten, daß durch $\underline{\delta}$ eine akzeptierende Berechnung von $\mathfrak{A}$ auf $\underline{w}'$ gegeben wäre. Seien $\underline{\sigma}$ und $\underline{\sigma}'$ die Folgen der Uhrenstände, die durch $\underline{\delta}$ für $\underline{w}'$ vorgegeben wären. Für $k < m$ würde sowohl $\rho'_k(r) < 1$ wie auch $\sigma'_k(r) < 1$ für jede Uhr r gelten. Für $k \geq m$ hätte man wegen der Voraussetzung und $w_k = w'_k$ die Gleichheit $\rho'_k(r) = \sigma'_k(r)$ für jede Uhr r. Damit wären die

Folgen ρ' und σ' äquivalent, wodurch nach Bemerkung 4.1.1 auch $\underline{w'}$ von $\mathfrak{A}$ akzeptiert würde, aber es gilt $\underline{w'} \notin L$. Widerspruch!

Fall 2) Wir betrachten $\underline{w'}$ mit $w_{j-1} = \frac{1}{2m}$ sowie $w_j = \frac{3}{2m}$ und $w'_k = \frac{1}{m}$ für $k < j-1$ und $k > j$. Wie im ersten Fall läßt sich zeigen, daß für $k < m$ sowohl $\rho'_k(r) < 1$ wie auch $\sigma'_k(r) < 1$ für jede Uhr r gelten würde. Ist r eine Uhr mit $\rho_j(r) = 0$, so folgt aus der Voraussetzung $\rho'_{m-1}(r) = \sigma'_{m-1}(r)$, da r zwischen j und m noch einmal zurückgesetzt würde. Daraus würde dann aber auch $\rho'_k(r) = \sigma'_k(r)$ für alle $k \geq m$ folgen. Ist r eine Uhr mit $\rho_j(r) \neq 0$, so gilt $\rho'_j(r) = \sigma'_j(r)$ und es würde $\rho'_k(r) = \sigma'_k(r)$ für jedes $k > j$ folgen. Damit wären die Folgen $\underline{\rho'}$ und $\underline{\sigma'}$ äquivalent, wodurch $\underline{w'}$ von $\mathfrak{A}$ akzeptiert würde. Aber es gilt wieder $\underline{w'} \notin L_m$. Widerspruch! ■

Damit ist auch der Hierarchiesatz 4.3.5 bewiesen. ■

Aus dem Beweis erhalten wir noch einen Alternativbeweis für die Behauptung, daß **NTL** nicht unter Komplementbildung abgeschlossen ist (siehe 1.4.6).

4.3.8 Beispiel Wir betrachten die Sprache L, die alle unendlichen unären Intervallwörter $\underline{w}$ enthält, für die es eine Position i mit $W_i + 1 \notin \{W_0, W_1, W_2, \ldots\}$ gibt. Die Sprache L enthält also ein unendliches Wort, wenn es in ihm einen Moment x gibt, für den gilt, daß $x+1$ kein Moment des Wortes ist.

Die Sprache L ist uhrenerkennbar, denn sie wird durch den folgenden $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$-Satz definiert:

$$\exists X \exists x (Xx \wedge \neg \exists y (x < y \wedge Xy) \wedge \forall y (x < y \rightarrow \mathrm{d}(X, y) \neq 1)\,.$$

Man stelle sich x als die gesuchte Position i und X als die Einermenge $\{i\}$ vor.

Das Komplement von L enthält ein unendliches Intervallwort, wenn es mit jedem Moment x auch den Moment $x+1$ hat, d. h., es enthält alle Intervallwörter $\underline{w}$, für die $W_i + 1$ für jedes i in der Menge $\{W_0, W_1, W_2, \ldots\}$ liegt.

4.3.9 Lemma *Das Komplement von L aus 4.3.8 ist nicht uhrenerkennbar.*

Beweis. Würde das Komplement von L von einem Uhrenautomaten $\mathfrak{A}$ mit $m-1$ Uhren erkannt werden, wo müßte $\mathfrak{A}$ auch das Wort $\underline{w}$ aus dem Beweis von 4.3.7 akzeptieren. Mit den gleichen Argumenten wie in dem Beweis von 4.3.7 würde dann auch das dort konstruierte Wort $\underline{w'}$ akzeptiert werden, was aber nicht zum Komplement von L gehört. ■

4.4 Zweiwege-Uhrenautomaten von Alur und Henzinger

In der Einleitung der vorliegenden Arbeit hatten wir schon auf das Problem hingewiesen, das von Alur und Henzinger in [AH91a] gestellt wurde: Man finde eine Klasse von Sprachen von Intervallwörtern, die

(Bool) gegen boolesche Operationen abgeschlossen ist,

(Leer) ein elementar entscheidbares Leerheitsproblem besitzt und

(Max) „maximal“ mit diesen Eigenschaften ist.

Offensichtlich ist mit (Bool) der effektive (!) Abschluß unter den booleschen Operationen gemeint, woraus mit (Leer) auch die Entscheidbarkeit des Universalitäts- und des Inklusionsproblems folgen würde. Außerdem ist es sinnvoll (und sicherlich auch im Sinne von Alur und Henzinger), zusätzlich die folgende Forderung aufzustellen:

(Eff) Die Klasse soll durch einen entscheidbaren Beschreibungsformalismus gegeben sein.

Dieser Bedingung sollte jeder „vernünftige“ Formalismus genügen.

Die Klasse **DTL** aller deterministisch uhrenerkennbaren Sprachen erfüllt die Bedingungen (Bool), (Leer) und (Eff). Ebenso werden diese drei Bedingungen von den Sprachen erfüllt, die durch Formeln der Logik MITL$^{\mathrm{p}}$ (siehe Abschnitt 5.4) definiert werden. Doch beide Klassen sind unvergleichbar und beide enthalten fundamentale Beispielsprachen, die jeweils nicht zu der anderen Klasse gehören, womit man berechtigterweise sagen kann, daß für beide die Bedingung (Max) nicht erfüllt ist.

Alur und Henzinger definieren in Zusammenhang mit der Zeitlogik MITL$^{\mathrm{p}}$ in ihrer Arbeit [AH92] ein Modell eines Zweiwege-Uhrenautomaten. Sie schränken ihre Betrachtungen auf solche Automaten ein, die deterministisch sind und die eine obere Schranke für die Anzahl der Lesevorgänge jedes Ereignisses besitzen. Wir wollen die Klasse der Sprachen, die durch diese Automaten erkannt werden, mit $\mathbf{2DTL}_\omega$ bezeichnen. Alur und Henzinger zeigen, daß $\mathbf{2DTL}_\omega$ gegen boolesche Operationen abgeschlossen ist und daß das Leerheitsproblem für die Automaten in exponentieller Zeit gelöst werden kann. Damit sind (Bool) und (Leer) erfüllt. Darüberhinaus gilt die Inklusion $\mathbf{DTL} \subseteq \mathbf{2DTL}_\omega$ und sie zeigen ferner, daß jede durch eine MITL$^{\mathrm{p}}$-Formel definierte Sprache zu $\mathbf{2DTL}_\omega$ gehört. Das deuten sie als einen Beleg für (Max). Insgesamt sehen sie $\mathbf{2DTL}_\omega$ als einen geeigneten Kandidaten für die Beantwortung ihrer Frage an. Wir werden jedoch nachweisen, daß $\mathbf{2DTL}_\omega$ der Bedingung (Eff) nicht genügt.

Es wird hier auf eine formale Definition des von Alur und Henzinger vorgeschlagenen Zweiwege-Modells verzichtet — dafür wird auf die angegebene Quelle verwiesen —, stattdessen wird seine Funktionsweise mit ausreichender Präzision informell erläutert.

Wie ein herkömmlicher endlicher Zweiwegeautomat aus einem Rabin-Scott-Automaten entsteht, so entsteht auch ein Zweiwege-Uhrenautomat aus einem herkömmlichen Uhrenautomaten. D. h., in jeder Transition wird jeweils noch die Laufrichtung für den nächsten Leseschritt angegeben und jedes Eingabewort mit Anfangs- und

Endmarken versehen. Läuft der Automat nach links, so laufen die Uhren nicht weiter, sondern zurück. Dadurch können auch negative Uhrenstände entstehen, weshalb als Testprädikate die Menge $\mathcal{J}$ zugelassen wird. D. h., Uhrenstände können auch mit negativen ganzen Zahlen verglichen werden.

Determinismus bedeutet hier wie üblich, daß in jeder Konfiguration des Automaten jeweils höchstens eine Transition bei vorgegebenem, zu lesendem Ereignis anwendbar ist.

Alur und Henzinger sprechen von einem *k-beschränkten* Zweiwege-Uhrenautomaten, wenn jeder Lauf des Automaten die Eigenschaft hat, daß er jede Stelle höchstens $(2k+1)$-mal passiert (also jedes Ereignis höchstens $(2k+1)$-mal liest). Die Klasse der von deterministischen k-beschränkten Automaten erkannten Sprachen wird mit $\mathbf{2DTL}_k$ bezeichnet, insbesondere gilt also $\mathbf{2DTL}_0 = \mathbf{DTL}$. Die Klasse der *beschränkten* Zweiwege-Uhrenautomaten enthält diejenigen Zweiwege-Uhrenautomaten, die für eine natürliche Zahl k die Bedingung erfüllen, daß sie k-beschränkt sind. Die Klasse der durch sie erkannten Sprachen wird mit $\mathbf{2DTL}_\omega$ bezeichnet, d. h.,

$$\mathbf{2DTL}_\omega \;=\; \bigcup_{k<\omega} \mathbf{2DTL}_k\,.$$

Neben den oben aufgeführten Ergebnissen beweisen Alur und Henzinger die Korrektheit der folgenden Inklusionskette:

$$\mathbf{2DTL}_0 \subsetneqq \mathbf{2DTL}_1 \subsetneqq \mathbf{2DTL}_2 \subsetneqq \ldots \subsetneqq \mathbf{NTL}\,. \tag{4.7}$$

Dazu muß gesagt werden, daß in [AH92] für die Definition von **NTL** ein Automatenmodell benutzt wird, das gegenüber dem Modell des gewöhnlichen Uhrenautomaten erweitert ist. Damit umgehen die Autoren Schwierigkeiten, die bei der Simulation eines deterministischen k-beschränkten Uhrenautomaten durch einen nichtdeterministischen Uhrenautomaten auftreten. Sie treffen keine Aussage darüber, ob das allgemeinere Automatenmodell zu dem üblichen äquivalent ist. Wir werden mit Hilfe des Charakterisierungssatzes in Unterabschnitt 5.1.4 eine stärkere Inklusionsbeziehung beweisen, aus der insbesondere $\mathbf{2DTL}_\omega \subsetneqq \mathbf{NTL}$ im eigentlichen Sinne folgt (siehe 4.5.2).

Hier sei angemerkt, daß in [AH92] nur Sprachen endlicher Intervallwörter betrachtet werden. Die Autoren weisen darauf hin, daß die meisten Ergebnisse — und dazu zählen auch die hier genannten — unmittelbar auf unbeschränkte und unendliche Intervallwörter übertragen werden können.

Wir kommen nun zu dem angekündigten Ergebnis über die Nichteffektivität von $\mathbf{2DTL}_\omega$.

4.4.1 Satz *Relativ zur Klasse aller deterministischen Zweiwege-Uhrenautomaten ist die Klasse der beschränkten deterministischen Zweiwege-Uhrenautomaten nicht entscheidbar. (Dabei wird vorausgesetzt, daß das Alphabet mindestens drei Elemente besitzt.)*

D. h., es ist nicht entscheidbar, ob ein gegebener, deterministischer Zweiwege-Uhrenautomat beschränkt ist.

Beweis. Wir reduzieren das Problem, ob eine Zweiregistermaschine eine haltende Berechnung hat, auf das gegebene. Zuerst beschreiben wir eine Reduktion für den Fall der endlichen Wörter. Im Anschluß daran gehen wir auf unendliche bzw. unbeschränkte Intervallwörter ein.

Wir benötigen den Begriff der *Kodierung* einer Berechnung einer Zweiregistermaschine, der auch schon in [Alu91] benutzt wird, um die Unentscheidbarkeit des Universalitätsproblems für Uhrenautomaten nachzuweisen.

Wir nehmen dabei an, daß jedes Programm M einer Zweiregistermaschine die folgende Gestalt hat:

$$0\,\alpha_0;\ 1\,\alpha_1;\ \ldots;\ r-1\,\alpha_{r-1};\ r\,\texttt{stop}\,. \tag{4.8}$$

Dabei sei jede Anweisung α_i von der Form `x:=x+1`, `y:=y+1`, `if x=0 then` j `else x:=x-1` oder `if y=0 then` j `else y:=y-1` mit $j < r$. O. B. d. A. nehmen wir an, daß kein Sprung in die gleiche Zeile führt.

Wir benutzen der Übersichtlichkeit halber das Alphabet $A = \{x, y, 0, \ldots, r\}$ mit zwei unterschiedlichen, von allen Elementen von $r+1$ verschiedenen Buchstaben x und y. Man kann sich überlegen, daß man bei einer sparsameren Kodierung mit einem dreielementigen Alphabet auskommt (vgl. auch den Beweis der Unentscheidbarkeit des Universalitätsproblems für Uhrenautomaten in [AD]).

Eine Kodierung der Berechnung eines solchen Programmes ist nun ein endliches Intervallwort $u = v_0 \ldots v_s$ über A, das sich aus Teilwörtern v_l zusammensetzt, die die folgende Gestalt haben:

$$v_l \;=\; (i,1)(x,w_0)\ldots(x,w_p)(y,w'_0)\ldots(y,w'_{p'})\,,$$

wobei $w_0+\ldots+w_p = 1$ und $w'_0+\ldots+w'_{p'} = 1$ gelte. Außerdem werden in Abhängigkeit von i Bedingungen an v_{l+1} mit

$$v_{l+1} \;=\; (i',1)(x,t_0)\ldots(x,t_q)(y,t'_0)\ldots(y,t'_{q'})\,,$$

gestellt:

- Ist α_i die Anweisung `x:=x+1`, so gelte $i' = i+1$, $q = p+1$, $q' = p'$ und $t_k = w_k$ für $k < p$ sowie $t'_k = w'_k$ für $k \le p'$.
- Ist α_i die Anweisung `if x=0 then` j `else x:=x-1`, dann unterscheiden wir zwei Fälle:
  - Ist $p > 0$, dann gelte $i' = i+1$, $q = p-1$, $q' = p'$ und $t_k = w_k$ für $k < p-1$ sowie $t'_k = w'_k$ für $k \le p'$.

– Ist $p = 0$, dann gelte $i' = j$, $q = p$, $q' = p'$ und $t_k = w_k$ für $k \leq p$ sowie $t'_k = w'_k$ für $k \leq p'$.

- Anweisungen mit y werden entsprechend behandelt.

Damit kodiert jedes Wort v_l eine Konfiguration der Zweiregistermaschine, wobei die Registerinhalte durch die Anzahl der mit x bzw. y beschrifteten Ereignisse (p bzw. q) wiedergegeben werden. Die Dauervorgaben, die eine Korrespondenz zwischen den Positionen in v_l und v_{l+1} herstellen, ermöglichen die Konstruktion eines deterministischen Zweiwege-Uhrenautomaten $\mathfrak{A}$, der die Kodierungen der Anfangsstücke von Berechnungen von M erkennt: Die Überprüfung der Beschriftungen erledigt er mit seiner endlichen Kontrolle und das Erhöhen und Vermindern der Register wird durch häufiges Hin- und Herlaufen zur Überprüfung der Korrespondenz erledigt. (Vgl. die Ausführungen von Alur und Henzinger in [AH92] zur Arbeitsweise von Zweiwege-Uhrenautomaten und den Beweis der Unentscheidbarkeit des Universalitätsproblems für Uhrenautomaten in [AD].) Dabei ist die Anzahl der Passagen einer Stelle in v_l linear abhängig von $p + q$. Handelt es sich also um eine haltende Registermaschine, dann ist der Automat beschränkt. Umgekehrt gilt: Ist der Automat nicht beschränkt, dann hält die Registermaschine nicht. Ist er doch beschränkt, dann gibt es eine obere Schranke für die Registerwerte von M, so daß man durch Simulation der Maschine und sukzessives Heraufsetzen einer möglichen Schranke für den maximalen Registerwert effektiv feststellen kann, ob die Maschine hält oder in eine Endlosschleife läuft. Damit könnten wir unter der Voraussetzung, daß sich die Eigenschaft eines deterministischen Zweiwege-Uhrenautomaten, beschränkt zu sein, effektiv bestimmen ließe, entscheiden, ob eine Zweiregistermaschine hält. Widerspruch!

Im Falle der unendlichen Intervallwörter konstruiert man einen Automaten, der die Menge aller unendlichen Intervallwörter über dem angegebenen Alphabet enthält, die mit einem Anfangsstück einer Kodierung einer Berechnung der gegebenen Zweiregistermaschine anfangen und dann trivial zu einem unendlichen Intervallwort fortgesetzt werden (z. B. durch Anhängen von $(a, 1)^\omega$.) Dann führt die gleiche Argumentation zum Erfolg. ■

4.5 Eindeutige Uhrenautomaten

Auf der Suche nach einer Teilklasse von **NTL**, die die Eigenschaften (Bool) bis (Eff) erfüllt, wenden wir uns nun den ‚eindeutigen' Uhrenautomaten zu.

4.5.1 Definition (Eindeutigkeit) Ein Uhrenautomat heißt *eindeutig*, wenn es für jedes akzeptierte Wort genau eine akzeptierende Berechnung gibt. Eine Sprache von Intervallwörtern ist *eindeutig uhrenerkennbar*, wenn sie von einem eindeutigen Uhrenautomaten erkannt wird. Die Klasse aller eindeutig uhrenerkennbaren Sprachen wird mit **UTL** bezeichnet.

Offensichtlich ist die Klasse **UTL** gegen disjunkte Vereinigung und Durchschnittbildung abgeschlossen. Wäre sie gegen Komplementbildung abgeschlossen, so auch gegen die Bildung beliebiger (endlicher) Vereinigungen. Beides ist bislang offen. Wir werden jedoch zeigen, daß eine der typischen uhrenerkennbaren Sprachen, die zu $\mathbf{NTL} \setminus \text{co-}\mathbf{NTL}$ gehört und damit den Nichtabschluß von **NTL** unter Komplementbildung belegt, nicht eindeutig uhrenerkennbar ist. Deshalb fehlen vorerst geeignete Kandidaten, um beweisen zu können, daß **UTL** nicht unter Komplementbildung abgeschlossen ist. In diesem Zusammenhang ist auch erwähnenswert, daß die Automaten, die in dem Beweis der Unentscheidbarkeit des Universalitätsproblems für Uhrenautomaten von Alur und Dill in [AD] konstruiert werden, nicht eindeutig sind und wesentlich von Nichtdeterminismus Gebrauch machen.

Außerdem beweisen wir in diesem Abschnitt, daß **UTL** die Bedingung (Eff) erfüllt. Bedingung (Leer) ist trivialerweise erfüllt, da eindeutige Uhrenautomaten auch nichtdeterministische Automaten sind und für diese (Leer) gilt. Als Argument dafür, daß **UTL** der Bedingung (Max) genügt, werden wir die Gültigkeit der Inklusion $\mathbf{2DTL}_\omega \subseteq \mathbf{UTL}$ nachweisen, allerdings erst in Unterabschnitt 5.1.4. Wir wollen den entsprechenden Satz jedoch schon hier formulieren:

4.5.2 Satz (Beweis in Unterabschnitt 5.1.4) *Es gilt* $\mathbf{2DTL}_\omega \subseteq \mathbf{UTL}$.

Wir beginnen mit der Untersuchung der Eindeutigkeitseigenschaft.

4.5.1 Entscheidbarkeit der Eindeutigkeitseigenschaft für Uhrenautomaten

4.5.3 Lemma *Es ist entscheidbar, ob ein gegebener Uhrenautomat eindeutig ist oder nicht.*

Beweis. Wir betrachten noch einmal die Formel akz aus Abschnitt 2.4, die äquivalent zu einem gegebenen Σ-Automaten $\mathfrak{A}$ ist. Sie ist von der Form

$$\mathsf{akz} \triangleq \exists X_0 \ldots \exists X_{s-1} \exists Y_0 \ldots \exists Y_{t-1} \mathsf{akzBer}.$$

Aus der Konstruktion der Formel ist klar, daß $\mathfrak{A}$ genau dann nicht eindeutig ist, wenn

$$\mathsf{akzBer} \triangleq \mathsf{akzBer}(X_0, \ldots, X_{s-1}, Y_0, \ldots, Y_{t-1})$$

zwei erfüllende Belegungen hat, mit anderen Worten, wenn die Formel

$$\mathsf{eindeutig} = \mathsf{akzBer}(\underline{X}, \underline{Y}) \wedge \mathsf{akzBer}(\underline{X}', \underline{Y}') \wedge \left(\bigvee_{i<s} X_i \neq X_i' \vee \bigvee_{j<t} Y_j \neq Y_j' \right)$$

erfüllbar ist. (Es steht $X \neq Y$ für $\exists x((Xx \wedge \neg Yx) \vee (\neg Xx \wedge Yx))$.) Dies ist aber entscheidbar, wie wir im nächsten Kaptel sehen werden (siehe 5.1.6). Da die Formel akz

und damit auch die Formel eindeutig nach 2.4.1 effektiv aus einem Uhrenautomaten gewonnen werden können, kann die Eindeutigkeit des gegebenen Automaten $\mathfrak{A}$ also effektiv getestet werden. ■

4.5.2 Eine erblich mehrdeutige Sprache

Im Rest des Abschnitts wollen wir von der im folgenden Beispiel definierten Sprache L_{ne} zeigen, daß sie nicht eindeutig uhrenerkennbar ist.

4.5.4 Beispiel Es sei L_{ne} die Sprache aller endlichen Intervallwörter u, die

1) $||u|| < 2$ erfüllen,

2) sich in der Form $u = xyz$ mit $||x|| > 0$ und $||y|| = 1$ schreiben lassen und

3) eine Zerlegung der Form $u = xy$ mit $||x|| = 1$ erlauben.

Die Bedingungen 2) und 3) lassen sich auch in einer etwas anderen Weise formulieren, wenn man u in der Form $\underline{w}$ annimmt:

2*) Es gibt zwei Positionen i und j mit $0 < i < j$, so daß $W_j - W_i = 1$ gilt.

3*) Es gibt eine Position i mit $W_i = 1$.

Die Sprache L_{ne} ist uhrenerkennbar. In Abbildung 4.3 ist ein entsprechender Uhrenautomat zu finden.

4.5.5 Lemma *Die Sprache L_{ne} ist nicht eindeutig uhrenerkennbar.*

Die Mehrdeutigkeit von L_{ne} ist darin begründet, daß man die Zerlegung eines gegebenen Wortes in xyz mit $||y|| = 1$ und $||x|| > 0$ (nichtdeterministisch) raten muß und nicht eindeutig bestimmen kann. Der formale Beweis erstreckt sich über mehrere Seiten. Ihm liegt im wesentlichen eine Analyse der Menge der akzeptierenden Berechnungen der die Sprache L_{ne} erkennenden Automaten zugrunde. Es wird gezeigt, wie man eine Berechnung mit einem diskreten Wort identifizieren kann, und benutzt, daß die Menge der so entstehenden diskreten Wörter regulär ist.

Beweis. Wir beginnen mit einer Definition, die für den Beweis hilfreich ist. Ein Uhrenautomat heiße *normiert*, wenn er L_{ne} erkennt und die beiden folgenden Eigenschaften besitzt:

1) Es gibt eine Uhr, die durch keine Transition zurückgesetzt wird.

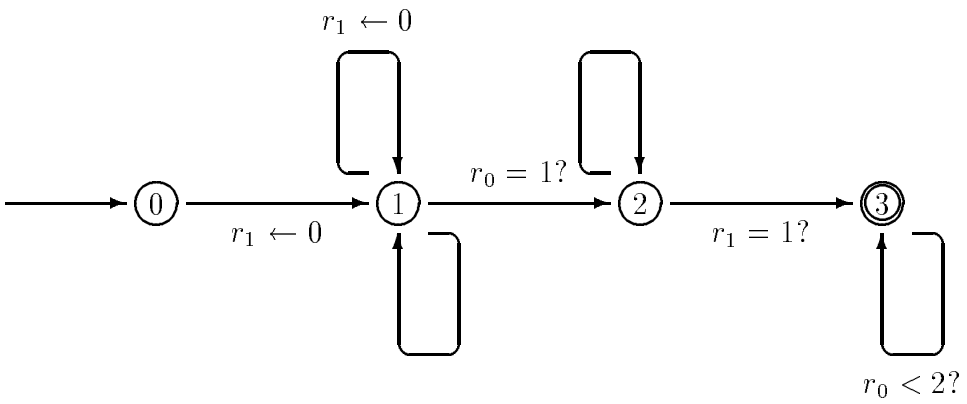


Abbildung 4.3: Ein Uhrenautomat, der eine erblich mehrdeutige Sprache erkennt

Dargestellt ist ein Uhrenautomat, der die uhrenerkennbare Sprache L_{ne} aus 4.5.4 erkennt, die von keinem eindeutigen Uhrenautomaten erkannt wird. Jeder Pfeil ist zusätzlich mit a: $\dot{r}_0 = \dot{r}_1 = 1$ zu versehen.

2) Ist durch die Transitionsfolge $\underline{\delta}$ eine akzeptierende Berechnung auf einem Wort $\underline{w}$ aus L_{ne} gegeben und ist i die Position mit $W_i = 1$, so setzt keine der Transitionen δ_j mit $j \geq i$ eine Uhr zurück.

Auf die Betrachtung von normierten Automaten werden wir uns im folgenden einschränken, was unwesentlich ist:

Behauptung 1 Falls es einen eindeutigen Uhrenautomaten gäbe, der L_{ne} erkennen würde, dann gäbe es auch einen normierten, eindeutigen Automaten, der L_{ne} erkennen würde.

Beweis. Wir nehmen an, es gäbe einen eindeutigen Automaten $\mathfrak{A}$, der L_{ne} erkennen würde, und konstruieren aus ihm einen äquivalenten normierten, eindeutigen Automaten $\mathfrak{A}'$.

Um Bedingung 1) genüge zu tun, wird einfach eine neue Uhr zu der bestehenden Uhrenmenge von $\mathfrak{A}$ hinzugefügt, ohne daß die Transitionen oder andere Teile von $\mathfrak{A}$ geändert werden. Um Bedingung 2) Rechnung zu tragen, simuliert der Automat $\mathfrak{A}'$ den Automaten $\mathfrak{A}$, wobei er jedoch auf Rücksetzungen nach dem Zeitpunkt Eins verzichtet. Von dem Zeitpunkt ab, an dem die erste Rücksetzung einer Uhr r erfolgt wäre, wird r in $\mathfrak{A}'$ nicht mehr benutzt. Der Automat $\mathfrak{A}'$ merkt sich in der endlichen Kontrolle, daß $\mathfrak{A}$ die Uhr r zurückgesetzt hätte, und jeder folgende Registertest wird ausgewertet, als hätte r einen beliebigen Wert zwischen Null und Eins angenommen. Diese Annahme ist aufgrund der Längenbeschränkung aller Wörter (4.5.4.1)) aus L_{ne} korrekt.

Formal sieht die Konstruktion von $\mathfrak{A}' = (Q', R', s', I', F')$ aus $\mathfrak{A} = (Q, R, s, I, F)$ wie folgt aus: Die Menge R' entstehe aus R durch Hinzufügen einer neuen Uhr r_0. Die Menge Q' sei das kartesische Produkt aus Q und der Potenzmenge von R, also $Q' = Q \times \wp(R)$. Weiterhin sei $s' = (s, \{\emptyset\})$ und $F' = F \times \wp(R)$. Für jede Transition (q, a, ϕ, ζ, q') in Δ gehören alle Transitionen der Form

$$((q, M), a, \phi' \wedge \psi, \zeta', (q', M'))$$

zu Δ', die eine der folgenden Bedingungen erfüllen:

- Es ist $\psi \triangleq r_0 \leq 1$, und es gelten $\phi' = \phi$ und $\zeta = \zeta'$ sowie $M = M'$.
- Es ist $\psi \triangleq r_0 > 1 \wedge r_0 < 2$ und $\zeta' = \emptyset$ sowie $M' = M \cup \mathrm{dom}(\zeta)$, und ϕ' entsteht aus ϕ durch Ersetzen jedes atomaren Tests $r_i \sim d$ mit $r_i \in M$ durch den Wahrheitswert von $0.5 \sim d$. (Hier steht 0.5 stellvertretend für alle Werte zwischen 0 und 1.)

Offensichtlich erfüllt der in dieser Weise konstruierte Automat die beiden Forderungen, ist äquivalent zu dem ursprünglichen und ist eindeutig, falls es der ursprüngliche war. □

Im Rest des Beweises nehmen wir an, daß ein normierter Uhrenautomat $\mathfrak{A}$ mit Registersatz $R = \{r_0, \ldots, r_{s-1}\}$ gegeben sei, wobei r_0 eine Uhr sei, die nicht zurückgesetzt werde.

Jeder akzeptierenden Berechnung von $\mathfrak{A}$ wird jetzt ein diskretes Wort zugeordnet, in dem die wichtigsten Informationen aus der Berechnung festgehalten werden: an welcher Position eine Uhr zum letzten Mal den Wert Null zeigt und an welcher Position sie zum ersten Mal einen Wert größer oder gleich Eins hat. Dabei wird unterschieden, ob eine Uhr (mindestens) einmal genau den Wert Eins zeigt oder nie.

Wir benutzen $3s - 1$ Marken: $N_0, \ldots, N_{s-1}, E_0, \ldots, E_{s-1}, g_1, \ldots, g_{s-1}$. Dabei steht die Marke N_j für „r_j zeigt zum letzten Mal den Wert Null“, die Marke E_j für „r_j zeigt den Wert Eins“ und die Marke g_j für „r_j hat zum ersten Mal einen Wert größer als oder gleich Eins und dieser ist echt größer als Eins“. Die Menge B bestehe aus allen Mengen, die aus diesen Marken gebildet werden können.

Sei nun $\underline{\delta}$ eine Transitionsfolge, die eine akzeptierende Berechnung mit zugehörigen Uhrenständen $\underline{\rho}$ und $\underline{\rho'}$ auf einem Wort $\underline{w}$ bestimmt. Das *von $\underline{\delta}$ und $\underline{w}$ induzierte Wort* $\underline{b}$ über B habe die gleiche Länge wie $\underline{w}$ und für jedes $i < |\underline{w}|$ sei die Menge der Marken, die zu b_i gehören, durch die folgenden Regeln bestimmt:

- Falls $i + 1$ die größte Zahl mit $\rho(r_j) = 0$ ist, so sei $N_j \in b_i$.
- Falls $\rho'_i(r_j) = 1$ gilt, so sei $E_j \in b_i$.
- Falls $\rho'_i(r_j) > 1$ und $\rho'_{i-1}(r_j) < 1$ gelten, so sei $g_j \in b_i$.

Die Menge der in dieser Weise erhaltenen Wörter werde mit L' bezeichnet. (Beachte, daß L' von $\mathfrak{A}$ abhängt.)

Wir sagen, daß ein Wort u über B zu einem Wort $\underline{w}$ aus L_{ne} *paßt*, wenn es eine Transitionsfolge $\underline{\delta}$ gibt, so daß u das von $\underline{\delta}$ und $\underline{w}$ induzierte Wort ist. (Insbesondere muß dann $\underline{w}$ ein Element von L_{ne} und $\underline{\delta}$ eine konsistente Transitionsfolge sein, die eine akzeptierende Berechnung auf $\underline{w}$ bestimmt.) Um zu zeigen, daß L_{ne} nicht eindeutig uhrenerkennbar ist, brauchen wir also nur nachzuweisen, daß es in L_{ne} ein Wort gibt, zu dem zwei verschiedene Wörter aus L' passen.

Behauptung 2 Die Sprache L' ist regulär.

Beweis. Aus $\mathfrak{A}$ läßt sich ein Uhrenautomat konstruieren, der die Menge

$$L'' \;=\; \{(\underline{b}, \underline{w}) \mid \underline{b} \text{ paßt zu } \underline{w}\}$$

erkennt. (Das letzte Zurücksetzen einer Position rät der Automat nichtdeterministisch.) Dann gewinnt man die gewünschte Sprache, indem man die diskrete Projektion der Sprache L'' bildet. Daraus folgt aber die Regularität von L', da die diskrete Projektion jeder uhrenerkennbaren Sprache regulär ist ([AD90], siehe 5.1.6.1)). □

Behauptung 3 Für jedes Wort $\underline{b}$ aus L' gilt:

1) Es existiert genau eine Position p mit $E_0 \in b_k$.
2) Für jedes j kleiner als s gibt es höchstens eine Position i mit $N_j \in b_i$.
3) Für jedes j kleiner als s gibt es höchstens eine Position i mit $E_j \in b_i$ oder $g_j \in b_i$.

Beweis. Die erste Eigenschaft folgt aus der Bedingung, daß r_0 nicht zurückgesetzt wird und aus 4.5.4.3). Die zweite Eigenschaft ist durch die Konstruktion der Wörter aus L' gesichert, während die dritte Eigenschaft eine Folgerung aus der Forderung, daß nach dem Zeitpunkt Eins keine Uhr mehr zurückgesetzt werde, ist. □

Im folgenden sei nun $\underline{b}$ ein festes Wort aus L'. Für dieses können wir aufgrund der vorangehenden Behauptung die folgenden Festlegungen treffen.

Es sei p die eindeutige Zahl mit der Eigenschaft $E_0 \in b_{p-1}$. Es sei $N\colon s \to 1 + |\underline{b}|$ folgendermaßen definiert: Wenn es für j eine Position i mit $N_j \in b_i$ gibt, so sei $N(j) = i + 1$, andernfalls sei $N(j) = 0$. Weiterhin sei $E\colon s \to 1 + |\underline{b}|$ die partielle Funktion, die für jedes j definiert sei, für das es ein i mit $E_j \in b_i$ gibt. Dann sei $E(j) = i + 1$. Schließlich sei $g\colon s \to 1 + |\underline{b}|$ die partielle Funktion, die für jedes j definiert sei, für das es ein i mit $g_j \in b_i$ gibt. Dann sei $g(j) = i + 1$.

Mit diesen Bezeichnungen kann dann die Menge aller Positionen in $\underline{b}$, an denen nicht $\emptyset$ steht, beschrieben werden:

Behauptung 4 Es ist

$$\{N(j)-1 \mid j < s\} \cup \{E(j)-1 \mid E(j) \text{ ist definiert}\} \cup \{g(j)-1 \mid g(j) \text{ ist definiert}\}$$

die Menge der Positionen in $\underline{b}$, an denen nicht $\emptyset$ steht.

Beweis. Dies ist eine direkte Folgerung aus 3 und den Definitionen von N, E und g. □

In der folgenden Behauptung wird ein hinreichendes Kriterium dafür, daß ein Intervallwort zu einem gegebenen Wort aus L' paßt, formuliert:

Behauptung 5 Sei $\underline{w}$ ein Intervallwort aus L_{ne} mit der gleichen Länge wie $\underline{b}$. Außerdem seien die folgenden Bedingungen erfüllt:

1) Es ist $W_p = 1$.
2) Für jedes j kleiner als s gilt:
   (a) Sind weder $E(j)$ noch $g(j)$ definiert, so gilt $||\underline{w}|| - W_{N(j)} < 1$.
   (b) Ist $E(j)$ definiert, so gilt $W_{E(j)} - W_{N(j)} = 1$.
   (c) Ist $g(j)$ definiert, so gelten $W_{E(j)-1} - W_{N(j)} < 1$ und $W_{E(j)} - W_{N(j)} > 1$.

Dann paßt $\underline{w}$ zu $\underline{b}$.

Beweis. Da $\underline{b}$ aus L' stammt, gibt es ein Intervallwort $\underline{w'}$ aus L_{ne} und eine Transitionsfolge $\underline{\delta}$, so das $\underline{w'}$ die gleiche Länge wie $\underline{b}$ hat, $\underline{\delta}$ eine akzeptierende Berechnung auf $\underline{w'}$ bestimmt und $\underline{b}$ das durch $\underline{w'}$ und $\underline{\delta}$ induzierte Wort ist.

Durch $\underline{\delta}$ und $\underline{w'}$ sind Folgen $\underline{\rho}$ und $\underline{\rho'}$ von Uhrenständen bestimmt, genauso wie durch $\underline{\delta}$ und $\underline{w}$ Folgen $\underline{\sigma}$ und $\underline{\sigma'}$ bestimmt sind. Nach 4.1.1 braucht also lediglich gezeigt zu werden, daß $\underline{\sigma'}$ und $\underline{\rho'}$ äquivalent sind.

Die in der Behauptung genannten Bedingungen werden aufgrund der Konstruktion von $\underline{b}$ auch von $\underline{w'}$ erfüllt. (Man denke sich $\underline{w'}$ für $\underline{w}$.) Man kann sogar die Werte der Folge ρ' bis auf Äquivalenz aus $\underline{b}$ ablesen:

1) Für jedes $i < |\underline{w}|$ gilt $\rho'_i(r_j) < 2$.

2) Für jedes $j < s$ gilt:

   (a) Für alle $i < p-1$ gilt $0 < \rho'_i(r_j) < 1$.

   (b) Es ist $\rho'_{p-1}(r_j) = 1$ genau dann, wenn $E_j \in b_{p-1}$ gilt. Andernfalls ist $0 < \rho'_{p-1}(r_j) < 1$.

   (c) Ist weder $E(j)$ noch $g(j)$ definiert, so ist $0 < \rho'_i(r_j) < 1$ für alle $i \geq p$.

(d) Ist $E(j)$ definiert, so gilt $0 < \rho'_i(r_j) < 1$ für $p \leq i < E(j) - 1$, weiterhin $\rho'_{E(j)-1}(r_j) = 1$ und außerdem $1 < \rho'_i(r_j) < 2$ für $E(j) < i'$.

(e) Ist $g(j)$ definiert, so gilt $0 < \rho'_{i'}(r_j) < 1$ für $p \leq i < g(j) - 1$ und $1 < \rho'_i(r_j) < 2$ für $g(r_j) - 1 \leq i$.

In gleicher Weise ergeben sich dann unter Beachtung der in der Behauptung genannten Bedingungen auch die Werte für $\underline{\sigma'}$. Damit sind dann $\underline{\rho'}$ und $\underline{\sigma'}$ äquivalent, was $\underline{w} \in L_{\mathrm{ne}}$ beweist. □

Wir werden gleich das Iterationslemma für reguläre Sprachen auf L' anwenden. Dafür brauchen wir die folgende kombinatorische Aussage.

Behauptung 6 Es sei C das dreielementige Alphabet 3. Für jede natürliche Zahl r und jede natürliche Zahl t gibt es eine Zahl $M = M(r,t)$ mit folgender Eigenschaft:

Ist $x = x'1x''$ ein diskretes Wort über C in dem 1 genau einmal und 2 höchstens r-mal auftritt und $|x'| \geq M(r,t)$ sowie $|x''| \geq M(r,t)$ gelten, so läßt sich x schreiben als $x^{(1)}0^m x^{(2)}0^m x^{(3)}$ mit $m > t + |x^{(2)}|$ und so daß $x^{(2)}$ den Buchstaben 1 enthält.

Beweis. Wir zeigen die Behauptung per Induktion über r für $M(r,t) = 4^r(t+2)$.

Ist $r = 0$, so reicht es, M gleich $t+2$ zu wählen. Also stimmt die Behauptung für $r = 0$.

Nehmen wir also $r > 0$ an und schreiben x' in der Form $0^{m'}z'$ und x'' in der Form $z''0^{m''}$ mit jeweils maximalem m' bzw. m''. Ohne Einschränkung nehmen wir $m' \leq m''$ an.

Ist $n - m' - 1 \geq M(r-1,t)$, so folgt die Behauptung aus der Induktionsvoraussetzung. Andernfalls ist $n - m' - 1 < M(r-1,t)$, also $m' > n - M(r-1,t) - 1$, und $|z'1z''| \leq 2M(r-1,t) + 1$. Wegen $M(r,t) = 4M(r-1,t-1)$ folgt dann aber $m' > 4M(r-1,t) - M(r-1,t) - 1$. Daraus ergibt sich mit $M(s,t) \geq t+2$ für alle s die Abschätzung $m' > 2M(r-1) + 1 + t$, woraus $m' > |z'1z''| + t$ folgt. Aus der Annahme $m' \leq m''$ folgt dann auch $m'' > |z'1z''| + t$, was zeigt, daß eine zulässige Zerlegung gegeben ist, wenn $x^{(1)} = \epsilon$ und $x^{(3)} = 0^{m''-m'}$ sowie $x^{(2)} = z'1z''$ gewählt werden. □

Nehmen wir an, ein L' erkennender endlicher Automat habe k Zustände. Dann wählen wir eine natürliche Zahl l mit $l > M(s,k^4)$ und eine sehr kleine positive reelle Zahl ε (kleiner als $\frac{1}{2l}$).

Wir betrachten das Wort $\underline{w}$ der Länge $4l-1$ mit

$$w_i = \begin{cases} \frac{i}{2l}\,, & \text{falls } 0 \le i < 2l\,, \\ \frac{i}{2l} + \varepsilon\,, & \text{falls } i = 2l\,, \\ \frac{i}{2l}\,, & \text{falls } 2l < i < 3l\,, \\ \frac{i}{2l} - \varepsilon\,, & \text{falls } i = 3l\,, \\ \frac{i}{2l} + \varepsilon\,, & \text{falls } i = 3l+1\,. \\ \frac{i}{2l}\,, & \text{falls } 3l+1 < i < 4l\,. \end{cases}$$

Damit gehört $\underline{w}$ zu L_{ne}. Das nach 4.5.4.2) geforderte Teilwort der Dauer Eins ist eindeutig und befindet sich zwischen Position l und Position $3l$.

Wir betrachten ein zu $\underline{w}$ passendes Wort $\underline{b}$ aus L' und übernehmen sinngemäß die Definitionen von N, E und g.

Behauptung 7 Für $\underline{b}$ gilt:

1) $l \in \{N(j) \mid j < s\}$.
2) Für jedes j, für das $N(j)$ definiert ist und $l \neq N(j)$ sowie $N(j) < 2l$ gelten, ist $g(j)$ definiert, und es gilt $g(j) = N(j) + 2l$.
3) Für jedes j, für das $N(j)$ definiert ist und $N(j) = l$ gilt, ist $E(j)$ definiert, und es gilt $E(j) = 3l$.

Beweis. Die erste Eigenschaft folgt aus der Eindeutigkeit des Teilwortes der Dauer Eins in $\underline{w}$. Würde l nicht zu der genannten Menge gehören, so würde auch jedes Wort von $\mathfrak{A}$ akzeptiert werden, das aus $\underline{w}$ durch minimales Verschieben des Momentes l nach links oder rechts entsteht. Diese Wörter gehören aber definitionsgemäß nicht zu L_{ne}. Die beiden anderen Eigenschaften ergeben sich direkt aus der Konstruktion von $\underline{w}$. □

Sei nun $\underline{b} = \underline{x}b\underline{y}$ eine Zerlegung von $\underline{b}$ mit $E_0 \in b$.

Das Wort $\underline{x}$ hat an Position l einen von $\emptyset$ verschiedenen Buchstaben, etwa b'. Außerdem gibt es höchstens $s-1$ weitere Positionen, die mit einem von $\emptyset$ verschiedenen Buchstaben besetzt sind. Schließlich ist $l > M(s, k^4)$. Deshalb läßt sich $\underline{x}$ wie folgt aufteilen, wenn man in Behauptung 6 für 0 die Menge $\emptyset$ und für 1 den Buchstaben b' wählt:

$$\underline{x} = x^{(1)}\emptyset^m x^{(2)}\emptyset^m x^{(3)}\,,$$

wobei m größer als $|x^{(2)}|+k^4$ ist und $x^{(2)}$ die Position l von x enthält. Dementsprechend gibt es nach Behauptung 7 eine korrespondierende Zerlegung

$$\underline{y} = y^{(1)}\emptyset^m y^{(2)}\emptyset^m y^{(3)}$$

von $\underline{y}$. Damit kann $\underline{b}$ in der Form

$$\underline{b} \;=\; x^{(1)}\emptyset^m x^{(2)}\emptyset^m x^{(3)} b y^{(1)}\emptyset^m y^{(2)}\emptyset^m y^{(3)}$$

geschrieben werden.

Da L' von einem endlichen Automaten mit k Zuständen erkannt wird, kann u in den Teilwörtern $\emptyset^m$ gepumpt werden, allerdings möglicherweise mit unterschiedlichen Perioden p_0, p_1, p_2, p_3. Alle diese Perioden können $\leq k$ gewählt werden, so daß sich eine gemeinsame Periode $q \leq p_0p_1p_2p_3 \leq k^4$ ergibt.

Aufgrund der Voraussetzung über die Größe von m läßt sich nun ein Vielfaches m' von q finden, für das $m' \leq m$ und $m' \geq |x^{(2)}|$ gelten. Das Wort

$$\underline{b'} \;=\; x^{(1)}\emptyset^{m-m'} x^{(2)}\emptyset^{m+m'} x^{(3)} b y^{(1)}\emptyset^{m-m'} y^{(2)}\emptyset^{m+m'} y^{(3)}$$

gehört damit zu L'. Außerdem gilt wieder aufgrund der Länge von m':

(*) Es gibt keine Position, die in $\emptyset^m x^{(2)}\emptyset^m$ und $\emptyset^{m-m'} x^{(2)}\emptyset^{m+m'}$ mit zwei von $\emptyset$ verschiedenen Buchstaben aus B besetzt ist. Gleiches gilt auch für die entsprechenden Wörter mit Mittelstück $y^{(2)}$.

Nun ändern wir $\underline{w}$ zu $\underline{w'}$ ab, indem $w_{3l-m'-1}$ durch $\frac{1}{2l} - \varepsilon$ und $w_{3l-m'}$ durch $\frac{1}{2l} + \varepsilon$ ersetzt werden. Dann passen nach Behauptung 5 sowohl $\underline{b}$ wie auch $\underline{b'}$ zu $\underline{w'}$, wenn man (*) berücksichtigt. Damit kann $\mathfrak{A}$ nicht eindeutig sein und der Beweis von 4.5.3 ist beendet. ■

Es sei hier angemerkt, daß sich 4.5.3 bei entsprechender Anpassung der Sprache L_{ne} auch unmittelbar auf unendliche und unbeschränkte Intervallwörter übertragen läßt.

4.5.6 Korollar *Es gilt*

$$\mathbf{UTL} \subsetneq \mathbf{NTL}.$$

4.6 Die Feinstruktur von NTL

Zum Abschluß der Untersuchungen von Teilklassen von **NTL** wollen wir die Inklusionsbeziehungen, die zwischen ihnen bestehen, zu einem Gesamtbild, das die Feinstruktur von **NTL** erkennen läßt, fügen.

Aus der Ungleichungskette (4.7) erhält man mit 4.5.2 die echte Inklusion $\mathbf{DTL} \subsetneq \mathbf{UTL}$. Aus 4.5.6 ergibt sich $\mathbf{UTL} \subsetneq \mathbf{NTL}$. Wir können also festhalten:

$$\mathbf{DTL} \subsetneq \mathbf{UTL} \subsetneq \mathbf{NTL}. \tag{4.9}$$

Nimmt man zu (4.9) die Inklusionskette (4.7) hinzu und zieht außerdem den Hierarchiesatz (siehe 4.3.5) in Betracht, so ergibt sich ein Bild, wie es in Abbildung 4.4

zu finden ist. Dort stehen durchgezogene Linien für strikte und gestrichelte Linien für schwache Inklusionen, d. h., für Inklusionen, deren Striktheit noch offen ist. Mit $\mathbf{DTL}^k$ wird die Klasse der Sprachen bezeichnet, die durch einen deterministischen Uhrenautomaten mit höchstens k Uhren erkannt werden. Die strikte Inklusion

$$\mathbf{DTL}^0 \subsetneqq \mathbf{DTL}^1 \subsetneqq \mathbf{DTL}^2 \subsetneqq \ldots$$

folgt ebenfalls aus dem Satz über die Uhrenhierarchie (siehe 4.3.5): Da der in Abbildung 4.2 für die Beispielsprache L_m angegebene Uhrenautomat $\mathfrak{A}^m$ deterministisch ist, gilt sogar $L_m \in \mathbf{DTL}^m \setminus \mathbf{NTL}^{m-1}$ (für jedes $m > 0$).

Diagramm 4.4 ist insofern unvollständig, als nichtverbundene Klassen nicht unbedingt unvergleichbar sein müssen.

Es sei noch einmal darauf hingewiesen, daß alle Inklusionen unabhängig davon, ob man endliche, unendliche oder unbeschränkte Intervallwörter in Betracht zieht, und alle strikten Inklusionen schon für ein unäres Alphabet gelten.

Zum Abschluß sei bemerkt, daß die Situation bei regulären Sprachen diskreter Wörter viel einfacher ist. Für endliche diskrete Wörter kollabiert die Hierarchie wegen der Äquivalenz von Determinismus und Nichtdeterminismus sofort. Bei unendlichen diskreten Wörtern ist die Situation etwas komplizierter: Betrachtet man Automaten mit der Büchischen Akzeptierbedingung, so fallen Determinismus und Nichtdeterminismus auseinander (siehe z. B. [Tho90a]), während in [Arn83] gezeigt wird, daß Eindeutigkeit und Nichtdeterminismus gleichwertig sind. Legt man die Mullersche Akzeptierbedingung zugrunde, so ergibt sich wieder die Äquivalenz von Determinismus und Nichtdeterminismus [McN66] und damit ein Zusammenfallen des Graphen auf einen Knoten.

4.7 Erkennbarkeit unter zusätzlichen Bedingungen an die Ereignisdauer

Führt man sich einmal die Beweise der Unentscheidbarkeit des Universalitätsproblems, der Echtheit der Uhrenhierarchie, der Existenz einer inherent mehrdeutigen Sprache und des Nichtabschlusses von **NTL** unter Komplementbildung vor Augen, so stellt man fest, daß alle Beweise die „Dichte“ der Intervallwörter ausnutzen: Es gibt keine obere Schranke für die Anzahl der beobachtbaren Zeitpunkte, die in einem Intervall fester Länge auftreten können.

In diesem Abschnitt wollen wir deshalb der Frage nachgehen, wie sich die Situation verändert, wenn man sich bei der Betrachtung von Uhrenautomaten auf eine Menge von Intervallwörtern beschränkt, die die beschriebene Dichteeigenschaft nicht besitzt. Es werden drei unterschiedlich starke Einschränkungen untersucht, die in aufsteigender Stärke im folgenden aufgezählt sind:

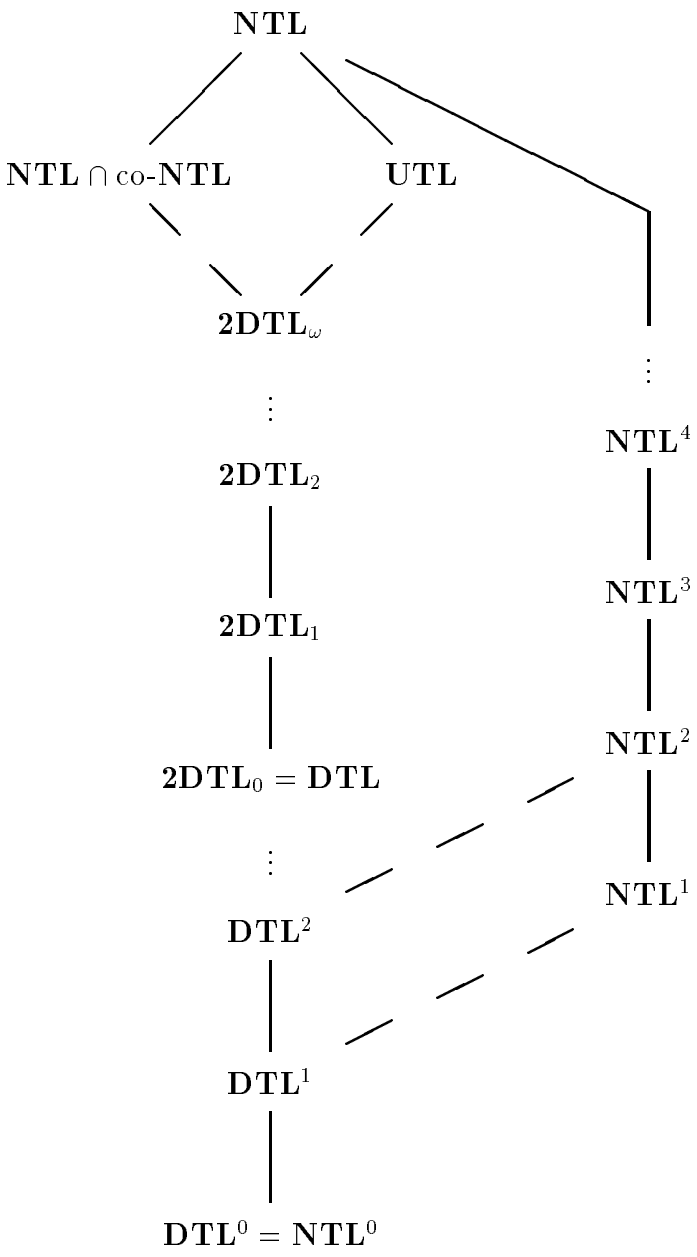


Abbildung 4.4: Feinstruktur der Klasse **NTL**

Durchgezogene Linien stehen für echte Inklusionen. Bei gestrichelten Linien ist die Echtheit der jeweiligen Inklusion noch offen. Es steht $\mathbf{DTL}^k$ für die Klasse der Sprachen, die durch einen deterministischen Uhrenautomaten mit k Uhren erkannt werden.

Beschränkte Variation: Intervallwörter, bei denen in jedem Intervall der Länge kleiner als Eins höchstens $k-1$ beobachtbare Zeitpunkte liegen,
Mindestdauer: Intervallwörter, deren Ereignisse die Mindestdauer Eins haben,
Einheitsdauer: Intervallwörter, deren Ereignisse sämtlich die Dauer Eins haben.

Das positive Hauptergebnis, das wir in dem nächsten Unterabschnitt erzielen werden, besagt, daß die Einschränkung ‚beschränkte Variation' zum Abschluß unter Komplementbildung führt. Daran schließt sich unmittelbar die Frage an, wie es sich mit stärkeren Automatenmodellen, z. B. Uhrenautomaten mit Haltemöglichkeit oder hybriden Automaten, verhält, wenn man auch sie bezüglich beschränkter Variation betrachtet. Wir werden zeigen, daß schon für die kleinste Erweiterung des Automatenmodells selbst bei der stärksten Einschränkung (auf Einheitsdauer) das Leerheitsproblem unentscheidbar ist oder aber „uninteressante" Sprachklassen beschrieben werden.

Um die Ergebnisse exakt formulieren zu können, führen wir neue Begriffe ein. Wir setzen allgemein voraus, daß Σ eine $\mathbf{P}$-Berechnungsstruktur sei.

4.7.1 Definition (relative Erkennbarkeit) Sei M eine Menge von Intervallwörtern. Ist $\mathfrak{A}$ ein Σ-Automat, so heißt $M \cap \mathrm{L}(\mathfrak{A})$ die von $\mathfrak{A}$ *relativ* zu M erkannte Sprache. Eine Sprache L von Intervallwörtern heißt Σ-erkennbar relativ zu M, wenn es einen Σ-Automaten gibt, der L relativ zu M erkennt.

Wir sprechen in diesem Zusammenhang auch von der Menge der relativ zu M Σ-erkennbaren Sprachen. Sagen wir, daß diese Menge unter Komplementbildung abgeschlossen ist oder eine boolesche Algebra bildet, so meinen wir jeweils Komplementbildung bezüglich M. Wir sprechen von einem *effektiven booleschen System*, wenn die booleschen Operationen auf den zugehörigen Automaten effektiv durchführbar sind, das heißt z. B., daß sich zu jedem Automaten $\mathfrak{A}$ effektiv ein Automat $\mathfrak{A}'$ konstruieren lassen muß, so daß $M \cap \mathrm{L}(\mathfrak{A}') = M \cap {\sim}\mathrm{L}(\mathfrak{A})$ gilt. Wir sprechen von einem *streng effektiven booleschen System*, wenn zusätzlich der relative Leerheitstest, d. h., die Frage „$M \cap \mathrm{L}(\mathfrak{A}) \stackrel{?}{=} \emptyset$", effektiv beantwortet werden kann.

In Abschnitt 3.2 haben wir, ohne es weiter auszuführen, schon relative Erkennbarkeit studiert, als wir Uhrenautomaten über unbeschränkten Intervallwörtern betrachtet haben.

Für jede Menge $P \subseteq \mathbf{Z}$ von Parametern sei die $\mathbf{P}$-Berechnungsstruktur Σ_P festgelegt durch:

$$\Sigma_P \;=\; (\mathbf{R}, \{0\}, \{\mathrm{add}_k \mid k \in P\}, \mathcal{J})\,.$$

Wir benutzen die Schreibweise $-\mathbf{N}$ für die Menge $\{-n \mid n \in \mathbf{N}\}$.

Offensichtlich ist eine Sprache von Intervallwörtern genau dann uhrenerkennbar, wenn sie $\Sigma_{\{1\}}$-erkennbar ist. Entsprechendes gilt für Uhrenautomaten mit Haltemöglichkeit und $\Sigma_{\{0,1\}}$ sowie hybride Automaten und $\Sigma_{\mathbf{Z}}$.

4.7.1 Beschränkte Variation

Wir beginnen mit der Definition der ‚Variation' eines Intervallwortes.

4.7.2 Definition (Variation) Für jedes Intervallwort $u = (\underline{a}, \underline{w})$ ist die mit $\mathbf{var}(u)$ bezeichnete *Variation* von u diejenige Zahl aus $\mathbf{N}_\infty$, die durch

$$\mathbf{var}(u) \;=\; \max\{k+1 \mid \exists i(i+k-1 < |u| \wedge W_{i+k} - W_i < 1)\}$$

gegeben ist. Ein Intervallwort u ist von *beschränkter Variation* k, wenn $\mathbf{var}(u) \leq k$ gilt.

Ein Intervallwort $u = (\underline{a}, \underline{w})$ ist also genau dann von beschränkter Variation k, wenn für alle i mit $i + k - 1 < |u|$ die Ungleichung $W_{i+k} - W_i \geq 1$ gilt. Insbesondere gilt $W_{i+1} - W_i \geq 1$, falls die Variation Eins ist. Außerdem ist die Variation jedes endlichen Intervallwortes endlich und die Variation eines beschränkten unendlichen Intervallwortes unendlich. (Beachte, daß die Variation eines unbeschränkten Intervallwortes sowohl endlich als auch unendlich sein kann.)

4.7.3 Beispiel Es sei V^k die Menge, die durch

$$V^k \;=\; \{u \mid \mathbf{var}(u) \leq k\}$$

gegeben ist, d. h., V^k ist die Menge aller Intervallwörter von beschränkter Variation k. Anders ausgedrückt: Ein Intervallwort u gehört zu V^k, wenn für jede Zerlegung $u = xyz$ aus $||y|| < 1$ die Abschätzung $|y| < k$ folgt. In Abbildung 4.5 ist ein Uhrenautomat mit k Uhren zu sehen, der V^k deterministisch erkennt. Er ist an den Automaten in Abbildung 4.2 angelehnt.

Da der Automat $\mathfrak{B}^k$ ein deterministischer Büchi-Uhrenautomat ist, können wir aufgrund von 1.4.4 festhalten:

4.7.4 Bemerkung Die Menge V^k ist deterministisch uhrenerkennbar und hat Uhrenindex $\leq k$.

Da die Klasse der uhrenerkennbaren Sprachen unter Durchschnittbildung abgeschlossen ist, gilt:

4.7.5 Bemerkung Jede Sprache, die relativ zu V^k uhrenerkennbar ist, ist auch selbst uhrenerkennbar.

Es gilt sogar noch mehr:

4.7.6 Lemma *Zu jedem Uhrenautomaten $\mathfrak{A}$ läßt sich effektiv ein vollständiger, deterministischer Uhrenautomat $\mathfrak{A}'$ konstruieren, der die von $\mathfrak{A}$ relativ zu V^k erkannte Sprache erkennt.*

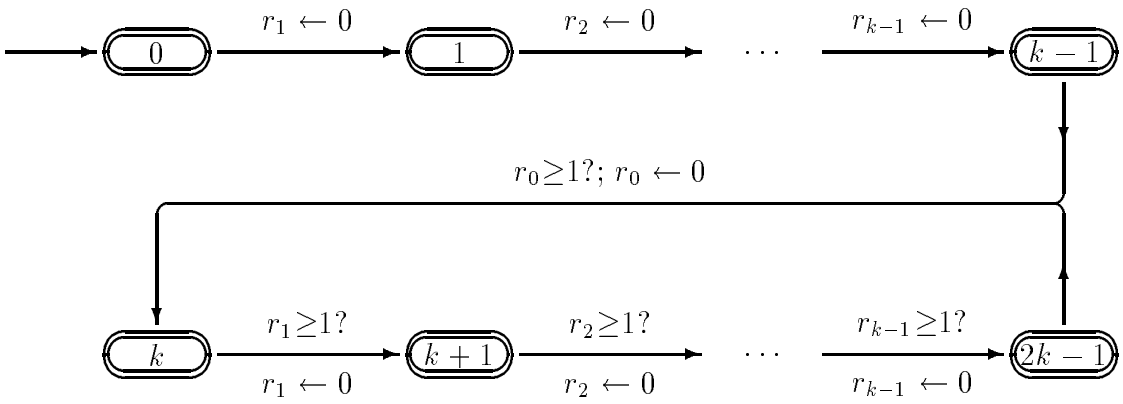


Abbildung 4.5: Der Uhrenautomat $\mathfrak{B}^k$, der die Wörter von beschränkter Variation k erkennt

Der angegebene Automat erkennt die Sprache V^k. Es ist an jedem Pfeil A: $\dot{r}_0 = \ldots = \dot{r}_{k-1} = 1$ zu ergänzen.

Beweis. Wir gehen von einem Uhrenautomaten $\mathfrak{A}$ mit $\mathfrak{A} = (Q, R, s, \Delta, F)$ und $R = \{r_0, \ldots, r_{s-1}\}$ aus.

Wegen 1.4.10, 4.7.4 und der Tatsache, daß die Klasse der deterministisch uhrenerkennbaren Sprachen effektiv gegen Bildung von Durchschnitten abgeschlossen ist, reicht es, einen registerdeterministischen Uhrenautomaten $\mathfrak{A}''$ zu konstruieren, für den $\mathrm{L}(\mathfrak{A}'') \cap V^k = \mathrm{L}(\mathfrak{A}) \cap V^k$ gilt.

Wir nehmen an, daß m der maximale Vergleichswert von $\mathfrak{A}$ ist und betrachten das Verhalten von $\mathfrak{A}$ beim Lesen eines Wortes aus V^k. Jede Uhr, sofern sie nicht zurückgesetzt wird, zeigt nach $km + 1$ Leseschritten einen Wert an, der größer als m ist. Dann ist ihr Wert bis zur nächsten Rücksetzung nicht weiter von Belang: Bei der Auswertung von Registertests kann für die Uhr ein Wert von z. B. $m+1$ angenommen werden, da alle Werte, die größer als m sind, auch m-äquivalent sind (vgl. 4.1.1).

Diese Überlegung führt zu dem folgenden Ansatz für die Konstruktion von $\mathfrak{A}''$: Der Automat $\mathfrak{A}''$ simuliert den alten Automaten $\mathfrak{A}$, wobei er jedoch $km + 1$ Uhren $r'_0, \ldots, r'_{mk}$ benutzt, deren Menge mit R' bezeichnet werde. Die Uhr r'_i wird an den Positionen j mit $j \bmod (km+1) = i$ zurückgesetzt, also jeweils nach $km+1$ Schritten. Würde in $\mathfrak{A}$ eine Uhr r_i an einer Position l zurückgesetzt und ist j der Rest von l modulo $mk + 1$, so merkt sich $\mathfrak{A}''$ die Uhr r'_j als Referenz für r_i (in der endlichen Kontrolle) und wertet die nachfolgenden Registertests so aus, als würde r'_j den Wert von r_i haben. Würde r_i in $\mathfrak{A}$ nicht nach km Schritten zurückgesetzt, so merkt sich $\mathfrak{A}''$, daß r_i einen Wert größer als m anzeigen würde und ersetzt in jedem nachfolgenden Registertest den Wert von r_i durch $m + 1$.

Offensichtlich ist durch das angegebene Verfahren der Registerdeterminismus des Automaten gesichert.

Nun wenden wir uns der formalen Konstruktion von $\mathfrak{A}'' = (Q', R', s', \Delta', F')$ zu. Entsprechend der obigen Überlegungen setzen wir wie folgt:

$$\begin{aligned} Q' &= Q \times \wp(s) \times \wp(mk+1)^s \times mk+1\,, \\ F' &= F \times \wp(s) \times \wp(mk+1)^s \times mk+1\,, \\ s' &= (s, \emptyset, \{(i,0) \mid i \in s\}, 0)\,. \end{aligned}$$

In der ersten Komponente des neuen Zustandsraumes wird der Zustand des alten Automaten verfolgt, die zweite Komponente merkt sich, welche der alten Uhren den Wert m überschritten hätten, die dritte Komponente stellt die Zuordnung der alten Uhren zu den neuen dar und die vierte Komponente zählt modulo $mk+1$. Für jede Transition (q, a, ϕ, ζ, q') aus Δ, jede Teilmenge M von s, jede Funktion $f\colon s \to mk+1$ und jedes $i < mk+1$ werde die Transition

$$((q, M, f, i), a, \phi', \{([i+1], 0)\}, (q', M', f', [i+1]))$$

mit

$$\begin{aligned} M' &= M \cup \{j \mid f(j) = [i+1]\} \setminus \{j \mid j \in \mathrm{dom}(\zeta)\}\,, \\ f'(j) &= \begin{cases} [i+1]\,, & \text{falls } r_j \in \mathrm{dom}(\zeta), \\ f(j)\,, & \text{sonst,} \end{cases} \qquad \text{für } j < s \end{aligned}$$

in Δ' aufgenommen. Dabei stehe $[i+1]$ für den Rest von $i+1$ modulo $mk+1$ und es entstehe ϕ' aus ϕ durch Ersetzen eines Testes $r_j \sim d$ durch den Test $r'_{f(j)} \sim d$, falls j nicht zu M gehört, und durch den Wahrheitswert von $m+1 \sim d$, andernfalls. ■

Damit erhalten wir aus 1.4.6:

4.7.7 Satz *Die Menge der relativ zu V^k uhrenerkennbaren Sprachen bildet ein streng effektives boolesches System, das auch effektiv unter Projektion abgeschlossen ist.*

Beweis. Der effektive Abschluß unter Durchschnitt und Vereinigung sowie Projektion folgt aus 1.4.7. Aus 4.7.6 und der Entscheidbarkeit des Leerheitsproblem für Uhrenautomaten folgt die Entscheidbarkeit des relativen Leerheitsproblems. ■

Das Resultat 4.7.6 läßt sich noch verbessern.

4.7.8 Lemma *Ist P eine Menge mit $P \subseteq \mathbf{N} \setminus \{0\}$ oder $P \subseteq -\mathbf{N} \setminus \{0\}$, so läßt sich zu jedem Σ_P-Automaten $\mathfrak{A}$ effektiv ein vollständiger, deterministischer Σ_P-Automat $\mathfrak{A}'$ konstruieren, der die von $\mathfrak{A}$ relativ zu V^k erkannte Sprache erkennt.*

Beweis. Die Idee aus dem Beweis von 4.7.6 kann prinzipiell übernommen werden. Da nach $km+1$ Leseschritten ohne Rücksetzung jedes Register einen Wert hat, der größer als m ist, kann man sich darauf beschränken, Register nur $km+1$ Schritte zu verfolgen. Weil aber in den Automaten mehrere Schrittfunktionen zur Anwendung kommen können, reicht es nicht aus, an jeder Position nur ein Register zurückzusetzen und neu zu starten. Stattdessen setzt man soviele Register zurück, wie nötig sind, um alle Registerbelegungen verfolgen zu können, die in $km+1$ Leseschritten ohne Rücksetzung entstehen können. Benutzt der Automat l verschiedene Schrittfunktionen, so sind dies l^{km+1} verschiedene Möglichkeiten, jeweils gegeben durch eine Folge von $km+1$ Schrittfunktionen. Insgesamt werden also $(km+1)l^{km+1}$ verschiedene Register nötig. ■

Als Folgerung erhalten wir:

4.7.9 Korollar *Für jede Menge P mit $P \subseteq \mathbf{N} \setminus \{0\}$ oder $P \subseteq -\mathbf{N} \setminus \{0\}$ bildet die Menge der relativ zu V^k Σ_P-erkennbaren Sprachen ein effektives boolesches System, das auch effektiv unter Projektionsbildung abgeschlossen ist.*

Es stellt sich die Frage, ob das Leerheitsproblem in diesen Fällen entscheidbar ist, d. h., ob die Menge der relativ zu V^k Σ_P-erkennbaren Sprachen von Intervallwörtern ein streng effektives boolesches System bilden. Dies ist nicht der Fall, wie wir im nächsten Unterabschnitt sehen werden.

4.7.2 Mindestdauer

Wir setzen $k = 1$, d. h., wir beschäftigen uns mit der Menge V^1 der Menge aller Intervallwörter mit Variation Eins. Offensichtlich gilt

$$V^1 = \{(\underline{a}, \underline{w}) \mid \forall i(i < |(\underline{a}, \underline{w})| \to w_i \geq 1\}\,.$$

Dadurch unterscheidet sich V^1 von V^2, V^3, ... in einem wesentlichen Punkt: Jedes in einem Intervallwort aus V^1 vorkommende Ereignisse hat eine Mindestdauer.

4.7.10 Satz *Für jede mindestens zweielementige Menge P mit $P \subseteq \mathbf{N}$ oder $P \subseteq -\mathbf{N}$ ist das Leerheitsproblem für Σ_P-Automaten relativ zu V^1 unentscheidbar. Es ist Σ_1^1-vollständig für unendliche Wörter.*

Die in [ACHH93] und [Čer92] vorgestellten Beweise für den allgemeinen Fall beliebiger Intervallwörter können hier nicht unmittelbar übernommen werden, da sie Intervallwörter größerer Variation benutzen.

Beweis. Wir beschränken uns darauf, die Behauptung für den Fall $P = \{1, 2\}$ nachzuweisen. Die anderen Fälle können in ähnlicher Weise behandelt werden.

Der Beweis erfolgt durch Reduktion des Problems, ob eine Zweiregistermaschine eine haltende bzw. eine rekurrente Berechnung hat, auf das gegebene. Dazu konstruieren wir aus jedem Programm M für eine Zweiregistermaschine mit Registern x und y

einen Σ_P-Automaten $\mathfrak{A}$, der genau dann eine akzeptierende Berechnung auf einem Wort aus V^1 hat, wenn M eine rekurrente Berechnung besitzt.

Wir nehmen dabei an, daß das Programm M die Gestalt (4.8) hat und o. B. d. A. keine Sprunganweisung auf ihre eigene Zeile zeigt.

Der Wert n eines Registers der Zweiregistermaschine soll in $\mathfrak{A}$ dadurch repräsentiert werden, daß zwei Registerinhalte die Differenz $1 + 2^{-n}$ haben. Genauer gesagt, der Zustandsraum des Automaten soll $\{0, \ldots, r\}$ enthalten, und es soll für jeden Zustand $i < r+1$ Registerpaare (x_i, x_i') und (y_i, y_i') geben, so daß die folgenden Aussagen äquivalent sind:

1) Es gibt eine Berechnung von M, die Zeile i erreicht und in der dann x den Wert k hat und y den Wert l.

2) Es gibt eine Berechnung von $\mathfrak{A}$, die eine Konfiguration (i, ρ) erreicht, so daß gelten:

$$(*)\ \ \rho(x_i) - \rho(x_i') = 1 + 2^{-k} \text{ und } \rho(y_i) - \rho(y_i') = 1 + 2^{-l},$$
$$(**)\ \ \rho(r) \leq 20 \text{ für } r \in \{x_i, x_i', y_i, y_i'\},$$
$$(***)\ \ \rho(r) = 0 \text{ für } r \notin \{x_i, x_i', y_i, y_i'\}.$$

Die Konstruktion von $\mathfrak{A}$ erfolgt dadurch, daß für jede Additionszeile i `x:=x+1` von M eine Transitionsfolge zwischen die Zustände i und $i+1$ (mit Hilfszuständen) gesetzt wird, die die entsprechende Programmzeile simuliert. Für eine Sprung- bzw. Subtraktionszeile i `if x=0 then` j `else x:=x+1` wird eine „Transitionsgabel“ zwischen i und $i+1$ bzw. j gesetzt.

Wir wollen die Konstruktion für die Additionszeile i `x:=x+1` vorstellen. Die Konstruktion für `y` verläuft analog. Die im Falle der Subtraktionszeile notwendigen Modifikationen werden am Schluß erläutert.

Wir betrachten Abbildung 4.6, in der die gesuchte Transitionfolge, die zwischen den Zuständen i und $i+1$ verläuft und die Zeile i `x:=x+1` simulieren soll, dargestellt ist. Jede der Transitionen setze ein Register r zurück, wende auf es die Schrittfunktion add_1 an und trage zu dem Registertest das Konjunktionsglied $r \geq 1 \wedge r \leq 2$ bei. Dadurch ist garantiert, daß jedes Ereignis eine Dauer zwischen Eins und Zwei (beide Werte eingeschlossen) hat.

Die Formel $\phi(x, x')$ sei durch

$$\phi(x, x') \ \triangleq \ \bigvee_{i<40} (x = i \wedge x' = i)$$

gegeben. Steht dieser Test an einer Transition, die x und x' mit add_1 fortschreibt, so kann die Transition nur dann genommen werden, wenn x und x' vor der Anwendung der Transition beide gleichen Inhalt haben. Mit Hilfe dieses Tests können also

zwei Register auf ihre Gleichheit überprüft werden, sofern ihre Werte nicht schon 40 überschritten haben. Dies ist eine der wesentlichen Ideen in der Konstruktion.

Auf der linken Seite der Transitionsfolge, die wir mit $\underline{\delta}$ bezeichnen wollen, sind die Transitionen durch Angabe der Fortschreibungen, Registertests und Rücksetzungen genau angegeben. Register, die nicht aufgeführt sind (abgesehen von r) werden nicht getestet und mit der letzten Transition zurückgesetzt, sofern es sich nicht um x_{i+1}, x'_{i+1}, y_{i+1} oder y'_{i+1} handelt. Jede fehlende Schrittfunktion ist entgegen der üblichen Konvention durch add_1 zu interpretieren. Auf der rechten Seite der Transitionsfolge ist verzeichnet, wie die Dauer der Ereignisse ist, wenn die Transitionsfolge ganz durchschritten wird.

Es soll nun die Korrektheit der Konstruktion begründet werden. Dazu genügt es, nachzuweisen, daß für jede Registerbelegung von $\mathfrak{A}$ mit (*), (**) und (***) die folgende Behauptung Richtigkeit hat:

1) Es gibt ein unäres Wort $\underline{w}$ und eine Registerbelegung ρ', so daß gelten:

   (#) $(i,\rho) \xrightarrow[\underline{w}]{\underline{\delta}} (i+1,\rho')$,

   (+) $\rho'(x_{i+1}) - \rho'(x'_{i+1}) = 1 + 2^{-k-1}$ und $\rho'(y_{i+1}) - \rho'(y'_{i+1}) = 1 + 2^{-l}$,

   (++) $\rho(r) \leq 20$ für $r \in \{x_i, x'_i, y_i, y'_i\}$,

   (+++) $\rho(r) = 0$ für $r \notin \{x_i, x'_i, y_i, y'_i\}$.

2) Ist ρ' eine Registerbelegung und $\underline{w}$ ein unäres Wort, so daß (#) gilt, dann sind auch (+), (++) und (+++) erfüllt.

Wir begründen zuerst die Gültigkeit von 2) und nehmen dazu (#), die Existenz des unären Wortes $\underline{w}$ und der Registerbelegung ρ' an.

Da $\phi(x_i, x'_i)$ an Zustand C gelten muß, folgt $\rho(x_i) + w_0 + w_1 + w_2 = \rho(x'_i) + 2w_0 + w_1 + w_2$, also

$$w_0 \;=\; \rho(x_i) - \rho(x'_i) = 1 + 2^{-k}\,. \tag{4.10}$$

Genauso folgt

$$w_1 \;=\; 1 + 2^{-l}\,, \tag{4.11}$$

woraus sich sofort $\rho'(y_{i+1}) - \rho'(y'_{i+1}) = 1 + 2^{-l}$, also der eine Teil von (+) ergibt. Da an Zustand F der Test $\phi(a, a')$ erfüllt sein muß, folgt $2w_3 + w_4 = w_3 + 2w_4$, also

$$w_3 = w_4\,. \tag{4.12}$$

Weiterhin muß offensichtlich

$$w_6 \;=\; 1 \tag{4.13}$$

gelten. Damit ergibt sich wegen des Tests $\phi(f, f')$ an Zustand $i+1$:

$$2w_0 + w_1 + w_2 + w_3 + w_4 + w_5 + 2w_6 \;=\; w_0 + w_1 + w_2 + 2w_3 + 2w_4 + w_5 + w_6\,.$$

Einsetzen der Gleichungen (4.12) und (4.13) und Subtraktion von w_1, w_2 und w_5 auf beiden Seiten liefert :

$$2w_0 + 2w_3 + 2 \;=\; w_0 + 4w_3 + 1\,.$$

Weiteres Vereinfachen ergibt dann:

$$w_0 + 1 \;=\; 2w_3\,,$$

woraus mit (4.10)

$$w_3 \;=\; 1 + 2^{-k-1}$$

und damit auch $\rho'(x_{i+1}) - \rho'(x'_{i+1}) = 1 + 2^{-k-1}$, also der andere Teil von $(+)$ folgt. Weil $w_i \leq 2$ jeweils gilt und jedes der Register x_{i+1}, x'_{i+1}, y_{i+1} und y'_{i+1} in $\underline{\delta}$ einmal zurückgesetzt wird, ist auch $(++)$ erfüllt. Schließlich gilt nach Konstruktion immer $(+++)$.

Um 1) zu zeigen, muß man sich nur noch klar machen, daß w_2, w_5 und w_7 geeignet gewählt werden können, wenn man die anderen Glieder aus $\underline{w}$ mit den Werten aus der obigen Herleitung belegt. Eine geeignete Wahl von w_2, w_5 und w_7 ist aber aufgrund der in ϕ auftretenden hohen Vergleichswerte möglich.

Im Falle einer Subtraktionszeile i `if x = 0 then` j `else x:=x-1` verzweigt der Automat ausgehend vom Zustand i in zwei „Transitionsäste“, nämlich in einen, der zu $i+1$ führt und ausgeführt wird, wenn `x` größer als Null ist, und in einen anderen, der zu j führt und ausgeführt wird, wenn `x` gleich Null ist.

Wegen $(^*)$ ist `x` genau dann Null, wenn $\rho(x'_i) - \rho(x_i) = 1$ gilt. Dies kann mit Hilfe einer geeigneten Transitionsfolge unter Verwendung von ϕ (nichtdeterministisch) getestet werden kann. Ob andernfalls $1 < \rho(x'_i) - \rho(x_i) \leq 2$ gilt, kann in ähnlicher Weise unter Verwendung einer zu ϕ dualen Formel überprüft werden.

Nach dem Test brauchen in dem Ast, der nach j führt, lediglich die Inhalte von `x` und `y` gerettet zu werden. Dafür kann die bei der Additionszeile für `y` benutzte Konstruktion genutzt werden. Der Ast, der nach $i+1$ verzweigt, kann nach dem Test prinzipiell wie die ganze Transitionsfolge für die Additionszeile aufgebaut werden. Der Subtraktion muß dadurch Rechnung getragen werden, daß die Rollen von x_i und x'_i vertauscht werden.

Im Falle der endlichen Wörter wird r der Endzustand des Automaten, im Falle der unendlichen Wörter wird der Zustand, zu dem die korrespondierende Programmzeile in M immer wieder aufgesucht werden soll, als Endzustand ausgezeichnet. ■

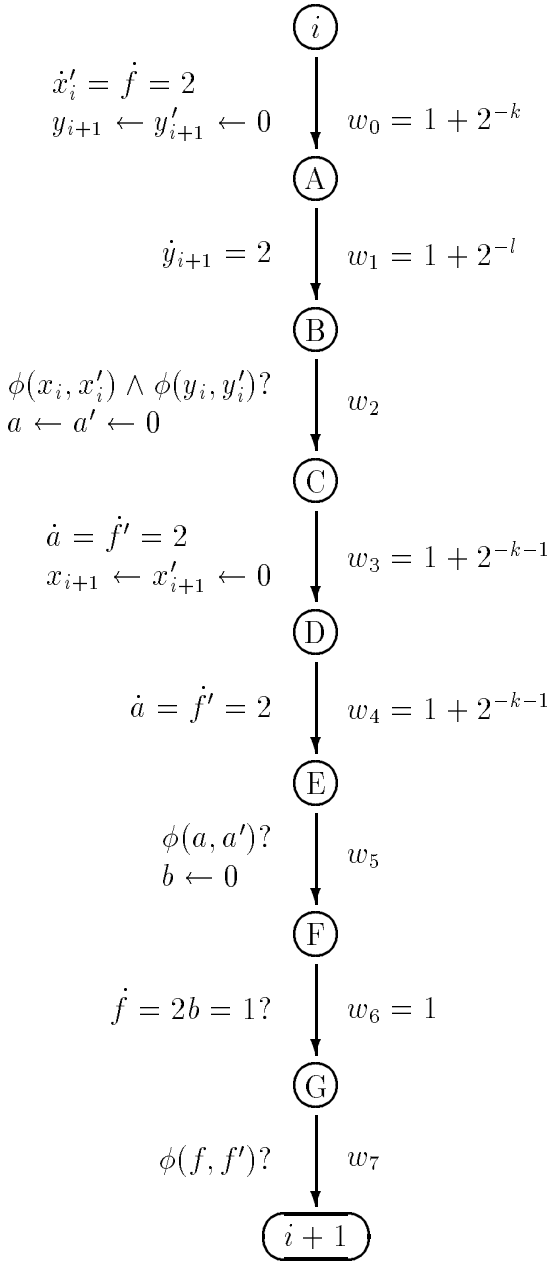


Abbildung 4.6: Simulation einer Addition einer Zweiregistermaschine

Jedes an einem Pfeil nicht aufgeführte Register wird mit add_1 fortgeschrieben. Außerdem wird jeder Pfeil zusätzlich mit $\dot{r} = 1$ und $r \geq 1 \wedge r \leq 2$ beschriftet. Die unterste Transition setzt alle Register bis auf x_{i+1}, x'_{i+1}, y_{i+1} und y'_{i+1} zurück.

4.7.3 Einheitsdauer

Die letzte Menge von Intervallwörtern, relativ zu der wir Erkennbarkeit durch Σ_P-Automaten studieren wollen, ist die Menge E aller Wörter, deren Ereignisse die Dauer Eins haben, d. h., wir setzen:

$$E \;=\; \{(\underline{a}, \underline{w}) \mid \forall i(i < |\underline{w}| \to w_i = 1)\}\,.$$

Diese Intervallwörter erinnern sehr an diskrete Wörter, was auch durch den folgenden Satz bestätigt wird:

4.7.11 Satz *Für eine Sprache L von Intervallwörtern mit $L \subseteq E$ und eine beliebige Parametermenge P mit $P \subseteq \mathbf{N}$ oder $P \subseteq -\mathbf{N}$, die mindestens ein von Null verschiedenes Element enthält, sind die drei folgenden Aussagen äquivalent:*

1) Die diskrete Projektion von L ist regulär.

2) L ist Σ_P-erkennbar.

3) L ist Σ_P-erkennbar relativ zu E.

Die jeweiligen Umformungen können effektiv erfolgen.

Beweis. Aus Symmetriegründen können wir uns auf den Fall $P \subseteq \mathbf{N}$ beschränken. Es sei n ein von Null verschiedenes Element aus P.

1) $\Rightarrow$ 2) Aus jedem beliebigen endlichen Automaten $\mathfrak{B}$ für die diskrete Projektion von L kann auf triviale Weise ein Σ_P-Automat $\mathfrak{A}$ gewonnen werden, der L erkennt: Man fügt ein Register r zu $\mathfrak{A}$ hinzu, wendet in jeder Transition auf dieses die Schrittfunktion add_n an, überprüft den Test $r = n$ und setzt das Register wieder zurück.

2) $\Rightarrow$ 3) Diese Behauptung ist wegen $L \subseteq E$ trivialerweise richtig.

3) $\Rightarrow$ 1) Sei dazu $\mathfrak{A}$ ein Σ_P-Automat mit Registermenge R und maximalem Vergleichswert m. Da $\mathfrak{A}$ nur über Wörtern aus E interpretiert wird, können als Registerwerte nur natürliche Zahlen in Frage kommen. Da außerdem keine Schrittfunktionen mit negativem Parameter erlaubt sind, brauchen die Registerwerte nur bis zum maximalen Vergleichswert verfolgt zu werden. Damit reicht es also, zwischen den Registerwerten $0, 1, \ldots, m+1$ zu unterscheiden, was leicht in der endlichen Kontrolle durchgeführt werden kann.

Dieser Idee folgend gewinnt man nun einen Automaten für die diskrete Projektion von L, wenn man wie folgt verfährt: Man versieht die Zustandsmenge von $\mathfrak{A}$ mit einer zusätzlichen Komponente, die alle Registerbelegungen $\rho\colon R \to m+2$ enthält. Für jede Transition $(q, a, \theta, \phi, \zeta, q')$ aus $\mathfrak{A}$ und jede Registerbelegung ρ nimmt man dann die Transition $((q, \rho), a, (q', \rho')$ in die Transitionsrelation des neuen Automaten auf, wenn $\rho \overset{1}{\theta}$ den Test ϕ erfüllt und sich ρ' aus $\rho \overset{1}{\theta} \zeta$ ergibt, indem jeder Wert $m+2$ durch $m+1$ ersetzt wird. Der Rest der Konstruktion ist kanonisch. ∎

Demgegenüber steht ein Lemma, das schon in [KPSY93] bewiesen wurde: Ist P eine Berechnungsstruktur, die mindestens eine positive und eine negative Zahl enthält, dann ist das Leerheitsproblem für Σ_P-Automaten relativ zu E unentscheidbar. Denn man kann in vergleichbar einfacher Weise das Halteproblem für Zweiregistermaschinen auf das vorgegebene Problem reduzieren.

Die wesentlichen Ergebnisse dieses Abschnittes sind der Übersichtlichkeit halber noch einmal in Abbildung 4.7 dargestellt.

Die Uhrenautomaten sind in der Tabelle in der Zeile zu finden, die mit $|P| = 1$, $0 \notin P$ beginnt, denn sie benutzen nur eine Schrittfunktion, nämlich add_1, und der zugehörige Parameter ist von Null verschieden.

Es ist auf den ersten Blick ersichtlich, weshalb die Uhrenautomaten eine herausragende Rolle spielen. Durch den einzig schwächeren Automatentyp (eine Zeile darüber) werden nur uninteressante Sprachen beschrieben, da in ihnen keine zeitlichen Bedingungen ausgedrückt werden können. Und schon der nächststärkere Automatentyp (eine Zeile darunter) führt zu einem unentscheidbaren Leerheitsproblem in allen Fällen bis auf die Einschränkung auf Einheitsdauer, und in diesem Fall werden wieder nur uninteressante Sprachen beschrieben. Für alle anderen, stärkeren Automatentypen ist das Leerheitsproblem unentscheidbar, unabhängig davon, relativ zu welcher Klasse von Intervallwörtern man Erkennbarkeit betrachtet.

Für die Uhrenautomaten ist außer dem allgemeinen Fall — der Interpretation über beliebigen Intervallwörtern — die Einschränkung auf beschränkte Variation von Interesse. Teile der Dichteeigenschaft gehen bei dieser Einschränkung zwar verloren, doch dafür erhält man ein streng effektives boolesches System, insbesondere den Abschluß unter Komplementbildung. (Vgl. auch die Betrachtungen im Resümee am Schluß der Arbeit.)

<table>
<tr><th colspan="2"></th><th>Einheitsdauer</th><th>beschränkte Variation
Mindestdauer</th><th>beliebige
Intervallwörter</th></tr>
<tr><td colspan="2">$P = \emptyset$
oder
$P = \{0\}$</td><td rowspan="3">reg.</td><td rowspan="2">reg.</td><td>reg.</td></tr>
<tr><td colspan="2">$|P| = 1, 0 \notin P$</td><td>str. eff. BS
Projektion</td><td>LP rek.
kein Kompl.
Projektion</td></tr>
<tr><td rowspan="2">$|P| \geq 2$</td><td>$P \subseteq \mathbf{N}$
oder
$P \subseteq -\mathbf{N}$</td><td rowspan="2">LP n. rek.</td><td>LP n. rek.</td></tr>
<tr><td>P mit pos.
u. neg. Element</td><td>reg.</td><td>LP n. rek.</td><td>LP n. rek.</td></tr>
</table>

Abbildung 4.7: Zusammenfassung der Ergebnisse zu relativer Erkennbarkeit

Die Parametermenge P gibt die Berechnungsstruktur Σ_P vor.

LP:	Leerheitsproblem,
str. eff. BS:	streng effektives boolesches System,
kein Kompl.:	kein Abschluß unter Komplementbildung,
Projektion:	effektiver Abschluß unter Projektionsbildung,
rek.:	entscheidbar („rekursiv“),
n. rek.:	nicht entscheidbar („nicht rekursiv“),
reg.:	jede Sprache von Intervallwörtern der betrachteten Einschränkung ist das (relative) Urbild einer regulären Sprache.

Kapitel 5

Ausdrucksstarke Logiken zur Beschreibung zeitabhängiger Systeme

Nach den allgemeinen Vorbereitungen im ersten Teil der Arbeit und den speziellen automatentheoretischen Untersuchungen des letzten Kapitels sind wir jetzt in der Lage, die zentrale Frage dieser Arbeit anzugehen: Wie sehen ausdrucksstarke Logiken zur Beschreibung zeitabhängiger Systeme aus, die eine entscheidbare Theorie besitzen oder zumindest ein entscheidbares Erfüllbarkeitsproblem? Und welche Eigenschaften besitzen derartige Logiken?

Aus der Entscheidbarkeit des Leerheitsproblems für Uhrenautomaten und dem speziellen Charakterisierungssatz für Uhrenautomaten gewinnen wir die Entscheidbarkeit des Erfüllbarkeitsproblems für die Logik $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$. Die These, daß es sich hierbei um eine ausdrucksstarke Logik mit entscheidbarem Erfüllbarkeitsproblem handelt, wollen wir durch „Abschätzungen“ von $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ nach oben und unten begründen. Zum einen zeigen wir mit Hilfe der Unentscheidbarkeitsergebnisse aus Abschnitt 4.7, daß die natürlichen Erweiterungen von $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ ein unentscheidbares Erfüllbarkeitsproblem besitzen (Abschnitt 5.2). Zum anderen werden wir zeigen, daß die bislang stärkste bekannte Logik zur Beschreibung zeitabhängiger Systeme, die eine entscheidbare Theorie besitzt, nämlich $\mathrm{MITL}^{\mathrm{p}}$, in $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ eingebettet werden kann. Darüberhinaus werden wir sogar zeigen, daß man bei einer Erweiterung von $\mathrm{MITL}^{\mathrm{p}}$ um Automatenoperatoren zu den gleichen Ergebnissen kommt (beides in Abschnitt 5.4) und daß die von Manna und Pnueli in [MP93] betrachtete Logik TL_{Γ} eine Teilklasse der von $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ definierten Mengen definiert und ebenfalls eine entscheidbare Theorie besitzt (Abschnitt 5.3).

Das erst jüngst von Alur, Courcoubetis und Henzinger erzielte Ergebnis über eine Erweiterung des Modells des Uhrenautomaten, das auch ein entscheidbares Leerheitsproblem besitzt, steht im Mittelpunkt des Abschnitts 5.5. Dort werden wir das

automatentheoretische Ergebnis aus logischer Sicht verwerten und dadurch eine Teilklasse von $\exists\mathrm{MSO}\Pi\int$ aussondern können, die ein entscheidbares Erfüllbarkeitsproblem besitzt, aber dennoch ausdrucksstärker als $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ ist.

Die spezielleren Ergebnisse des letzten Kapitels über Uhrenautomaten werden uns in Abschnitt 5.1 zu einem tieferen Verständnis von $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ verhelfen. Zum einen wird gezeigt, daß durch die Anzahl der Mengenvariablen in Formeln aus $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ eine unendliche Hierarchie gegeben ist. Zum anderen wollen wir uns mit dem Abschluß unter Negation beschäftigen und $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ mit der vollen Logik $\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ vergleichen (beides Unterabschnitt 5.1.2). In Abschnitt 5.1 wird außerdem ein logischer Beweis für die Entscheidbarkeit des Erfüllbarkeitsproblems von $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ unter Zuhilfenahme des Entscheidbarkeitskriteriums 2.8.2 gegeben (Unterabschnitt 5.1.1). Weiterhin soll die Asymmetrie der in $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ auftretenden Integralformeln, die darin liegt, daß die obere Grenze explizit durch eine Elementvariable und die untere jedoch implizit durch eine Mengenvariable vorgegeben ist, untersucht werden. Wir werden beweisen, daß auch die Hinzunahme der dualen Prädikate (mit impliziter Vorgabe der oberen Grenze und expliziter Angabe der unteren) die Ausdrucksstärke von $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ nicht vergrößert (Unterabschnitt 5.1.3). Schließlich wird der Beweis der Inklusion $\mathbf{2DTL}_\omega \subseteq \mathbf{UTL}$ (siehe 4.4.1) nachgeholt, den wir im letzten Kapitel schuldig geblieben sind (Unterabschnitt 5.1.4).

Im letzten Abschnitt dieses Kapitels wollen wir die Auswirkungen, die der Komplementabschluß der uhrenerkennbaren Sprachen relativ zu der Menge der Intervallwörter einer beschränkten Variation (siehe Unterabschnitt 4.7.1) auf $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ hat, genau studieren. Es wird sich zum einen herausstellen, daß $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ und $\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ die gleiche Ausdrucksstärke haben, wenn man ihre Formeln in Wörtern einer beschränkten Variation interpretiert. Zum anderen werden wir beweisen, daß die in Formeln von S1S unter Hinzunahme von direkten Abstandsmessungen formulierbare Theorie der Intervallwörter einer beschränkten Variation entscheidbar ist.

5.1 Die schwache Abstandslogik $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$

Wir beginnen mit einer Namengebung:

5.1.1 Definition (schwache Abstandslogik) Die Sprache $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ wird auch als *schwache Abstandslogik* bezeichnet.

Das Attribut „schwach“ wird hier dem Namen hinzugesetzt, um zu verdeutlichen, daß nicht die volle monadische Logik zweiter Stufe in der Signatur $\Pi\overleftarrow{\mathrm{d}}$ zur Verfügung steht und daß nur relative Abstandsmessungen möglich sind. In Abschnitt 5.2 werden wir die „volle Abstandslogik“ kennenlernen.

Aus der Entscheidbarkeit des Leerheitsproblems für Uhrenautomaten über endlichen, unendlichen und unbeschränkten Intervallwörtern (siehe [AD90]) können wir

aufgrund des speziellen Charakterisierungssatzes für Uhrenautomaten (siehe 3.7.2) und 2.8.1 schließen:

5.1.2 Satz *Das Erfüllbarkeitsproblem für* $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ *ist entscheidbar.*

In gewisser Hinsicht kann man die Benutzung des Ergebnisses von Alur und Dill umgehen, wenn man das Entscheidbarkeitskriterium 2.8.2 verwendet. Das soll im nächsten Unterabschnitt demonstriert werden.

5.1.1 Alternativbeweis zur Entscheidbarkeit des Erfüllbarkeitsproblems von $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$

Die wesentliche Idee des Beweises für die Entscheidbarkeit des Leerheitsproblems für Uhrenautomaten in [AD90] ist eine Einteilung der unendlichen Menge der Uhrenstände eines gegebenen Automaten in endlich viele Äquivalenzklassen.

In diesem Abschnitt wird diese Idee vorgestellt und erläutert, wie man sie benutzen kann, um die Entscheidbarkeit der Erfüllbarkeit von $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ „direkt“ zu beweisen. Als Korollar erhalten wir dann die Entscheidbarkeit des Leerheitsproblems für Uhrenautomaten und den Satz, daß die diskrete Projektion jeder uhrenerkennbaren Sprache regulär ist.

Nach dem Entscheidbarkeitskriterium 2.8.2 ist es hinreichend, wenn die Menge der Kodierungen aller Intervallwörter in uniformer Weise endlich axiomatisiert werden kann. Daß dies zutrifft, werden wir hier beweisen. Demnach ist unser Ziel der Nachweis, daß die Voraussetzungen von 2.8.2 im Falle der Berechnungsstruktur Σ_{UA} erfüllt sind.

Es seien s und k natürliche Zahlen und $s = \{0, \ldots, s-1\}$ eine Uhrenmenge.

Wir betrachten die Menge $\mathcal{R}$ der Uhrenstände $\rho\colon s \to \mathbf{P}_0$. Sie wurde schon in Abschnitt 1.2 mit zwei Operationen versehen, nämlich mit Rücksetzungen und Fortschreibungen. Für letztere wollen wir hier die in Abschnitt 4.1 eingeführte Schreibweise verwenden.

Es sei η der Uhrenstand mit $\eta(r) = 0$ für alle $r < s$.

Wir versehen $\mathcal{R}$ mit einer Verfeinerung Δ_k von $\equiv_k$. Es gelte $\rho\Delta_k\rho'$, falls die folgenden Bedingungen erfüllt sind:

1) Für jedes r mit $r < s$ und jede natürliche Zahl c mit $c \leq k$ gilt $\rho(r) > c$ genau dann, wenn $\rho'(r) > c$ gilt, und es gilt $\rho(r) = c$ genau dann, wenn $\rho'(r) = c$ gilt.

2) Für r und r' mit $r, r' < s$ ist $\mathrm{frac}(\rho(r)) \leq \mathrm{frac}(\rho(r'))$ genau dann, wenn $\mathrm{frac}(\rho'(r)) \leq \mathrm{frac}(\rho'(r'))$ gilt. Dabei stehe $\mathrm{frac}(c)$ für den Nachkommaanteil der reellen Zahl c.

Eine Äquivalenzklasse von Δ_k wird *Region* genannt.

Im Gegensatz zu $\equiv_k$ ist Δ_k außer mit Tests und Rücksetzungen auch mit Fortschreibungen verträglich:

5.1.3 Bemerkung Seien ρ und ρ' aus $\mathcal{R}$ mit $\rho \Delta_k \rho'$.

1) Ist $\sim$ eine Vergleichsrelation und d eine natürliche Zahl nicht größer als k, dann gilt für jedes Register r die Beziehung $\rho(r) \sim d$ genau dann, wenn $\rho'(r) \sim d$ gilt.

2) Für jede Rücksetzung ζ gilt $(\rho\zeta)\Delta_k(\rho'\zeta)$.

3) Für jedes w mit $w \in \mathbf{P}$ existiert ein w' mit $w' \in \mathbf{P}$ und $(\rho + w)\Delta_k(\rho' + w')$.

Punkt 1) führt zu der folgenden Festlegung: Für jede Region H und jedes Register r gelte $H(r) \sim d$ genau dann, wenn $\rho(r) \sim d$ für einen Uhrenstand $\rho \in H$ gilt.

Aus 2) folgt, daß Rücksetzoperationen auch sinnvoll auf die Menge $\mathcal{R}/\Delta_k$ aller Regionen übertragen werden können: Man setzt einfach $(\rho/\Delta_k)\zeta = (\rho\zeta)/\Delta_k$.

Punkt 3) führt zur Definition einer *Nachfolgerrelation* auf der Menge der Regionen:

$$\Theta_k \quad = \quad \{(\rho/\Delta_k, (\rho + w)/\Delta_k) \mid w \in \mathbf{P}\}\,. \tag{5.1}$$

Ist $\underline{\rho}$ eine Folge von Uhrenständen, so schreiben wir $\underline{\rho}/\Delta_k$ für die Folge

$$\rho_0/\Delta_k, \rho_1/\Delta_k, \rho_2/\Delta_k, \ldots$$

Zwei Folgen $\underline{H}$ und $\underline{H'}$ von Regionen *stimmen* mit einer Folge $\underline{\zeta}$ von Rücksetzungen *überein*, wenn gilt:

$$\begin{array}{rcll} H_0 & = & \eta/\Delta_k, & \\ H_i & \Theta_k & H_i' & \text{für } i < |\underline{H}|, \\ H_{i+1} & = & H_i'\zeta_i & \text{für } i < |\underline{H}|. \end{array}$$

Es soll $|\underline{H'} = \underline{\zeta}|$ gelten und $|\underline{H}| = 1 + |\underline{H'}|$. Die Folge $\underline{\zeta}$ ist dann allein durch $\underline{H'}$ bestimmt.

Sei $\underline{w}$ ein unäres Intervallwort und $\underline{\zeta}$ eine gleich lange Folge von Rücksetzungen. Dann *bestimmen* $\underline{w}$ und $\underline{\zeta}$ in natürlicher Weise zwei Folgen $\underline{\rho}$ und $\underline{\rho'}$ von Uhrenständen:

$$\begin{array}{rcll} \rho_0 & = & \eta & \\ \rho_i' & = & \rho_i + w_i & \text{für } i < |\underline{w}|, \\ \rho_{i+1} & = & \rho_i\zeta_i & \text{für } i < |\underline{w}|. \end{array}$$

Die beiden Folgen $\underline{\rho}/\Delta_k$ und $\underline{\rho'}/\Delta_k$ stimmen mit $\underline{\zeta}$ überein.

Das entscheidende Lemma in diesem Zusammenhang ist das folgende:

5.1.4 Lemma ([AD]) *Sind $\underline{H}$ und $\underline{H}'$ Folgen von Regionen, die mit einer Folge $\underline{\zeta}$ übereinstimmen, so gibt es ein unäres Wort $\underline{w}$, das zusammen mit $\underline{\zeta}$ Folgen $\underline{\rho}$ und $\underline{\rho'}$ bestimmt, für die $\underline{H} = \underline{\rho}/\Delta_k$ und $\underline{H}' = \rho'/\Delta_k$ gilt.*

Der Beweis wird induktiv über die Länge der Folgen geführt und nutzt 5.1.3.3) im Induktionsschritt aus.

Mit Hilfe dieses Lemmas wollen wir nun zeigen:

5.1.5 Lemma *Zu jeder natürlichen Zahl m und jeder endlichen Menge $\mathcal{T}$ von m-normalen erweitert-atomaren Σ_{UA}-Formeln läßt sich effektiv ein $\mathrm{MSO}\Pi(B_m(\mathcal{T}))$-Satz*

$$\mathsf{Interpretation}_m(\mathcal{T})$$

konstruieren, wobei $\mathcal{T}$ eine endliche Obermenge von $\mathcal{T}'$ mit m-normalen erweitert-atomaren Σ_{UA}-Formeln ist, so daß für jedes diskrete Wort v über $B_m(\mathcal{T})$ gilt:

$$v \models \mathsf{Interpretation}_m(\mathcal{T})$$

gdw. v ist die $(m,\mathcal{T})$-Kodierung eines Intervallwortes.

Beweis. Für jede endliche Menge $\mathcal{T}'$ von m-normalen erweitert-atomaren Σ_{UA}-Formeln gibt es eine natürliche Zahl k, so daß jede Formel aus $\mathcal{T}'$ von der Gestalt $\mathrm{d}(X_i, x) \sim d$ mit $d \leq k$ und $i < m$ ist. (Wir identifizieren dabei die erweiterten Symbole von Σ_{UA} mit den erweiterten Symbolen von $\Pi\overleftarrow{\mathrm{d}}$.) Für $\mathcal{T}$ wählen wir die Menge dieser Formeln. Dann ist $B_m(\mathcal{T})$ die Menge $A \times \wp(\mathcal{T}) \times \wp(m)$, die wir abkürzend mit B bezeichnen wollen.

Für die folgenden Untersuchungen wird die Menge $m = \{0, \ldots, m-1\}$ als Uhrenmenge zugrunde gelegt.

Behauptung Sei $(\underline{a}, \underline{T}, \underline{N})$ ein Wort über B und $\underline{\zeta}$ die Folge von Rücksetzungen, die durch $\zeta_i = \{(s,0) \mid s \in N_i\}$ definiert ist. Dann sind die beiden folgenden Aussagen äquivalent:

1) Das Wort $(\underline{a}, \underline{T}, \underline{N})$ ist eine $(m,\mathcal{T})$-Kodierung.
2) Es gibt Folgen $\underline{H}$ und $\underline{H}'$ von Regionen, die mit $\underline{\zeta}$ übereinstimmen und so daß für jedes jede Formel $\alpha \triangleq \mathrm{d}(X_r, x) \sim d$ aus $\mathcal{T}$ und jedes $i < |\underline{\zeta}|$ gilt:

$$\alpha \in T_i \quad \text{gdw.} \quad H'_i(r) \sim d\,. \tag{5.2}$$

Beweis. Sei $\underline{M}$ durch $M_r = \{i \mid r \in N_i\}$ mit $r < m$ gegeben.

1)⇒2) Ist $(\underline{a}, \underline{T}, \underline{N})$ die Kodierung eines Intervallwortes $u = (\underline{a}, \underline{w})$, so bestimmt $\underline{w}$ zusammen mit $\underline{\zeta}$ zwei Folgen $\underline{\rho}$ und $\underline{\rho'}$ von Uhrenständen, so daß

$$\mathrm{d}^u(\mathrm{vg}(i, M_r), i) \;=\; \rho'_i(r) \qquad \text{für } r < m \text{ und } i < |u|$$

gilt, was man durch Induktion über i einsieht. Ist demnach ν eine Belegung mit $\nu(X_i) = M_i$ für $i < m$ und $\alpha \triangleq \mathrm{d}(X_r, x) \sim d$ aus $\mathcal{T}$, dann gilt

$$(u^{\Pi\mathfrak{A}}, \nu) \models \alpha \quad \text{gdw.} \quad \rho'_{\nu(x)}(r) \sim d\,.$$

Da $(\underline{a}, \underline{T}, \underline{N})$ eine Kodierung von u und $\underline{M}$ ist, folgt daraus:

$$\alpha \in T_{\nu(x)} \quad \text{gdw.} \quad \rho'_{\nu(x)}(r) \sim d\,.$$

Es gilt also

$$\alpha \in T_{\nu(x)} \quad \text{gdw.} \quad (\rho'_{\nu(x)}/\Delta_k)(r) \sim d\,.$$

Deshalb ist 2) erfüllt, wenn wir $\underline{H} = \underline{\rho}/\Delta_k$ und $\underline{H'} = \underline{\rho'}/\Delta_k$ wählen.

2)⇒1) In der umgekehrten Richtung geht man von den Folgen $\underline{H}$ und $\underline{H'}$ aus und gewinnt aus ihnen unter Benutzung von 5.1.4 zwei Folgen $\underline{\rho}$ und $\underline{\rho'}$ von Uhrenständen und ein unäres Wort $\underline{w}$. Wir setzen $u = (\underline{a}, \underline{w})$. Durch Induktion über i weist man (5.3) nach. Wegen $\underline{H'} = \rho'/\Delta_k$ gilt also

$$\mathrm{d}^u(\mathrm{vg}(i, M_r), i) \sim d \quad \text{gdw.} \quad H'_i(r) \sim d \qquad \text{für } r < m,\ d \leq k \text{ und } i < |u|.$$

Für jede Belegung ν mit $\nu(X_i) = M_i$ für $i < m$ und jede Formel $\alpha \triangleq \mathrm{d}(X_r, x) \sim d$ aus $\mathcal{T}$ ergibt sich demnach

$$(u^{\Pi\mathfrak{A}}, \nu) \models \alpha \quad \text{gdw.} \quad H'_i(r) \sim d\,.$$

Mit (5.2) folgt schließlich:

$$(u^{\Pi\mathfrak{A}}, \nu) \models \alpha \quad \text{gdw.} \quad \mathrm{d}(X_r, x) \sim d \in T_i\,.$$

Also ist $(\underline{a}, \underline{T}, \underline{N})$ eine Kodierung von u und $\underline{M}$. □

In der Formel $\mathsf{Interpretation}_m(\mathcal{T})$ braucht also nur Bedingung 2) der obigen Behauptung formuliert zu werden.

Sei $K_0, \ldots, K_{q-1}$ eine Aufzählung der Δ_k-Klassen mit $K_0 = \eta/\Delta_k$. Seien weiterhin $\underline{J}$ und $\underline{J'}$ Folgen von jeweils q neuen Variablen.

Wir konstruieren $\mathsf{Interpretation}_m(\mathcal{T})$ in der Form $\exists\underline{J}\exists\underline{J'}\psi$. Es soll für ein Wort $u = (\underline{a}, \underline{T}, \underline{N})$ mit $\underline{\zeta}$ wie in der obigen Behauptung genau dann $(u, \nu) \models \psi$ gelten, wenn es Folgen $\underline{H}$ und $\underline{H'}$ gibt, die Bedingung 2) der obigen Behauptung erfüllen, und wenn zusätzlich für $i < q$ und $j < |u|$ gilt:

$$j \in \nu(J_i) \quad \text{gdw.} \quad H_{j+1} = K_i\,,$$
$$j \in \nu(J'_u) \quad \text{gdw.} \quad H'_j = K_i\,.$$

Dann ist $\exists\underline{J}\exists\underline{J'}\psi$ offensichtlich die gesuchte Formel.

die $\underline{J}$ bilden eine Partition des Universums
und die $\underline{J'}$ bilden eine Partition des Universums
und es existiert ein i mit $K_0\Theta_k K_i$ und 0 gehört zu J'_i
und für jedes x gilt:
  für jedes $i < q$ gilt:
    wenn $J'_i x$ und $x < x+1$, dann $J_j(x+1)$ für ein j mit $K_i\Theta_k K_j$
    für jedes $a \in A$, $T \subseteq \mathcal{T}$ und $N \subseteq m$ gilt:
      wenn $J'_i x$ und $\mathrm{P}_{(a,T,N)}x$, dann $K_i \in \mathcal{K}_T$
      wenn $J_i x$ und $\mathrm{P}_{(a,T,N)}x$, dann $J'_j x$ für das j mit $K_j = K_i\zeta_N$

Abbildung 5.1: Eine Formel, die die Menge der Kodierungen aller Intervallwörter definiert

Der Beweis schließt mit einer halbformalen Beschreibung von ψ, die in Abbildung 5.1 zu finden ist. Dabei sei für jede Menge $N \subseteq m$ die Rücksetzung ζ_N durch $\zeta_N = \{(r,0) \mid r \in N\}$ definiert und es sei für jedes $T \subseteq \mathcal{T}$ die Menge $\mathcal{K}_T$ die Menge aller Regionen H mit

$$\mathrm{d}(X_r, x) \sim d \ \in \ T \quad \text{gdw.} \quad H(r) \sim d \qquad \text{für } r < m.$$

Quantifizierte Ausdrücke wie „für jedes $a \in A$“ können als Konjunktion geschrieben werden. ■

Aus diesem Lemma, dem speziellen Charakterisierungssatz für Uhrenautomaten 3.7.2 und dem Entscheidbarkeitskriterium 2.8.2 gewinnen wir:

5.1.6 Korollar *1) Das Erfüllbarkeitsproblem der Logik* $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ *ist entscheidbar.*

2) Das Leerheitsproblem für Uhrenautomaten ist entscheidbar ([AD90]).

3) Die diskrete Projektion jeder uhrenerkennbaren Sprache ist regulär. Ein sie erkennender Automat kann effektiv aus einem Uhrenautomaten für die gegebene Sprache gewonnen werden ([AD90]).

5.1.2 Nichtabschluß unter Negation und Quantorenhierarchie

Durch die Korrespondenz zwischen Uhrenautomaten und existentiell-monadischen $\Pi^{\overleftarrow{\mathrm{d}}}$-Formeln können wir das Wissen, das wir uns im letzten Kapitel über Uhrenautomaten angeeignet haben, auf $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ übertragen. Die Aussagen gelten jeweils für endliche, unendliche und unbeschränkte Intervallwörter, was wir der Übersichtlichkeit halber nicht jedesmal erwähnen werden.

Zum einen können wir daraus, daß die Klasse **NTL** der uhrenerkennbaren Sprachen nicht unter Komplementbildung abgeschlossen ist (siehe 4.2.5), schließen:

5.1.7 Satz (Nichtabschluß unter semantischer Negation) *Die schwache Abstandslogik* $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ *ist nicht unter Negation abgeschlossen.*

Damit ist gemeint, daß es eine existentiell-monadische $\Pi^{\overleftarrow{\mathrm{d}}}$-Formel gibt, deren Negat (als $\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$-Formel) zu keiner existentiell-monadischen $\Pi^{\overleftarrow{\mathrm{d}}}$-Formel äquivalent ist. Eine solche Formel können wir 4.2.3 entnehmen:

$$\exists X \exists x (Xx \wedge \neg \exists y (x < y \wedge Xy) \wedge \forall y (x < y \rightarrow \mathrm{d}(X, y) \neq 1)) .$$

Da die Menge $\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ aller monadischen $\Pi^{\overleftarrow{\mathrm{d}}}$-Formeln definitionsgemäß schon syntaktisch unter Negation abgeschlossen ist, erhalten wir außerdem:

5.1.8 Korollar *Die Logik* $\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ *ist ausdrucksstärker als* $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$.

Unter anderem hatten wir in 3.7.3 gesehen, daß die Analogie zwischen existentiell-monadischen $\Pi^{\overleftarrow{\mathrm{d}}}$-Formeln und Uhrenautomaten so weit geht, daß d-Rang und Uhrenindex einer uhrenerkennbaren Sprache übereinstimmen.

Aus 4.3.5 erhält man deshalb als Folgerung:

5.1.9 Satz (Quantorenhierarchie) *Für jedes* $m > 0$ *ist die Sprache* L_m *(siehe 4.3.6) durch einen existentiell-monadischen* $\Pi^{\overleftarrow{\mathrm{d}}}$*-Satz definierbar, der* m *monadische Existenzquantoren besitzt, aber durch keinen existentiell-monadischen* $\Pi^{\overleftarrow{\mathrm{d}}}$*-Satz mit weniger Mengenquantoren.*

Beweis. Die obere Schranke für die Anzahl der benutzten Quantoren ist dem nachfolgenden Beispiel (siehe 5.1.10) zu entnehmen. Angenommen, man könnte L_m durch eine existentiell-monadische $\Pi^{\overleftarrow{\mathrm{d}}}$-Formel mit $m-1$ Quantoren beschreiben. Dann könnte diese Formel auch so gewählt werden, daß eine feste Variable x als obere Schranke in allen erweitert-atomaren Teilformeln erscheinen würde. Dies würde jedoch 3.7.3 und 4.3.7 widersprechen. ■

5.1.10 Beispiel (4.3.6 fortgef.) Die Sprache L_m wird durch den folgenden Satz mit m monadischen Existenzquantoren definiert:

$$\begin{aligned}
&\exists X_0 \ldots \exists X_{m-1} \\
&\quad \Big(\forall x \bigwedge_{i<m} \bigwedge_{i<i'<m} \neg (X_i x \wedge X_{i'} x) \wedge (X_0 0 \wedge X_1 \mathbf{1} \wedge \ldots \wedge X_{m-1}(\mathbf{m-1}) \\
&\quad \wedge \bigwedge_{j<m} \forall x (X_j x \leftrightarrow X_j(x+\mathbf{m})) \wedge \bigwedge_{j<m} \forall x (X_j x \rightarrow \mathrm{d}(X_j, x+\mathbf{m}-\mathbf{1}) = 1) \Big) .
\end{aligned}$$

Man beachte, daß bei der Umwandlung in eine reine existentiell-monadische $\Pi^{\overleftarrow{\mathrm{d}}}$-Formel keine zusätzlichen Mengenvariablen eingeführt werden müssen. (Vgl. dazu den Beweis von 2.7.5.)

Das Ergebnis 5.1.9 steht im Kontrast zu einem Satz von Wolfgang Thomas (siehe [Tho82]), der besagt, daß jede reguläre Sprache (endlicher oder unendlicher) diskreter Wörter durch einen existentiell monadischen Π-Satz mit einer Mengenvariablen definierbar ist.

5.1.3 Abstandsmessung in die Zukunft

Definitionsgemäß sind die in $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ benutzten erweitert-atomaren Formeln von der Gestalt

$$\mathrm{d}(X,x) \;\sim\; d\,. \tag{5.3}$$

Eine solche Formel bedeutet: „Die Gesamtdauer der Ereignisse zwischen dem Ereignis x (einschließlich) und dem letzten Ereignis davor (ausschließlich), das zu der Menge X gehört, erfüllt die Bedingung $\sim d$. Gibt es kein solches Ereignis, so erfüllt die Gesamtdauer aller Ereignisse vom Beginn des Wortes bis zum Ereignis x (einschließlich) die Bedingung $\sim d$.“ Von x aus gesehen wird also eine Dauer- oder auch Abstandsmessung in die Vergangenheit vorgenommen. Es stellt sich damit sofort die Frage, ob man auch Abstandsmessungen in die Zukunft zulassen darf, ohne daß sich die Ausdrucksstärke von $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ ändert.

Wir wollen in diesem Abschnitt beweisen, daß jede Abstandsmessung in die Zukunft auf Abstandsmessungen in die Vergangenheit zurückgeführt werden kann.

Die zu (5.3) duale Version ist:

$$\mathrm{d}(x,X) \;\sim\; d\,. \tag{5.4}$$

Wir erweitern die Signatur $\Pi^{\overleftarrow{\mathrm{d}}}$ jeweils um ein neues erweitertes Symbol $\overrightarrow{\mathrm{d}}_{\sim d}$ des Typs $(1,1)$ und bezeichnen die entstehende Signatur mit $\Pi^{\overleftrightarrow{\mathrm{d}}}$.

Um formal die Bedeutung der neuen Symbole erklären zu können, definieren wir zuerst die zu vg duale Funktion nf: Ist M eine Menge von natürlichen Zahlen und m eine natürliche Zahl, so ist der *Nachfolger von m in M*, in Zeichen $\mathrm{nf}(m,M)$, die kleinste Zahl aus M, die größer als m ist, sofern es eine solche gibt, ansonsten sei $\mathrm{nf}(m,M)$ undefiniert.

Mit jedem Intervallwort u assoziieren wir eine $\Pi^{\overleftrightarrow{\mathrm{d}}}$-Struktur $u^{\Pi^{\overleftrightarrow{\mathrm{d}}}}$, die eine Extension von $u^{\Pi^{\overleftarrow{\mathrm{d}}}}$ sein soll. Die Interpretation von $\overrightarrow{\mathrm{d}}_{\sim d}$ in $u^{\Pi^{\overleftrightarrow{\mathrm{d}}}}$ sei die Menge, die (M,m) enthalte, wenn $\mathrm{nf}(m,M)$ definiert ist und $\mathrm{d}^u(M,m,\mathrm{nf}(m,M)) \sim d$ gilt oder wenn $\mathrm{nf}(m,M)$ nicht definiert ist, u endlich ist und $\mathrm{d}^u(M,m,|u|+1) \sim d$ gilt. Für $\overrightarrow{\mathrm{d}}_{\sim d}(X,x)$ schreiben wir (5.4). Eine solche Formel ist auch eine *relative Abstandsformel*, eine sogenannte *Zukunftsformel* in X.

Damit bedeutet (5.4): „Es gibt eine Stelle in X, die gößer als x ist, und die Gesamtdauer der Ereignisse zwischen x (einschließlich) und dem nächsten Ereignis danach (ausschließlich), das zu X gehört, erfüllt die Bedingung $\sim d$. Oder das gegebene Wort

ist endlich, es gibt kein auf x folgendes Ereignis, das zu X gehört, und die Gesamtdauer der Ereignisse von x (einschließlich) bis zum Ende des Wortes erfüllt die Bedingung $\sim d$."

Die Gleichwertigkeit von $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ und $\exists\mathrm{MSO}\Pi\overleftrightarrow{\mathrm{d}}$ wird in dem folgenden Satz ausgedrückt:

5.1.11 Satz *Jeder* $\exists\mathrm{MSO}\Pi\overleftrightarrow{\mathrm{d}}$*-Satz kann effektiv in einen äquivalenten* $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$*-Satz umgewandelt werden.*

Beweis. Nach 2.7.6 reicht es, eine äquivalente existentiell-monadische $\Pi'\overleftarrow{\mathrm{d}}$-Formel mit Mengentermen zu konstruieren.

Sei dazu $\phi \triangleq \exists X_0 \ldots \exists X_{l-1}\phi_0$ eine $\exists\mathrm{MSO}\Pi\overleftrightarrow{\mathrm{d}}$-Formel, die als Teilformel eine Zukunftsformel in X enthalte. Wir konstruieren eine zu ϕ äquivalente Formel ψ, die keine neuen Zukunftsformeln und zusätzlich keine Zukunftsformel in X enthält. Eine wiederholte Anwendung der Konstruktion liefert dann die gewünschte Formel.

Wir behandeln den Fall der unendlichen Intervallwörter und erläutern am Ende des Beweises die Modifikationen, die im Fall der endlichen Intervallwörter vorgenommen werden müssen.

Es sei m eine obere Schranke für die Vergleichswerte, die in den Teilformeln von ϕ, die Zukunftsformeln in X sind, auftreten.

Die gesuchte Formel ψ soll die Gestalt

$$\exists X_0 \ldots X_{l-1} \exists G_0 \ldots \exists G_m \exists E_0 \ldots \exists E_m \exists K (\mathsf{Kodierung}_m(\underline{G}, \underline{E}, K, X) \wedge \phi_0')$$

haben. Dabei soll für jede Interpretation $\mathfrak{I}$ die Beziehung $\mathfrak{I} \models \mathsf{Kodierung}_m$ genau dann gelten, wenn die beiden folgenden Bedingungen für jedes c mit $c \leq m$ erfüllt sind:

$$\mathfrak{I} \models \forall x(E_c x \leftrightarrow \mathrm{d}(x, X) = c) \quad \text{und} \quad \mathfrak{I} \models \forall x(G_c x \leftrightarrow \mathrm{d}(x, X) > c)\,;$$

und es soll K mit den Elementen belegt sein, die keinen X-Nachfolger besitzen. Die E-Variablen sollen also die ‚gleich'-Bedingungen und die G-Variablen die ‚größer'-Bedingungen kodieren. Die Formel ϕ_0' entstehe aus der Formel ϕ_0 durch Substitution der erweitert-atomaren Formeln der Gestalt $\mathrm{d}(y, X) \sim c$ durch α gemäß folgender Tabelle:

$\sim$	$=$	$>$	$<$	$\geq$	$\leq$	$\neq$
α	$E_c y$	$G_c y$	$\neg E_c y \wedge \neg G_c y \wedge \neg K y$	$E_c y \vee G_c y$	$\neg G_c y \wedge \neg K y$	$\neg E_c y \wedge \neg K y$

Aus der Konstruktion von ϕ_0' wird ersichtlich, daß die neue Formel ψ zu ϕ äquivalent ist, sofern $\mathsf{Kodierung}_m$ die geforderten Eigenschaften hat. Es reicht also, ein Konstruktionsverfahren für $\mathsf{Kodierung}_m$ anzugeben.

Wir wählen ein induktives Verfahren und beginnen mit $m = 0$. In diesem sei

$$\mathsf{Kodierung}_0 \quad \triangleq \quad \forall x(\neg E_0 x \wedge (Kx \leftrightarrow \neg\exists y(x < y \wedge Xx)) \wedge (G_0 x \leftrightarrow \neg Kx))$$

$\mathsf{Kodierung}_{m-1}$ und
für alle x aus X gilt:
  falls es ein y unter x mit $E_m y$ gibt, so gilt:
    y ist eindeutig
    und $\mathrm{d}(E_m - 1, x - 1) = m$
    und $G_m y'$ für alle y' unter x mit $y' < y$
    und $G_m y'$ für kein y' unter x mit $y \leq y'$
  und falls es kein y unter x mit $E_m y$ aber eins mit $G_m y$ gibt,
    gilt für das größte dieser Art:
    $\mathrm{d}(G_m - 1, x - 1) > m$
    und $y + 1 = x$ oder nicht $\mathrm{d}(G_m, x - 1) > m$
    und $G_m y'$ für alle y' unter x mit $y' < y$
  und falls es kein y unter x mit $E_m y$ oder $G_m y$ gibt, gilt:
    nicht $\mathrm{d}(X - 1, x - 1) \geq m$
und für alle x aus K gilt:
  x gehört weder zu E_m noch zu G_m

Abbildung 5.2: Zur Eliminierung von Zukunftsformeln

Es steht „ein y liegt unter x“ dafür, daß y kleiner als x ist und daß zwischen y und x kein Element aus X liegt. Außerdem ersetzt $Z - 1$ den Mengenterm $\{x \mid \neg x = x + 1 \wedge Z(x + 1)\}$.

(Beachte, daß $\mathrm{d}(x, X) = 0$ nie gilt und $\mathrm{d}(x, X) > 0$ nur dann nicht, wenn es kein Element in X gibt, das größer als x ist.)

Der Übersichtlichkeit halber wählen wir im Induktionsschritt ($m > 0$) eine halbformale Darstellung der Formel, die ohne weiteres in eine reine $\exists\mathrm{MSO}\Pi'\overleftarrow{\mathrm{d}}$-Formel mit Mengentermen umgesetzt werden kann. Die Formel ist in Abbildung 5.2 zu finden.

Wir benutzen die Sprechweise „ein y liegt unter x“, die bedeuten soll, daß y kleiner als x ist und daß zwischen y und x kein Element aus X liegt. (Dies kann z. B. durch $x < y \wedge \neg\exists x'(x < x' < y \wedge Xx')$ ausgedrückt werden.) Es stehe $Z - 1$ für den Mengenterm $\{x \mid \neg x = x + 1 \wedge Z(x + 1)\}$.

Im Fall der endlichen Intervallwörter ist die Variable K überflüssig, da definitionsgemäß für jede Stelle ohne X-Nachfolger stattdessen max genommen wird. Dementsprechend muß die universelle Quantifizierung in Abbildung 5.2 nicht nur alle Elemente aus X durchlaufen, sondern es muß auch max in Betracht gezogen werden. ■

5.1.4 Eindeutigkeit

In Abschnitt 4.5 haben wir eindeutige Uhrenautomaten eingehend untersucht. Auf der logischen Seite entspricht einem eindeutigen Automaten eine ‚eindeutige‘ existentiell-monadische Formel:

5.1.12 Definition (eindeutige Formeln) Sei Ω eine Wortsignatur. Eine existentiell-monadische Ω-Formel $\phi \triangleq \exists X_0 \ldots \exists X_{m-1} \phi_0$ heißt *eindeutig*, wenn es für jede Ω-Wortstruktur $\mathfrak{M}$ höchstens eine erfüllende Interpretation $(\mathfrak{M}, \nu)$ von ϕ_0 gibt.

Wir beweisen zuerst die Korrespondenz zwischen eindeutigen Uhrenautomaten und eindeutigen $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$-Formeln.

5.1.13 Lemma *Eine uhrenerkennbare Sprache ist genau dann eindeutig erkennbar, wenn sie durch einen eindeutigen* $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$*-Satz definiert wird.*

Beweis. Zuerst bemerken wir, daß die im speziellen Charakterisierungssatz für Uhrenautomaten (siehe 3.7.2) vorgestellten Konstruktionen zur Umwandlung von existentiell-monadischen Σ_{UA}-Formeln in existentiell-monadische $\Pi\overleftarrow{\mathrm{d}}$-Formeln und umgekehrt die Eigenschaft der Eindeutigkeit erhalten. Es reicht also, wenn wir anstelle von $\Pi\overleftarrow{\mathrm{d}}$-Formeln Σ_{UA}-Formeln betrachten.

($\Rightarrow$) Die Formel akzBer aus Abschnitt 2.4 hat für ein gegebenes Intervallwort u genauso viele erfüllende Interpretationen $(u^{\Sigma_{\mathrm{UA}}}, \nu)$, wie es akzeptierende Berechnungen des gegebenen Automaten auf u gibt. (Vgl. auch den Beweis von 4.5.3.)

($\Leftarrow$) Um dies einzusehen, analysieren wir die Konstruktion im Beweis der Implikation von 1) nach 2) im Charakterisierungssatz 2.3.2, siehe S. 33. Dazu sei eine eindeutige existentiell-monadische Σ_{UA}-Formel ϕ gegeben. Aus 2.5.3 folgt, daß für jedes Intervallwort höchstens eine Kodierung die Formel ϕ^* erfüllt. Also wird auch von dem zu ϕ^* äquivalenten Automaten $\mathfrak{A}$ höchstens eine Kodierung eines Intervallwortes erkannt. Aus der Konstruktion von $\hat{\mathfrak{A}}$ (vgl. auch den Beweis von 2.6.1) ergibt sich dann aber, daß $\hat{\mathfrak{A}}$ höchstens eine akeptierende Berechnung auf jedem Intervallwort ausführt. ■

Es sei nebenbei bemerkt, daß man sich mit der gleichen Überlegung leicht davon überzeugen kann, daß dieses Lemma allgemein für „eindeutige" Σ-Automaten und eindeutige $\exists\mathrm{MSO}\Pi\Sigma$-Formeln gilt.

Bevor wir zum Beweis von 4.4.1 übergehen, wollen wir noch eine wichtige Eigenschaft der im Beweis von 5.1.11 konstruierten Formel festhalten:

5.1.14 Lemma *Ist* ϕ *eine eindeutige existentiell-monadische* $\Pi\overleftrightarrow{\mathrm{d}}$*-Formel, so gibt es dazu eine äquivalente eindeutige existentiell-monadische* $\Pi\overleftarrow{\mathrm{d}}$*-Formel.*

Beweis. Man betrachte dazu die im Beweis von 5.1.11 vorgestellte Konstruktion zur Umwandlung einer existentiell-monadischen $\Pi\overleftarrow{\mathrm{d}}$-Formel ϕ in eine äquivalente existentiell-monadische $\Pi\overleftrightarrow{\mathrm{d}}$-Formel ϕ': Die Formel $\mathsf{Kodierung}_m$ ist derart, daß die nach dem Verfahren zur Elimination von Mengentermen (siehe 2.7.4) entstehende äquivalente $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$-Formel eindeutig ist. Außerdem ist ϕ'_0 eindeutig, sofern die Ausgangsformel eindeutig ist. Damit ist die gesamte Formel eindeutig. ■

Beweis von Satz 4.4.1 Wegen 5.1.13 und 5.1.14 reicht es, wenn wir zu jedem beschränkten, deterministischen Zweiwege-Uhrenautomaten eine äquivalente eindeutige $\Pi\overleftrightarrow{\mathrm{d}}$-Formel konstruieren. Mit einigen Abänderungen versehen kann die in Abschnitt 2.4 konstruierte Formel akz übernommen werden, die die Existenz einer akzeptierenden Berechnung beschreibt. Dabei werden Mengenvariablen für die Kodierung der die Berechnung bestimmenden Transitionsfolge und Abstandsformeln zur Überprüfung der Registertests benutzt.

Durch die k-Beschränkung wird die Kodierung der Transitionsfolge möglich. Man benutzt die k-fache Anzahl von Mengenvariablen zuzüglich von Hilfsvariablen, die die Umkehrpunkte eines Laufes beschreiben. Entscheidend bei der Überprüfung der Uhrentests ist, daß auch Abstandsvergleiche in die Zukunft zur Verfügung stehen. Diese ermöglichen nämlich die Überprüfung eines Uhrentests an Stellen, an denen eine Uhr getestet werden soll, die zum letzten Mal an einer Stelle rechts von der aktuellen Stelle zurückgesetzt wurde.

Die Eindeutigkeit der Formel wird durch den Determinismus des gegebenen Zweiwege-Uhrenautomten garantiert. ■

5.2 Unentscheidbare Erweiterungen und Modifikationen von $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$

Ausschlaggebend für die Entscheidbarkeit des Erfüllbarkeitsproblems von $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ ist der Umstand, daß direkte Abstandsvergleiche zwischen beliebigen Momenten nicht möglich sind. Wir wollen dies exakt formulieren.

5.2.1 Definition (Signatur $\Pi\mathrm{d}$) Die Signatur $\Pi\mathrm{d}$ entsteht aus Π durch Hinzunahme eines binären Symbols $\mathrm{d}_{\sim c}$ für jede Vergleichsrelation $\sim$ und jede natürliche Zahl c.

Aus jedem Intervallwort $u = (\underline{a}, \underline{w})$ erhalten wir nun eine $\Pi\mathrm{d}$-Struktur $u^{\Pi\mathrm{d}}$, indem wir festlegen, daß $\mathrm{d}_{\sim c}$ in $u^{\Pi\mathrm{d}}$ durch die Menge

$$\{(i,j) \mid i \leq j \wedge W_{i+1} - W_j \sim c\}$$

interpretiert wird und $u^{\Pi\mathrm{d}} \restriction \Pi = \underline{a}^{\Pi}$ fordern. D. h., $\mathrm{d}_{\sim c}(x,y)$ trifft zu, wenn der Abstand zwischen Position x und Position $y+1$ die Bedingung $\sim c$ erfüllt. Deshalb schreiben wir für $\mathrm{d}_{\sim c}(x,y)$ auch einfach

$$\mathrm{d}(x,y) \quad \sim \quad c\,.$$

Diese Schreibweise steht mit den Vereinbarungen in Unterabschnitt 3.7.1 in Einklang.

5.2.2 Definition (volle Abstandslogik) Die Sprache $\mathrm{MSO}\Pi\mathrm{d}$ wird auch als *volle Abstandslogik* bezeichnet.

Offensichtlich läßt sich die Formel $\mathrm{d}(X,x) \sim c$ äquivalent in

$$\exists y(y < x \wedge Xy \wedge \neg\exists y'(y < y' \wedge y' < x \wedge Xy') \wedge \mathrm{d}(y,x) \sim c)$$
$$\vee\ (\neg\exists y(Xy \wedge y < x) \wedge \mathrm{d}(0,x) \sim c)$$

umformen. Daraus folgt:

5.2.3 Bemerkung Jeder existentiell-monadische $\Pi\overleftarrow{\mathrm{d}}$-Satz ist äquivalent zu einem existentiell-monadischen Πd-Satz.

In [AFH91] ist nun schon das folgende Resultat zu finden:

5.2.4 Satz ([AFH91][1]) *Die in der Sprache der ersten Stufe in der Signatur* $\Pi \cup \{d_{=1}\}$ *formulierbare Theorie der Intervallwörter ist unentscheidbar. (Dies gilt sowohl für endliche wie auch für unendliche und unbeschränkte Intervallwörter. Es wird jeweils vorausgesetzt, daß das Alphabet mindestens drei Elemente besitzt.)*

Die Theorie der unendlichen bzw. der unbeschränkten Intervallwörter ist sogar Π^1_1*-schwierig.*

Der Beweis erfolgt durch Reduktion des Halteproblems für Zweiregistermaschinen (vgl. 4.4.1). ■

Ein weiteres negatives Ergebnis über mögliche Erweiterungen der schwachen Abstandslogik erhalten wir aus den Untersuchungen in Abschnitt 4.7 (vgl. auch Abbildung 4.7):

5.2.5 Satz *Ist* P *eine Teilmenge von* $\mathbf{Z}$*, die mindestens zwei Elemente besitzt, dann ist das Erfüllbarkeitsproblem von* $\exists\mathrm{MSO}\Pi\Sigma_P$ *unentscheidbar.* ■

Damit ist $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ zumindest in dem Rahmen, der durch hybride Automaten und Σ-Automaten vorgegeben ist, die ausdruckstärkste Logik mit entscheidbarem Erfüllbarkeitsproblem.

Abschließend sei noch erwähnt:

5.2.6 Satz *Die* $\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$*-Theorie der endlichen, unendlichen bzw. unbeschränkten Intervallwörter ist unentscheidbar.*

Beweis. Dazu wird das für Uhrenautomaten unentscheidbare Universalitätsproblem (siehe [AD90] bzw. [AD]) auf das gegebene Problem reduziert: Zu einem gegebenen Uhrenautomaten konstruiere man einen äquivalenten $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$-Satz ϕ und überprüfe, ob $\neg\phi$ erfüllbar ist. Ist dies der Fall, dann ist die Antwort auf das Universalitätsproblem ‚nein', andernfalls ‚ja'.

[1] In der angegebenen Arbeit [AFH91] wird dieses Ergebnis für die Logik MTL formuliert, ist aber auf die hier angegebene Logik erster Stufe direkt übertragbar.

Man vergleiche hierzu auch die klassischen Ergebnisse [TM51] und [She75] über die rellen Zahlen.

Für die Unentscheidbarkeit des Universalitätsproblems wird in [AD] ein mindestens dreielementiges Alphabet vorausgesetzt. Auf diese Einschränkung kann hier verzichtet werden, weil Buchstabenprädikate durch entsprechende monadische Variablen ersetzt werden können. ■

5.3 PTL und Erweiterungen um Zeitoperatoren nach Manna und Pnueli

Die temporalen Logiken mit Zeitabhängigkeit, die in diesem und im nächsten Unterabschnitt behandelt werden sollen, können aus syntaktischer und semantischer Sicht als Erweiterung der propositionalen Zeitlogik PTL ([Pnu77]) angesehen werden, weshalb wir diese noch einmal kurz vorstellen wollen.

5.3.1 PTL

Die Syntax von PTL ist durch das Regelsystem in Abbildung 5.3 beschrieben, in dem ϕ und ψ für Formeln stehen und a für einen Buchstaben des Alphabets steht.

$$\mathtt{tt} \mid \mathrm{p}_a \mid \phi \wedge \psi \mid \neg\phi \mid \bigcirc\phi \mid \ominus\phi \mid \phi\,\mathcal{U}\,\psi \mid \phi\,\mathcal{S}\,\psi$$

Abbildung 5.3: Die Syntax von PTL

PTL-Formeln werden in diskreten Wörtern interpretiert. Ist $u = \underline{a}$ ein diskretes Wort und $i < |u|$, so wird das Paar (u, i) als *Interpretation* bezeichnet. Induktiv wird $(u, i) \models \phi$ nach folgenden Regeln definiert:

Es gilt immer $(u, i) \models \mathtt{tt}$, und es gilt $(u, i) \models \mathrm{p}_a$, falls $a_i = a$ gilt. Damit steht p_a für: „Ereignis i hat den Typ a“. Negation und Konjunktion werden wie üblich behandelt.

Es gilt $(u, i) \models \bigcirc\phi$, falls $i + 1 < |u|$ ist und $(u, i + 1) \models \phi$ gilt. Entsprechend gilt $(u, i) \models \ominus\phi$, falls $i > 0$ ist und $(u, i - 1) \models \phi$ gilt. Damit steht $\bigcirc$ für die nächste Stelle und $\ominus$ für die vorhergehende.

Es gilt $(u, i) \models \phi\,\mathcal{U}\,\psi$, wenn es eine Stelle j mit $j \geq i$ gibt, so daß $(u, j) \models \psi$ und zusätzlich $(u, i') \models \phi$ für jede Stelle i' mit $i \leq i' < j$ gilt. Entsprechend gilt $(u, i) \models \phi\,\mathcal{S}\,\psi$, wenn es eine Stelle j gibt, die kleiner als oder gleich i ist und so daß $(u, j) \models \psi$ und zusätzlich $(u, j') \models \phi$ für jede Stelle j' mit $j < j' \leq i$ gilt. Es steht also $\phi\,\mathcal{U}\,\psi$ für „es gilt ϕ, bis irgendwann später ψ gilt“ und $\phi\,\mathcal{S}\,\psi$ für: „es gilt ϕ, seit irgendwann vorher einmal ψ galt“.

Ein Wort u *erfüllt* eine PTL-Formel ϕ, ist ein *Modell* von ϕ, falls $(u,0) \models \phi$ gilt. Dadurch definiert eine PTL-Formel eine Sprache von diskreten Wörtern, nämlich die Menge ihrer Modelle.

In [Wol83] wurde PTL um sogenannte Automatenoperatoren erweitert, was ebenfalls kurz erläutert werden soll. Die Elemente der Menge 2^n, also die Funktionen $n \to 2$, werden im folgenden auch als Spaltenvektoren mit n Zeilen und Einträgen aus $\{0,1\}$ geschrieben.

Unter einer *Automatenformel* versteht man Formeln der Gestalt

$$\mathfrak{A}^+(\phi_0,\ldots,\phi_{n-1}) \quad \text{oder} \quad \mathfrak{A}^-(\phi_0,\ldots,\phi_{n-1}), \tag{5.5}$$

wobei $\mathfrak{A}$ für einen Rabin-Scott-Automaten über dem Alphabet 2^n steht und ϕ_0, ..., ϕ_{n-1} schon konstruierte Formeln sind.

Eine Automatenformel ist an einer Stelle wahr, wenn sie beim Lesen der Wahrheitswerte der Formeln, aus denen sie zusammengesetzt ist, einen Endzustand erreicht. Dabei läßt man den Automaten nach rechts $(+)$ oder links $(-)$ laufen. Dies wird im folgenden formalisiert.

Ist u ein Wort und ist ϕ_0, ..., ϕ_{n-1} eine Folge von Formeln, so kodieren wir die Wahrheitswerte, die die Aussage $(u,i) \models \phi_j$ jeweils annimmt, in einem diskreten Wort $\underline{x}$ über 2^n gemäß der folgenden Gleichung:

$$x_i(j) \;=\; \begin{cases} 1\,, & \text{falls } (u,i) \models \phi_j, \\ 0\,, & \text{falls nicht } (u,i) \models \phi_j. \end{cases}$$

Damit ist die j-te Komponente im i-ten Vektor genau dann Eins, wenn $(u,i) \models \phi_j$ gilt. Das Wort $\underline{x}$ wird mit $[u,\underline{\phi}]$ bezeichnet.

Ist $\mathfrak{I} = (u,i)$ eine Interpretation und $\phi \triangleq \mathfrak{A}^+(\phi_0,\ldots,\phi_{n-1})$, so gilt $\mathfrak{I} \models \phi$, falls es eine Position j mit $j \geq i$ gibt, so daß $[u,\underline{\phi}](i,j+1)$ zu der von $\mathfrak{A}$ erkannten Sprache gehört. D. h., die Formel ϕ ist wahr an Stelle i, wenn es eine Stelle j in der Zukunft gibt, so daß die Wahrheitswerte der Formeln aus $\underline{\phi}$ zwischen beiden Positionen ein Wort ergeben, das von $\mathfrak{A}$ akzeptiert wird.

Symmetrisch dazu gilt $\mathfrak{I} \models \mathfrak{A}^-(\phi_0,\ldots,\phi_{n-1})$, wenn es eine Position j mit $j \leq i$ gibt, so daß $[u,\underline{\phi}](i+1,j)$ von $\mathfrak{A}$ akzeptiert wird. (Man beachte die Konventionen zur Beschreibung von Segmenten aus dem ersten Abschnitt der Arbeit, „Notation".)

Hat man Automatenformeln zur Verfügung, so werden der Bis- und der Seit-Operator sowie $\bigcirc$ und $\ominus$ entbehrlich:

5.3.1 Bemerkung 1) Ist $\mathfrak{A}$ ein endlicher diskreter Rabin-Scott-Automat, der die Sprache $2^2(\binom{1}{0} + \binom{1}{1})^*(\binom{0}{1} + \binom{1}{1})$ erkennt, so ist $\phi \mathcal{U} \psi$ zu $\mathfrak{A}^+(\phi,\psi)$ und $\phi \mathcal{S} \psi$ zu $\mathfrak{A}^-(\phi,\psi)$ äquivalent.

2) Ist $\mathfrak{B}$ ein endlicher diskreter Rabin-Scott-Automat, der die Sprache, die die beiden Wörter 01 und 11 der Länge 2 enthält, erkennt, so ist $\bigcirc\phi$ zu $\mathfrak{B}^+(\phi)$ und $\ominus\phi$ zu $\mathfrak{B}^-(\phi)$ äquivalent.

5.3.2 Die Erweiterung um den Altersoperator

Die erste Erweiterung von PTL um zeitabhängige Konstrukte, die hier betrachtet wird, heißt TL_Γ (siehe [MP93]). Man erhält ein Regelsystem für TL_Γ, indem man zu dem System in Abbildung 5.3 die Regel

$$\Gamma(\phi) \quad \sim \quad d$$

hinzunimmt, wobei $\sim$ eine der üblichen Vergleichsrelationen und d eine natürliche Zahl ist.

Eine Formel dieser Logik wird ähnlich wie eine PTL-Formel über einem Intervallwort zusammen mit einem seiner Stellen interpretiert.

Sei also $\mathfrak{I} = (u, i)$ eine entsprechende Interpretation mit $u = (\underline{a}, \underline{w})$. Die schon aus PTL bekannten Sprachkonstrukte behalten ihre Bedeutung, wobei man die Bewertung von u einfach ignoriert.

Auf die Bewertung wird einzig und allein durch den sogenannten *Altersoperator* (‚age'-Operator) Γ Bezug genommen: Falls $\mathfrak{I} \models \phi$ nicht gilt, so gelte $(u, i) \models \Gamma(\phi) \sim d$, falls $0 \sim d$ gilt. Andernfalls gibt es eine kleinste natürliche Zahl j mit der Eigenschaft, daß für alle j' mit $j \leq j' \leq i$ die Beziehung $(u, j') \models \phi$ erfüllt ist. Dann gelte $(u, i) \models \Gamma(\phi) \sim d$, falls $W_{i+1} - W_j \sim d$ gilt. Damit gibt $\Gamma(\phi)$ an, wie lange ϕ an der Position $i + 1$ schon gültig ist, in gewissem Sinne also das „Alter" von ϕ.

Ein Intervallwort erfüllt eine TL_Γ-Formel ϕ, falls ϕ an der Stelle Null erfüllt ist. Damit ist die Modellmenge einer TL_Γ-Formel ϕ genau wie bei PTL durch $\{u \mid (u, 0) \models \phi\}$ gegeben.

In [MP93] werden TL_Γ-Formeln zur Spezifikation zeitabhängiger Systeme benutzt, und zur Verifikation geben Manna und Pnueli Schlußregeln an. Wir können durch eine effektive Einbettung von TL_Γ in $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ die Entscheidbarkeit der Theorie von TL_Γ nachweisen. Insbesondere ergibt sich daraus, daß das Modellproblem für Uhrenautomaten und TL_Γ entscheidbar ist.

5.3.2 Lemma *Jede* TL_Γ*-Formel kann effektiv in eine äquivalente* $\Pi'^{\overleftarrow{\mathrm{d}}}$*-Formel mit Mengentermen umgewandelt werden, die ohne Mengenvariablen auskommt.*

Beweis. Induktiv wird zu jeder TL_Γ-Formel ϕ eine $\Pi'^{\overleftarrow{\mathrm{d}}}$-Formel $\phi^* \triangleq \phi^*(x)$ konstruiert, die die Menge aller Stellen definiert, an denen ϕ wahr ist: Es soll $(u, i) \models \phi$ genau dann gelten, wenn $(u^{\Pi'^{\overleftarrow{\mathrm{d}}}}, \nu) \models \phi^*$ für eine (jede) Belegung ν mit $\nu(x) = i$ gilt. Dann ist die Formel $\phi^*(0)$ zu ϕ äquivalent.

Für $\mathtt{tt}$ ist die Behauptung trivial. Ist ϕ gleich p_a, so setzt man $\phi^* = \mathrm{P}_a x$. Konjunktion und Negation übertragen sich wie üblich.

Für $\bigcirc\phi$ verfährt man wie folgt, wenn ϕ^* schon konstruiert wurde:

$$(\bigcirc\phi)^*(x) \triangleq \neg x = x+1 \wedge \phi^*(x+1)\,.$$

Dementsprechend wird auch $(\ominus\phi)^*$ konstruiert.

Für das Bis-Konstrukt ergibt sich

$$(\phi\,\mathcal{U}\psi)^* \triangleq \exists y(x < y \wedge \psi^*(y) \wedge \forall x'(x < x' < y \rightarrow \phi^*(x'))\,.$$

Eine entsprechende Konstruktion wird für eine Seit-Formel angewandt.

Zum Schluß behandeln wir den Altersoperator. Dazu betrachten wir die Formel $\Gamma(\phi) \sim d$. Die Formel $(\Gamma(\phi) \sim d)^*$ trifft eine Fallunterscheidung danach, ob ϕ an der jeweiligen Stelle gilt oder nicht:

$$(\Gamma(\phi) \sim d)^* \triangleq (\neg\phi^*(x) \rightarrow 0 \sim d) \wedge (\phi^*(x) \rightarrow \mathrm{d}(\{x \mid \neg\phi^*(x)\}, x) \sim d)\,.$$

Hierbei stehe $0 \sim d$ für den entsprechenden Wahrheitswert. ■

5.3.3 Satz *1) Das Erfüllbarkeitsproblem ist für* TL_Γ *entscheidbar.*

2) Die TL_Γ*-Theorie der Intervallwörter ist entscheidbar.*

3) Es ist entscheidbar, ob ein durch einen Uhrenautomaten gegebenes zeitabhängiges System eine Spezifikation in TL_Γ *erfüllt.*

Beweis. 1) Dies folgt aus 5.3.2 und der Entscheidbarkeit des Erfüllbarkeitsproblems für $\exists\mathrm{MSOII}\overleftarrow{\mathrm{d}}$ (siehe 5.1.6) unter Beachtung der Ergebnisse über Spracherweiterungen existentiell-monadischer Logiken, die wir in Abschnitt 2.7 erzielt haben.

2) Dies ist eine Folgerung aus 1), da TL_Γ unter Negation abgeschlossen ist.

3) Wir skizzieren ein Entscheidungsverfahren für das fragliche Modellproblem. Dazu gehen wir von einem Uhrenautomaten $\mathfrak{A}$ und einer TL_Γ-Spezifikation ϕ aus. Zu $\mathfrak{A}$ konstruieren wir eine äquivalente $\exists\mathrm{MSOII}\overleftarrow{\mathrm{d}}$-Formel ψ und zu ϕ die Formel $(\neg\phi)^*$ wie im Beweis von 5.3.2. Dann ist

$$\chi \triangleq \psi \wedge (\neg\phi)^*$$

eine existentiell-monadische $\Pi'\overleftarrow{\mathrm{d}}$-Formel mit Mengentermen, die genau dann erfüllbar ist, wenn es ein Intervallwort gibt, das von $\mathfrak{A}$ erkannt wird und ψ nicht erfüllt, d. h., wenn $\mathfrak{A}$ die Spezifikation ϕ nicht erfüllt. Wir brauchen also χ nur in eine äquivalente $\Pi\overleftarrow{\mathrm{d}}$-Formel umzuwandeln und anschließend die Antwort auf den Erfüllbarkeitstest zu negieren, um eine Antwort auf das Modellproblem zu erhalten. ■

5.3.3 Die Erweiterung um den Daueroperator

In [MP93] wird außer der Logik TL_Γ auch die umfassendere Logik $\mathrm{TL}_{\Gamma\int}$ betrachtet. Diese gewinnt man aus TL_Γ unter Hinzunahme der Regel

$$\int \phi \quad \sim \quad d\,. \tag{5.6}$$

Eine Interpretation (u, i) erfüllt diese Formel, falls

$$\sum_{j \le i : (u,j) \models \phi} w_j \quad \sim \quad d \tag{5.7}$$

gilt. Damit steht (5.6) für die Aussage: „die Gesamtdauer der Ereignisse, die an Stellen stehen, an denen ϕ galt, zuzüglich der Dauer des aktuellen Ereignisses, falls ϕ momentan gilt, erfüllt die Bedingung $\sim d$".

Analog zu 5.3.2 können wir hier zeigen:

5.3.4 Lemma *Jede* $\mathrm{TL}_{\Gamma\int}$*-Formel läßt sich effektiv in eine äquivalente* Π'dur*-Formel mit Mengentermen umwandeln, die ohne Mengenvariablen auskommt.*

Beweis. Wie im Beweis von 5.3.3 konstruiert man zu jeder $\mathrm{TL}_{\Gamma\int}$-Formel ϕ eine Πdur-Formel $\phi^* \triangleq \phi^*(x)$, die die Menge aller Stellen definiert, an denen ϕ wahr ist. Aufgrund der schon im Beweis von 5.3.3 angestellten Überlegungen, reicht es hier, wenn wir den Induktionsschritt für den Daueroperator durchführen.

Dazu sei $\phi \triangleq \int \phi_0 \sim d$ gegeben. Dann ist

$$\mathrm{dur}(\emptyset, \{x \mid \phi_0^*\}, x) \quad \sim \quad d$$

die gesuchte Formel. ■

Daraus können wir schließen:

5.3.5 Satz *Zu jedem* $\mathrm{TL}_{\Gamma\int}$*-Satz läßt sich effektiv ein äquivalenter Uhrenautomat mit Haltemöglichkeit konstruieren.*

Beweis. Dies folgt aus 5.3.4 unter Verwendung des speziellen Charakterisierungssatzes für Uhrenautomaten mit Haltemöglichkeit 3.7.9 und den Ergebnissen aus Abschnitt 2.7. ■

5.4 Metrische Zeitlogiken

Im Gegensatz zu der Erweiterung von PTL durch zusätzliche Operatoren, die auf die Bewertung der zugrundeliegenden Intervallwörter zurückgreifen, werden in den sogenannten metrischen Logiken die schon vorhandenen logischen Quantoren Bis und Seit mit zeitlichen Einschränkungen versehen, die durch reelle Intervalle angegeben werden.

5.4.1 Definition (reelle Intervalle) Es ist $\mathcal{K}$ die Menge aller nichtleeren (offenen, halboffenen, geschlossenen) Intervalle in $\mathbf{P}_0$ mit Grenzen in $\mathbf{N}_\infty$. Mit $\mathcal{K}_0$ wird die Teilmenge von $\mathcal{K}$ bezeichnet, die aus den Intervallen besteht, die mindestens zwei Elemente besitzen. Diese Intervalle werden auch als *nichtsingulär* bezeichnet.

Z. B. gehören die Intervalle $[3,3]$, $(0,5]$ und $[13,\infty)$ zu $\mathcal{K}$, nicht aber jedoch $[3,3]$ zu $\mathcal{K}_0$.

5.4.2 Bemerkung Für jedes Intervall I aus $\mathcal{K}$ gibt es Vergleichsrelationen $\sim_0$ und $\sim_1$ und natürliche Zahlen d_0 und d_1, so daß $I = \{r \in \mathbf{P}_0 \mid r \sim_0 d_0 \wedge r \sim_1 d_1\}$ gilt.

5.4.1 Metrische Zeitlogiken nach Alur und Henzinger

Die Logiken MTL („metric temporal logic"), MITL („metric interval temporal logic") und MITL$^{\mathrm{P}}$ („metric interval temporal logic with past operators") wurden in [AH90], [Alu91] bzw. [AH92] eingeführt. Ein einheitlicher Zugang wird durch die Einführung von MTL$^{\mathrm{P}}$ gewährleistet. Die Syntax dieser Sprache ist in dem Diagramm in Abbildung 5.4 wiedergegeben. Darin stehen ϕ und ψ für Formeln, es steht I für ein Intervall aus $\mathcal{K}$ und a für einen beliebigen Buchstaben des Alphabets.

$$\mathtt{tt} \mid \mathrm{p}_a \mid \phi \wedge \psi \mid \neg\phi \mid \phi\,\mathcal{U}_I\,\psi \mid \phi\,\mathcal{S}_I\,\psi$$

Abbildung 5.4: Die Syntax von MTL$^{\mathrm{P}}$

Aus MTL$^{\mathrm{P}}$ ergibt sich MTL durch Weglassen der letzten Regel (siehe [AH90]), MITL$^{\mathrm{P}}$ durch die Einschränkung auf nichtsinguläre Intervalle (also auf Intervalle aus $\mathcal{K}_0$, siehe [AH92]) und MITL durch Weglassen der letzten Regel und zusätzliche Einschränkung auf nichtsinguläre Intervalle (siehe [AFH91]).

Wie bei TL$_\Gamma$ und TL$_{\Gamma\int}$ werden die Formeln von MTL$^{\mathrm{P}}$ in einem Intervallwort zusammen mit einer Stelle interpretiert. Die Bedeutung der atomaren Formeln und der booleschen Kombinationen wird übernommen.

Zur Erläuterung der anderen Sprachkonstrukte setzen wir eine Interpretation $\mathfrak{I} = (u,i)$ als gegeben voraus.

Es gilt $\mathfrak{I} \models \phi\,\mathcal{U}_I\,\psi$ genau dann, wenn es ein j gibt, so daß $j \geq i$, $W_{j+1} - W_i \in I$, $(u,j) \models \psi$ und $(u,i') \models \phi$ für alle i' mit $i \leq i' < j$ gelten. Entsprechend erfolgt die Interpretation von $\phi\,\mathcal{S}_I\,\psi$: Es gilt $\mathfrak{I} \models \phi\,\mathcal{S}_I\,\psi$ genau dann, wenn es eine Position j gibt, so daß $j \leq i$, $W_{i+1} - W_j \in I$, $(u,j) \models \psi$ und $(u,j') \models \phi$ für alle j' mit $j < j' \leq i$ gelten. Es steht also $\phi\,\mathcal{U}_I\,\psi$ für die Aussage: „es gilt ϕ, bis irgendwann in der Zukunft ψ gilt, und der Abstand dieses Zeitpunktes zur Gegenwart liegt in I", während $\phi\,\mathcal{S}_I\,\psi$ ausdrückt: „es gibt einen Zeitpunkt in der Vergangenheit, zu dem ψ galt, seitdem ϕ gilt und dessen Abstand zur Gegenwart in das Intervall I fällt."

Insbesondere bezieht sich die Intervalleinschränkung in einer Formel der Gestalt $(\neg \mathtt{tt})\mathcal{U}_I \mathtt{tt}$ immer auf die Länge des aktuellen Ereignisses.

Wenn singuläre Intervalle als zeitliche Einschränkungen genutzt werden können, dann erhält man Logiken mit großer Ausdrucksstärke:

5.4.3 Satz ([AH90]) *Die* MTL*-Theorie der Intervallwörter ist unentscheidbar.*

Demgegenüber steht die Entscheidbarkeit des Erfüllbarkeitsproblems für MITL und MITL$^{\mathrm{p}}$ (siehe [Alu91] und [AH92]), aus dem auch die Entscheidbarkeit der entsprechenden Theorien folgt.

Die entscheidende Eigenschaft nichtsingulärer Intervalle, die in den Beweisen von Alur und Henzinger die tragende Rolle spielt, ist die folgende:

5.4.4 Bemerkung Ist $I \in \mathcal{K}_0$, $r \in \mathbf{P}_0$ und sind e und f zwei weitere reelle Zahlen mit den folgenden Eigenschaften: $e \leq r \leq f$ und $f - e < 1$, dann gilt $e \in I$ oder $f \in I$.

Diese Eigenschaft erlaubt eine effektive Einbettung von MITL$^{\mathrm{p}}$ in $\exists\mathrm{MSO}\Pi^{\overleftrightarrow{\mathrm{d}}}$, aus der dann ebenfalls die Entscheidbarkeit des Erfüllbarkeitsproblems folgt. Diese Einbettung kann allerdings für eine noch größere und ausdrucksstärkere Formelklasse durchgeführt werden, womit das Ergebnis aus [AH92] verbessert wird. Beides, die Spracherweiterung und die Einbettung, wird im folgenden Abschnitt vorgestellt.

5.4.2 Erweiterung um Automatenoperatoren

Um MITL$^{\mathrm{p}}$ zu EMITL$^{\mathrm{p}}$ (‚extended metric interval temporal logic') zu erweitern, werden neue Regeln für Automatenoperatoren hinzugenommen, die aus den herkömmlichen Automatenoperatoren durch Hinzufügen einer Intervalleinschränkung entstehen:

$$\mathfrak{A}_I^+(\phi_0, \ldots, \phi_{n-1}) \mid \mathfrak{A}_I^-(\phi_0, \ldots, \phi_{n-1}), \tag{5.8}$$

wobei $\mathfrak{A}$ wie in (5.5) für einen diskreten Rabin-Scott-Automaten über dem Alphabet 2^n und I für ein Element aus $\mathcal{K}_0$ steht.

Die Semantik der neuen Formeln ergibt sich in natürlicher Weise: Ist $\mathfrak{I} = (u, i)$ eine Interpretation mit $u = (\underline{a}, \underline{w})$ und $\phi \triangleq \mathfrak{A}_I^+(\phi_0, \ldots, \phi_{n-1})$, so gilt $\mathfrak{I} \models \phi$, falls es eine Position j mit $j \geq i$ gibt, so daß $[u, \underline{\phi}](i, j+1)$ zu der von $\mathfrak{A}$ erkannten Sprache gehört und $W_{j+1} - W_i \in I$ gilt. Entsprechendes gilt für Vergangenheitsformeln.

Bevor wir uns dem zentralen Satz in diesem Abschnitt zuwenden, wollen wir ein Beispiel für eine EMITL$^{\mathrm{p}}$-Formel näher betrachten.

5.4.5 Beispiel Wir wollen eine Formel konstruieren, die folgendes aussagt: „Auf jedes zweite Ereignis des Typs a folgt ein Ereignis des Typs b innerhalb der nächsten

fünf Zeiteinheiten." Wegen der Zählbedingung „jedes zweite Ereignis" kann keine Formel aus MITL oder MITL$^{\mathrm{p}}$ diese Bedingung ausdrücken. Gerade die Zählbedingung kann aber durch einen entsprechenden Automatenoperator erfaßt werden.

Wir benutzen die folgenden Abkürzungen:

$$\begin{aligned} \mathsf{amAnfang} &\triangleq \neg\mathfrak{A}^{-}_{[0,\infty)}(\mathtt{tt})\,, \\ \Diamond_{[0,5]}\phi &\triangleq \mathtt{tt}\,\mathcal{U}_{[0,5]}\phi\,, \\ \Box\phi &\triangleq \neg(\mathtt{tt}\,\mathcal{U}_{[0,\infty)}\neg\phi)\,. \end{aligned}$$

Dabei soll $\mathfrak{A}$ die Sprache über dem Alphabet 2 erkennen, die alle Wörter der Länge zwei enthält. Dann ist die Formel $\mathsf{amAnfang}$ nur an der ersten Stelle wahr. Die Formel $\Diamond_{[0,5]}\phi$ besagt, daß ϕ in den nächsten fünf Zeiteinheiten wahr werden wird, und $\Box\phi$ steht dafür, daß ϕ immer wahr ist.

Außerdem benötigen wir einen Automaten $\mathfrak{B}$, der die Sprache M, die durch $M = \left(\binom{1}{0}\binom{0}{0}^*\binom{1}{0}\binom{0}{0}^*\right)^*\binom{1}{0}\binom{0}{0}^*\left(\binom{1}{1}+\binom{1}{0}\binom{0}{0}^*\binom{0}{1}\right)$ gegeben ist, erkennt. Eine endliche Folge von zweizeiligen Vektoren über $\{0,1\}$ gehört zu M, wenn in der zweiten Komponente nur am Schluß eine Eins steht und in der ersten Komponente geradzahlig viele Einsen auftreten. Dann beschreibt $\mathfrak{B}^{-}_{[0,\infty)}(\mathrm{p}_a, \mathsf{amAnfang})$ jedes zweite Ereignis des Typs a.

Insgesamt kann die besagte Eigenschaft dann wie folgt ausgedrückt werden:

$$\Box\left(\mathfrak{B}^{-}_{[0,\infty)}(\mathrm{p}_a, \mathsf{amAnfang}) \to \Diamond_{[0,5]}\mathrm{p}_b\right)\,.$$

Eine weitere Ausdrucksschwäche der Logik MITL$^{\mathrm{p}}$ liegt darin, daß in ihr keine Bedingungen über die Dauer einer festen Anzahl von aufeinander folgenden Ereignissen formulierbar sind. Z. B. kann man nicht sagen: „die nächsten fünf Ereignisse haben eine Gesamtdauer von höchstens drei Zeiteinheiten." Solche Sachverhalte sind aber offensichtlich unter Benutzung von Automatenoperatoren in EMITL$^{\mathrm{p}}$ beschreibbar.

Wir wollen nun zeigen:

5.4.6 Satz *Jede* EMITL$^{\mathrm{p}}$*-Formel läßt sich effektiv in eine äquivalente* $\exists\mathrm{MSO}\Pi'^{\overleftrightarrow{\mathrm{d}}}$*-Formel mit Mengentermen überführen.*

Beweis. Wir konstruieren induktiv zu jeder EMITL$^{\mathrm{p}}$-Formel ϕ eine erweitert-existentielle (!) $\Pi'^{\overleftrightarrow{\mathrm{d}}}$-Formel $\phi^* \triangleq \phi^*(X)$ mit Mengentermen, so daß für jedes Intervallwort u gilt:

1) Es gibt eine Variablenbelegung ν, so daß $(u,\nu) \models \phi^*$ gilt.

2) Ist ν eine Belegung mit $(u,\nu) \models \phi^*$, so gilt für jedes $i < |u|$:

$$(u,i) \models \phi \quad \text{gdw.} \quad i \in \nu(X)\,.$$

Dann ist $\exists X(\phi^* \wedge X0)$ die gesuchte, zu ϕ äquivalente Formel. (Beachte, daß die Menge der erweitert-existentiellen Formeln unter (semantischer) Konjunktion, Disjunktion und existentiell-monadischer Quantifizierung abgeschlossen ist.)

Der Induktionsanfang ist trivial, denn für kann einfach die Formel $\forall x\, Xx$ und für $\mathrm{p}_a^* \triangleq \mathrm{p}_a^*(X)$ kann $\forall x(Xx \leftrightarrow \mathrm{P}_a x)$ genommen werden.

Der Induktionsschritt braucht wegen 5.3.1 (das analog auch für Formeln mit Intervallbeschränkungen gilt) nur für Automatenformeln, Konjunktionen und Negationen durchgeführt zu werden. Die beiden letzten Fälle sind leicht abzuhandeln:

Sind für EMITL$^{\mathrm{P}}$-Formeln ϕ und ψ die Formeln $\phi^* \triangleq \phi^*(Y)$ und $\psi^* \triangleq \psi^*(Z)$ schon konstruiert, so setzen wir

$$(\phi \wedge \psi)^*(X) \quad \triangleq \quad \exists Y \exists Z(\forall x(Xx \leftrightarrow Yx \wedge Zx) \wedge \phi^* \wedge \psi^*)\,.$$

Ist $\phi^* \triangleq \phi^*(Y)$ schon konstruiert, so wählt man für $(\neg\phi)^*$ einfach

$$(\neg\phi)^*(X) \quad \triangleq \quad \exists Y(\forall x(Xx \leftrightarrow \neg Yx) \wedge \phi^*)\,.$$

Der Rest des Beweises behandelt den komplizierten Fall der Automatenformeln. Aus Symmetriegründen beschränken wir uns darauf, die Konstruktion für eine Formel

$$\phi \quad \triangleq \quad \mathfrak{A}_I^+(\phi_0, \ldots, \phi_{n-1})$$

vorzustellen. (Beachte, daß uns Vergangenheits- wie Zukunftsformeln zur Verfügung stehen.) Die geringfügige Asymmetrie, die bei unendlichen Wörtern dadurch entsteht, daß sie kein rechtes Ende haben, erfordert eine kleine Zusatzüberlegung. Der Übersichtlichkeit halber nehmen wir zuerst an, daß die Formel nur in unbeschränkten Intervallwörtern interpretiert wird. Die Modifikationen, die für beschränkte unendliche und endliche Intervallwörter notwendig sind, werden am Schluß des Beweises beschrieben.

Nach der Induktionsvoraussetzung sind die Formeln $\phi_0^*(X_0), \ldots, \phi_{n-1}^*(X_{n-1})$ schon konstruiert. Wir nehmen an, daß $\mathfrak{A} = (Q, s, \delta, F)$ ein deterministischer Rabin-Scott-Automat ist, für den o. E. $Q = \{0, \ldots, p-1\}$ gilt.

Die Überlegungen zur Konstruktion von ϕ^* gliedern sich in zwei Teile. Im ersten Teil wird eine semantische Bedingung herausgearbeitet, die die Gültigkeit der Formel ϕ an einer Stelle beschreibt, und zwar in Abhängigkeit von der Art des Intervalls I. Im zweiten Teil wird dann begründet, weshalb die gefundene Bedingung in eine Formel der gewünschten Art umsetzbar ist.

Die angesprochene Unterscheidung der Intervalle ist wie folgt: Die Menge $\mathcal{K}_1$ enthalte alle beschränkten Intervalle mit linker Grenze größer als Null, $\mathcal{K}_2$ enthalte alle unbeschränkten Intervalle mit linker Intervallgrenze größer als Null und $\mathcal{K}_3$ enthalte alle Intervalle mit linker Intervallgrenze Null und rechter Intervallgrenze Eins.

Jedes Intervall aus $\mathcal{K}_0$ fällt dann in eine der Mengen $\mathcal{K}_1$, $\mathcal{K}_2$ oder $\mathcal{K}_3$, oder es läßt sich als Vereinigung eines Intervalls aus $\mathcal{K}_3$ mit einem Intervall aus $\mathcal{K}_1$ oder $\mathcal{K}_2$

schreiben. Damit reicht es also, für I die Intervalle aus $\mathcal{K}_1$, $\mathcal{K}_2$ oder $\mathcal{K}_3$ in Betracht zu ziehen.

Sei $u = (\underline{a}, \underline{w})$ ein unbeschränktes Intervallwort.

Das *Raster* von u ist eine streng monoton steigende Folge $r_0, r_1, \ldots$ von Positionen, die wie folgt induktiv definiert ist:

1) Es ist $r_0 = 0$.

2) Ist die Menge $\{j \mid j > r_i \wedge W_j - W_{r_i} < 1\}$ nicht leer, so sei r_{i+1} ihr Maximum, das wegen der Annahme, daß u ein unbeschränktes Intervallwort ist, existiert.

3) Gilt $W_{r_i+1} - W_{r_i} \geq 1$, so sei $r_{i+1} = r_i + 1$.

Die Elemente von $\underline{r}$ werden *Rasterpunkte* genannt. Sie mögen die Menge R bilden.

Behauptung 1 Für jeden Index i und jede von Null verschiedene natürliche Zahl k gilt $W_{r_{i+2k}} - W_{r_i} > k$.

Beweis. Die Definition von $\underline{r}$ sichert die Behauptung für $k = 1$. Daraus folgt die gesamte Behauptung per Induktion. □

Sei M eine Menge von Positionen. Dazu definieren wir zwei partielle Funktionen $e{:}\mathbf{N} \to M$ und $f{:}\mathbf{N} \to M$: Falls für ein i die Menge $\{k \mid r_i \leq k \leq r_{i+1}\} \cap M$ nicht leer ist, so sei $e(i)$ ihr kleinstes und $f(i)$ ihr größtes Element. Andernfalls seien beide Funktionen an der Stelle i undefiniert.

Behauptung 2 Es sei I ein Intervall aus $\mathcal{K}_1$ mit oberer Schranke m. Weiterhin sei i eine Position mit $r_{k-1} \leq i < r_k$.

Dann sind die beiden folgenden Aussagen äquivalent:

1) Es gibt eine Position j in M mit $W_j - W_i \in I$.

2) Es gibt ein $l < 2m$, so daß $e(k+l)$ und $f(k+l)$ definiert sind und $W_i - W_{e(k+l)} \in I$ oder $W_{f(k+l)} - W_i \in I$ gilt.

Beweis. Aus 2) folgt trivialerweise 1). Für den Beweis der umgekehrten Implikation nehmen wir an, daß j eine Position aus M ist, für die $W_j - W_i \in I$ gilt. Dann gibt es nach Behauptung 1 ein l mit $r_{k+l} \leq j \leq r_{k+l+1}$ und $l < 2m$. Dann sind $e(k+l)$ und $f(k+l)$ definiert und es gilt $e(k+l) \leq j \leq f(k+l)$. Sei $e' = e(k+l)$ und $f' = f(k+l)$. Ist $f' - e' \leq 1$, so folgt $j = e'$ oder $j = f'$, womit die Behauptung in diesem Fall bewiesen ist. In dem anderen Fall, wenn $f' - e' \geq 2$ ist, muß aufgrund der Konstruktion des Rasters $W_{f'} - W_{e'} < 1$ gelten, weshalb nach 5.4.4 eine der Beziehungen $W_{f'} - W_i \in I$ oder $W_i - W_{e'} \in I$ gilt. □

Sei nun $v = [u, \underline{\phi}]$. Sei weiterhin für jedes $q \in Q$ der Automat $\mathfrak{A}_q$ durch $\mathfrak{A}_q = (Q, q, \delta, F)$ definiert und I ein Intervall aus einer der Mengen $\mathcal{K}_1$, $\mathcal{K}_2$ oder $\mathcal{K}_3$. Im

ersten Fall sei zusätzlich m eine obere Schranke für I und im zweiten Fall die untere Grenze kleiner oder gleich m.

Für jeden Zustand q werden zwei Mengen, E_q und F_q, definiert: Es gehöre eine Stelle i mit $r_k \leq i \leq r_{k+1}$ zu E_q, wenn das Wort $v(r_k, i)$ von $\mathfrak{A}_q$ akzeptiert wird, aber keines der Wörter $v(r_k, j)$ mit $r_k \leq j < i$. Entsprechend gehöre i zu F_q, falls $v(r_k, i)$ von $\mathfrak{A}_q$ akzeptiert wird, aber keines der Wörter $v(r_k, j)$ mit $i < j \leq r_{k+1}$.

Nun werden die Mengen E_q und F_q jeweils durch eine endliche Anzahl von Mengen überdeckt. Für jede Zahl l mit $l \leq 2m$ sei

$$E_q^l \;=\; E_q \cap \bigcup_{j \in \mathbf{N}} \{k \mid r_{(2m+1)j+l} \leq k \leq r_{(2m+1)j+l+1}\}\,.$$

Es gilt also

$$E_q \;=\; \bigcup_{l \leq 2m} E_q^l\,. \tag{5.9}$$

In gleicher Weise sei für jedes l mit $l \leq 2m$ die Menge F_q^l definiert.

Behauptung 3 Ist I in $\mathcal{K}_1$, so sind für eine Position i mit $r_{k-1} \leq i < r_k$ die beiden folgenden Aussagen äquivalent:

1) Es gilt $(u, i) \models \phi$.
2) Es gibt ein l mit $l \leq 2m$ und ein q mit $q \in Q$ und eine Stelle j, so daß gilt:

$$\begin{aligned} q \;&=\; \delta(s, v(i, r_{k+l}))\,, \\ j \;&=\; \mathrm{nf}(i, E_q^l) \;\text{ oder }\; j \;=\; \mathrm{nf}(i, F_q^l)\,, \\ r_{k+l} \;&\leq\; j \;\leq\; r_{k+l+1}\,, \\ W_j - W_i \;&\in\; I\,. \end{aligned}$$

Beweis. Trivialerweise gilt die Implikation von 2) nach 1). Die umgekehrte Implikation folgt aus Behauptung 2 und (5.9) und der entsprechenden Gleichung für F_q. □

Ist I unbeschränkt, so kommen außer den unter 2) betrachteten Positionen auch die größeren in Frage:

Behauptung 4 Ist I in $\mathcal{K}_2$, so sind für eine Position i mit $r_{k-1} \leq i < r_k$ die beiden folgenden Aussagen äquivalent:

1) Es gilt $(u, i) \models \phi$.
2) Es gilt 2) wie in Behauptung 3 oder: Es gibt eine Position j, für die $v(i, j)$ von $\mathfrak{A}$ akzeptiert wird und die größer als r_{k+2m+1} ist. □

Für den Fall, daß I zu der Menge $\mathcal{K}_3$ gehört, ist die Situation ähnlich:

Behauptung 5 Ist I aus $\mathcal{K}_3$ ein Intervall, das Null nicht enthält, so sind für eine Position i mit $r_{k-1} \leq i < r_k$ die beiden folgenden Aussagen äquivalent:

1) Es gilt $(u, i) \models \phi$.
2) Es gilt 2) aus Behauptung 3 oder: Es gibt eine Position j mit $i < j \leq r_k$, für die $v(i, j)$ von $\mathfrak{A}$ akzeptiert wird, wobei im Falle $j + 1 = r_k$ zusätzlich $W_{j+1} - W_j \in I$ erfüllt ist. □

Damit braucht die gesuchte Formel $\phi^*(X)$ nur auszudrücken, daß x genau dann zu X gehört, wenn x in Abhängigkeit von I eine der drei Bedingungen 3.2), 4.2) bzw. 5.2) erfüllt, wobei man sich i durch x ersetzt vorstellen muß.

Wir wählen den folgenden Ansatz für ϕ^*, wobei $\underline{E}$ für die Folge $E_0^0, \ldots, E_{p-1}^{2m}$ sowie $\underline{F}$ für die Folge $F_0^0, \ldots, F_{p-1}^{2m}$ stehen soll:

$$\phi^*(X) \triangleq \exists R \exists \underline{E} \exists \underline{F} (\phi_0^* \wedge \ldots \wedge \phi_{n-1}^* \wedge \mathsf{Raster}(R) \wedge \mathsf{Hilfsmengen}(R, \underline{E}, \underline{F}) \wedge \mathsf{Bedingung}(X, R, \underline{E}, \underline{F})) .$$

Raster soll das Raster des jeweiligen Wortes definieren, Hilfsmengen soll unter Zuhilfenahme des Rasters die Mengen E_q^l und F_q^l bestimmen, und Bedingung soll mit Hilfe der in den obigen Behauptungen herausgearbeiteten Bedingungen, des Rasters und der Hilfsmengen die gesuchte Menge X zur Verfügung stellen.

Wir können die Formel Hilfsmengen konstruieren, wenn wir in der Lage sind, Bedingungen, die an Läufe des Automaten $\mathfrak{A}$ gestellt werden, zu formulieren und modulo $2m + 1$ zu zählen. Beides bereitet keine Schwierigkeiten, da wir auf die volle monadische Logik in der Signatur Π zurückgreifen dürfen (vgl. die Sätze von Büchi und Elgot, siehe 2.1.3). Die Eingabe für den Automaten wird aus den Mengenvariablen in $\underline{X}$ gebildet, denn diese liefern nach Induktionsvoraussetzung die Wahrheitswerte, die die Formeln $\phi_0, \ldots, \phi_{n-1}$ annehmen.

Die in den Behauptungen 3, 4 und 5 herausgearbeiteten Bedingungen, die die Menge X definieren, greifen auf das Raster, die Hilfsmengen E_q^l und F_q^l und auf den Automaten $\mathfrak{A}$ zurück und benutzen Abstandsmessungen. Die Hilfsmengen werden durch die entsprechende Formel Hilfsmengen zur Verfügung gestellt, der Automat kann — wie im letzten Absatz begründet wurde — unter Benutzung monadischer Formeln in der Signatur Π vollständig beschrieben werden, und die Abstandsmessungen erfolgen immer zwischen einer Position und einem ihrer Nachfolger in einer der Hilfsmengen, also reichen dafür Zukunftsformeln. Deshalb kann Bedingung durch eine erweitert-existentielle Formel realisiert werden, sofern das Raster eines Wortes definiert werden kann. Das wollen wir abschließend zeigen:

Behauptung 6 Es gibt eine existentiell-monadische $\Pi'^{\overleftrightarrow{\mathrm{d}}}$-Formel $\mathsf{Raster}(X)$ mit Mengentermen, die die folgende Eigenschaft hat: Ist u ein unbeschränktes Intervallwort, so gibt es genau eine erfüllende Belegung von X, die mit dem Raster von u übereinstimmt.

Beweis. Man erhält die geforderte Formel als Konjunktion zweier Formeln $\mathsf{fein}(X)$ und $\mathsf{grob}(X)$, von denen die erste sagt, daß das Raster nicht zu grob ist, und die zweite, daß es nicht zu fein ist, d. h.,

$$\mathsf{Raster}(X) \;\triangleq\; \mathsf{fein}(X) \wedge \mathsf{grob}(X)\,.$$

In fein soll ausgedrückt werden, daß 0 zu X gehört und mit jedem Element x auch ein größeres, das der direkte Nachfolger von x ist oder aber einen Abstand zu x hat, der kleiner als Eins ist. Die Formel grob soll ausdrücken, daß der Abstand $\mathrm{d}(x,y)$ größer als Eins ist, wenn x und y zwei aufeinanderfolgende Elemente von X sind. Dann ergibt die Konjunktion beider Formeln offensichtlich die gewünschte Aussage.

Die Formulierung von fein ist nicht weiter schwierig:

$$\mathsf{fein}(X) \;\triangleq\; X0 \wedge \forall x(Xx \to (\mathrm{d}(x,X) < 1 \vee X(x+1)))\,.$$

Die Konstruktion von grob ist schwieriger, denn die relativen Abstandsterme $\mathrm{d}(X,y)$ und $\mathrm{d}(x,X)$ liefern $\mathrm{d}(x+1,y)$ bzw. $\mathrm{d}(x,y-1)$, wenn x und y zwei aufeinanderfolgende Elemente von X sind.

Wir benutzen deshalb eine Unterscheidung danach, ob y ein Element von X mit geradzahliger Nummer oder ungeradzahliger Nummer ist, und spalten X in zwei disjunkte Mengen, die Menge der Elemente mit geradzahliger Nummer und die Menge der Elemente mit ungeradzahliger Nummer auf. Dafür benutzen wir die Formel

$$\begin{aligned}\mathsf{gerade}(X,Y,Z) \;\triangleq\; &\forall x((Xx \leftrightarrow ((Yx \wedge \neg Zx) \vee (\neg Yx \wedge Zx)))\\ &\wedge \forall y(Xx \wedge Xy \wedge x < y \wedge \neg\exists x'(x < x' < y \wedge Xx) \to (Yx \leftrightarrow \neg Zy)))\end{aligned}$$

die X in Y und Z aufspaltet. Außerdem benutzen wir $Y+1$ für $\{x \mid x > 0 \wedge x-1 \in Y\}$ und entsprechend $Z+1$ für $\{x \mid x > 0 \wedge x-1 \in Z\}$.

Die Formel grob läßt sich dann in der Form

$$\begin{aligned}\mathsf{grob}(X) \;\triangleq\; &\exists Y \exists Z(\mathsf{gerade}(X,Y,Z)\\ &\wedge \forall x((Yx \to \mathrm{d}(x,Z+1) \geq 1) \wedge (Zx \to \mathrm{d}(x,Y+1) \geq 1)))\end{aligned}$$

schreiben. □

Im Falle endlicher Wörter muß auch das maximale Element der zugehörigen Wortstruktur in das Raster aufgenommen werden. Im Falle beschränkter unendlicher Wörter muß in die Formulierung der Bedingung 2) von Behauptung 3 eine ähnliche Alternativbedingung aufgenommen werden, wie sie in Bedingung 2) von Behauptung 4 zu finden ist, damit der Tatsache Rechnung getragen wird, daß das Raster in diesem Falle endlich, die Menge der Positionen aber unendlich ist. ■

Als Folgerung erhalten wir sofort wie im Falle von TL_Γ (vgl. 5.3.3):

5.4.7 Korollar *1) Das Erfüllbarkeitsproblem für* EMITL$^{\mathrm{P}}$ *ist entscheidbar.*

2) Die EMITL$^{\mathrm{P}}$*-Theorie der Intervallwörter ist entscheidbar.*

3) Das Modellproblem ist für Uhrenautomaten und EMITL$^{\mathrm{P}}$ *entscheidbar.*

Eine genaue Analyse des Beweises von 5.4.6 zeigt, daß die zu einer EMITL$^{\mathrm{P}}$-Formel konstruierte äquivalente $\Pi'^{\overleftrightarrow{\mathrm{d}}}$-Formel mit Mengentermen eine eindeutige Formel ist. Deshalb folgt zusammen mit 5.1.13 und 5.1.14 sogar:

5.4.8 Korollar *Jede durch eine* EMITL$^{\mathrm{P}}$*-Formel definierbare Sprache von Intervallwörtern ist eindeutig uhrenerkennbar.*

5.5 Eine entscheidbare Erweiterung von $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$

In Abschnitt 4.7 haben wir gesehen, daß die natürlichen Erweiterungen der Berechnungsstruktur Σ_{UA} zu Automatenmodellen führen, die ein unentscheidbares Leerheitsproblem besitzen. Es stellt sich die Frage, ob man diese starken Modelle nicht durch geschickt gewählte Zusatzbedingungen so weit einschränken kann, daß das Leerheitsproblem wieder entscheidbar wird, die Klasse der erkannten Sprachen jedoch noch echt größer als die Klasse der uhrenerkennbaren Sprachen ist.

Eine Untersuchung in dieser Richtung ist in [ACH93] zu finden, wo eine „kleine“ Erweiterung des Modells des Uhrenautomaten vorgestellt wird, die noch ein entscheidbares Leerheitsproblem besitzt. Zu dem dort benutzten Automatenmodell korrespondiert in natürlicher Weise eine Teilklasse der Formeln in $\exists\mathrm{MSO}\Pi_{\int}$, die ausdrucksstärker als $\exists\mathrm{MSO}\Pi^{\overleftarrow{\mathrm{d}}}$ ist, aber aufgrund der Entscheidbarkeit des Leerheitsproblems für das entsprechende Automatenmodell ein entscheidbares Erfüllbarkeitsproblem besitzt. Dies soll hier gezeigt werden.

In diesem Unterabschnitt werden nur endliche Intervallwörter betrachtet. Außerdem gehen wir o. B. d. A. immer von Uhrenautomaten bzw. hybriden Automaten aus, die einen isolierten Anfangszustand und isolierte Endzustände haben, d. h., in denen keine Transition in den Anfangszustand führt und keine aus einem Endzustand herausführt. Eine *Endtransition* ist eine Transition, deren Zielzustand ein Endzustand ist. Alle anderen Transitionen heißen *innere Transitionen*. Ein Register eines hybriden Automaten wird *Uhr* genannt, wenn jede Transition die Schrittfunktion add_1 auf das Register anwendet.

In [ACH93] wird gezeigt, daß das Leerheitsproblem für Uhrenautomaten auf endlichen Intervallwörtern auch dann entscheidbar ist, wenn man sie mit einem einzigen zusätzlichen Register versieht, auf das beliebige Schrittfunktionen aus $\{\mathrm{add}_k \mid k \in \mathbf{N}\}$ angewendet werden können, das nicht zurückgesetzt wird und dessen Wert nur beim Lesen des letzten Ereignisses abgefragt werden kann. Formal bedeutet dies, daß nur in Endtransitionen Registertests benutzt werden dürfen, in denen das zusätzliche Register auftritt. Zum Beweis wird die Idee der Regionen, die wir in Abschnitt 5.1.1 kennengelernt haben, wieder aufgegriffen und ausgefeilt.

Das Resultat kann wie folgt interpretiert werden:

5.5.1 Satz *Das Erfüllbarkeitsproblem ist entscheidbar für Formeln der Gestalt*

$$\exists X_0 \ldots \exists X_{m-1}(\phi_0 \wedge \psi),$$

bei denen $\exists X_0 \ldots \exists X_{m-1}\phi_0$ eine existentiell-monadische $\Pi\overleftarrow{\mathrm{d}}$-Formel ist und für die es natürliche Zahlen $a_0, \ldots, a_{m-1}$ gibt, so daß ψ eine boolesche Kombination von relativen Integralformeln der Gestalt

$$\int_{\emptyset}^{\max} a_0X_0 + \ldots + a_{m-1}X_{m-1} \quad \sim \quad d \tag{5.10}$$

ist.

(Man beachte, daß die Koeffizienten a_i in allen atomaren Teilformeln von ψ gleich sind.)

Beweis. Wir beschreiben ein Verfahren, daß aus einem Satz der angegebenen Art einen äquivalenten Uhrenautomaten konstruiert, der die oben angeführten Forderungen erfüllt.

Es sei B das Alphabet $A \times 2^m$.

Die Formel ϕ_0 wandeln wir zuerst in eine Formel über B um, in der die Mengenvariablen in den Buchstabenprädikaten kodiert sind. Dazu ersetzen wir in ϕ_0 jede Teilformel X_ix durch $\bigvee_b \mathrm{P}_b x$, wobei b alle Tupel $(a, p_0, \ldots, p_{m-1})$ mit $p_i = 1$ durchlaufe. Entsprechend wird $\mathrm{P}_a x$ durch $\bigvee_b \mathrm{P}_b x$ ersetzt, wobei b alle Elemente aus B mit a in der ersten Komponente annehme.

Die entstehende Formel kann nach dem speziellen Charakterisierungssatz 3.7.6 in einen äquivalenten Uhrenautomaten $\mathfrak{A}$ umgewandelt werden, von dem wir o. B. d. A. annehmen können, daß er einen isolierten Anfangszustand und isolierte Endzustände hat. Es sei R die Registermenge von $\mathfrak{A}$ und

$$k_P \quad = \quad \sum_{i \in P} a_i \qquad \text{für } P \subseteq m.$$

Aus $\mathfrak{A}$ gewinnen wir den gewünschten hybriden Automaten, indem wir ein neues Register r_0 zu R hinzufügen und jede Transition

$$(q, (a, p_0, \ldots, p_{m-1}), \alpha, \zeta, q') \quad \text{durch} \quad (q, a, \theta, \alpha \wedge \beta, \zeta, q')$$

ersetzen, wobei θ durch

$$\theta(r) \quad = \quad \begin{cases} \mathrm{add}_1, & \text{falls } r \in R, \\ \mathrm{add}_{k_{\{i \mid p_i = 1\}}}, & \text{falls } r = r_0, \end{cases}$$

bestimmt sei. Der Test β sei $\mathtt{tt}$, falls q' ein innerer Zustand ist, andernfalls entstehe β aus ψ durch Ersetzen jeder Teilformel wie in (5.10) durch $r_0 \sim d$. ■

5.5.2 Satz *Die in 5.5.1 vorgestellte Logik ist echt ausdrucksstärker als* $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$.

Beweis. Wir betrachten die Sprache $L = \{w_0w_1w_2 \mid w_0 + w_2 = 1\}$ aus dem Beweis von 3.6.3. In diesem Beweis wird gezeigt, daß L nicht uhrenerkennbar ist. Aus dem speziellen Charakterisierungssatz können wir deshalb schließen, daß L nicht $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$-definierbar ist. Aber L wird durch die Formel

$$\exists X(\mathsf{max} = \mathbf{2} \wedge X0 \wedge \neg X\mathbf{1} \wedge X\mathbf{2} \wedge \int_{\emptyset}^{\mathsf{max}} X = 1)$$

definiert, die offensichtlich äquivalent in eine Formel der angegebenen Logik umgewandelt werden kann. ■

In [KPSY93] werden zwei Einschränkungen des Modells des CSHS betrachtet. In beiden wird eine Menge R_0 von Uhren ausgezeichnet, so daß in inneren Transitionen nur Register aus R_0 zurückgesetzt und abgefragt werden und gilt:

(*) Die Registertests der inneren Transitionen sind Konjunktionen von Tests der Form $r \geq d$ und $r \leq d$ mit $r \in R_0$.

Unter diesen Annahmen sprechen wir von einem CSHS *vom Typ (A)*, wenn R_0 einelementig ist. Wir sprechen von einem CSHS *vom Typ (B)*, wenn jeder Registertest einer Endtransition eine Disjunktion von Tests der Form $a_0r_0 + \ldots + a_{m-1}r_{m-1} \sim d$, also eine Disjunktion von linearen Tests ist.

Kesten, Pnueli, Sifakis und Yovine zeigen nun, daß für CSHS vom Typ (A) oder (B) das Leerheitsproblem entscheidbar ist. Im Beweis wird wesentlich daraus Nutzen gezogen, daß in den Registertests der inneren Transitionen nur schwache Vergleiche ($\leq$ bzw. $\geq$) auftreten und Negationen nicht erlaubt sind (vgl. (*)). Insofern können die beschriebenen Automaten schon einfache zeitliche Abhängigkeiten nicht ausdrücken. Z. B. ist die Sprache aller endlichen Intervallwörter $\underline{w}$ mit $w_i < 1$ für alle $i < |\underline{w}|$ durch kein CSHS des Typs (A) oder (B) beschreibbar (ohne Beweis).

Beide Typen erkennen allerdings die Sprache aus 3.6.3, die nicht uhrenerkennbar ist, aber durch einen Uhrenautomaten mit Haltemöglichkeit erkannt wird. Damit sind die Sprachklassen, die durch CSHS des Typs (A) bzw. (B) bestimmt sind, mit der Klasse der uhrenerkennbaren Sprachen unvergleichbar.

Einer ansprechenden Deutung aus logischer Sicht, wie wir sie für das Ergebnis aus [ACH93] erzielt haben, entziehen sich die Resultate aus [KPSY93], weil in Registertests von inneren Transitionen Negationen verboten sind (siehe (*)).

5.6 Beschränkte Variation und die volle Abstandslogik MSOΠd

Sei k eine natürliche Zahl größer als Null.

In Abschnitt 4.7 haben wir gesehen, daß die durch Uhrenautomaten relativ zu V^k erkennbaren Sprachen ein streng effektives boolesches System bilden, das zusätzlich effektiv unter Projektion abgeschlossen ist. Diese Tatsache erlaubt eine elegante Schlußfolgerung, die die in Abschnitt 5.2 vorgestellte volle Abstandslogik MSOΠd betrifft.

5.6.1 Lemma *Zu jedem Satz ϕ aus* MSOΠd *läßt sich effektiv ein Uhrenautomat $\mathfrak{A}$ konstruieren, für den* $\mathrm{L}(\mathfrak{A}) \cap V^k = \mathrm{Mod}(\phi) \cap V^k$ *gilt.*

Beweis. Da MSOΠd auch die Allquantifizierung von Mengenvariablen erlaubt, kann man auf Elementvariablen weitgehend verzichten, indem man jede Elementvariable durch eine Mengenvariable ersetzt und dafür sorgt, daß diese nur durch einelementige Mengen interpretiert wird. (Siehe auch den Beweis in [Tho90a] für den Satz, daß jede S1S-Formel in einen äquivalenten Büchi-Automaten umgewandelt werden kann.) Deshalb können wir o. B. d. A. davon ausgehen, daß ϕ aus Formeln der Gestalt

$$\exists x(Xx \wedge \neg\exists y(\neg x = y \wedge Xy)) \tag{5.11}$$

$$\exists x(Xx \wedge \mathrm{P}_a x) \tag{5.12}$$

$$\exists x(Xx \wedge \mathrm{Y}x) \tag{5.13}$$

$$\exists x\exists y(Xx \wedge Yy \wedge x < y) \tag{5.14}$$

$$\exists x\exists y(Xx \wedge Yy \wedge \mathrm{d}(x,y) \sim c) \tag{5.15}$$

durch Anwendung boolescher Operationen und Quantifizierung von Mengenvariablen entsteht. Induktiv über den Formelaufbau konstruieren wir nun zu jeder solchen Formel $\phi \triangleq \phi(X_0, \ldots, X_{m-1})$ einen Uhrenautomaten $\mathfrak{A}$ über dem Alphabet $B = A \times 2^m$, so daß für ein Intervallwort $u = (\underline{a}, \underline{w})$ aus V^k und eine Belegung ν die beiden folgenden Aussagen äquivalent sind:

1) Es gilt $(u^{\Pi\mathrm{d}}, \nu) \models \phi$.

2) Der Automat $\mathfrak{A}$ erkennt das Wort $(\underline{b}, \underline{w})$, wobei für jedes $i < |u|$ der Buchstabe $b_i = (a_i, p_0, \ldots, p_{m-1})$ wie folgt bestimmt ist:

$$p_j = 1 \quad \text{gdw. } i \in \nu(X_j) \qquad \text{für } j < m.$$

Für den Induktionsanfang müssen (5.11) bis (5.15) behandelt werden. Wir führen die Konstruktion nur für (5.15), den einzig schwierigen Fall, an einem Beispiel vor. Es läßt sich leicht auf den allgemeinen Fall erweitern.

In Abbildung 5.5 findet man einen Automaten für

$$\phi(X_0, X_1, X_2) \quad \triangleq \quad \exists x\exists y(X_0 x \wedge X_1 y \wedge \mathrm{d}(x,y) = 5)\,.$$

Im Induktionsschritt müssen die booleschen Kombinationen und die Mengenquantifizierungen behandelt werden. Für erstere reicht es, darauf hinzuweisen, daß die

Klasse der relativ zu V^k uhrenerkennbaren Sprachen ein effektives boolesches System bildet. Für letzteres reicht die Bemerkung, daß die Klasse der relativ zu V^k uhrenerkennbaren Sprachen effektiv unter Projektion abgeschlossen ist. (Beides ist in 4.7.7 zu finden.) ■

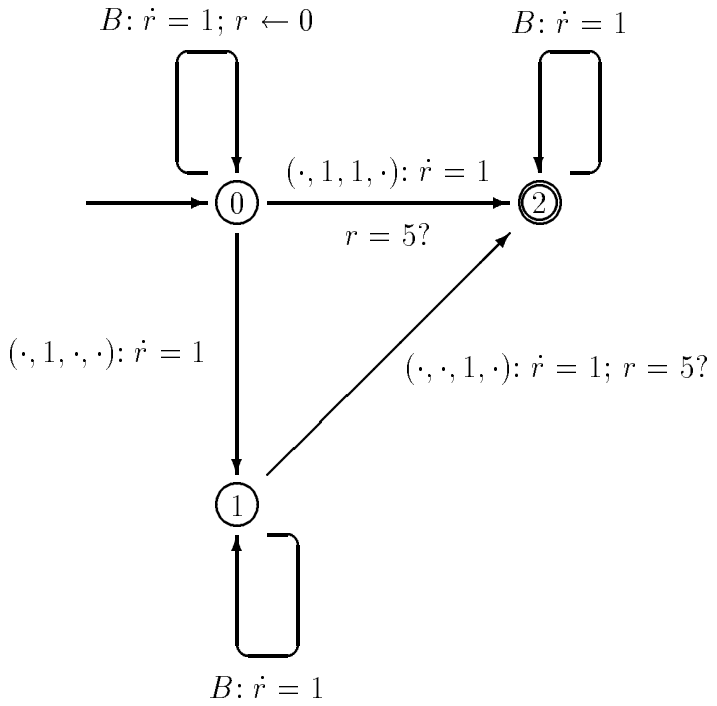


Abbildung 5.5

Hier steht $\cdot$ für einen beliebigen Buchstaben aus A bzw. für ein beliebiges Element aus $\{0, 1\}$.

Als Folgerung erhalten wir:

5.6.2 Korollar *Die* MSOΠd*-Theorie der endlichen, unendlichen bzw. unbeschränkten Intervallwörter einer beschränkten Variation (aus V^k) ist entscheidbar.*

Resümee und Ausblick

Die vorliegende Arbeit vermittelt die Einsicht, daß (existentiell-)monadische Logiken zweiter Stufe zur Beschreibung zeitlicher Zusammenhänge gewinnbringend eingesetzt werden können. Zuerst konnten wir zeigen, daß sie in natürlicher Weise das Verhalten von Automaten charakterisieren, die zur Modellierung zeitabhängiger Systeme benutzt werden. Über die Verbindung zu den Automaten gelang es uns dann, eine ausdrucksstarke, zur Formulierung zeitlicher Zusammenhänge geeignete und darüberhinaus mit einem entscheidbaren Erfüllbarkeitsproblem versehene Logik zu definieren: die schwache Abstandslogik $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$. Wir konnten uns davon überzeugen, daß diese Logik in gewissem Sinne eine „maximale“ Logik mit den aufgeführten Eigenschaften ist, denn für natürliche Erweiterungen haben wir jeweils die Unentscheidbarkeit des Erfüllbarkeitsproblems nachgewiesen. Die große Ausdrucksstärke von $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ erlaubt die Einschätzung zeitlogischer Spezifikationsformalismen: Wir konnten zeigen, daß die Logiken TL_Γ sowie $\mathrm{EMITL}^{\mathrm{p}}$ (eine echte Erweiterung der schon bekannten Logik $\mathrm{MITL}^{\mathrm{p}}$) eine entscheidbare Theorie besitzen. Stukturelle Eigenschaften der schwachen Abstandslogik $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$, der durch die in dieser Arbeit erzielten Ergebnisse eine grundlegende Rolle zukommt, konnten wir wieder durch Rückgriff auf bekannte oder neu gewonnene Erkenntnisse aus dem Gebiet der Automatentheorie ermitteln.

Dem Einsatz der beiden ausdrucksstärksten Logiken mit entscheidbarem Erfüllbarkeitsproblem, die in der vorliegenden Arbeit behandelt wurden, nämlich $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ und $\mathrm{EMITL}^{\mathrm{p}}$, sind in der Praxis aus Effizienzgründen Grenzen gesetzt: Das Leerheitsproblem der Uhrenautomaten ist PSPACE-vollständig (siehe [AD90]), womit das Erfüllbarkeitsproblem für $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ PSPACE-schwierig ist, während das Erfüllbarkeitsproblem für MITL schon EXPSPACE-vollständig ist (siehe [AFH91]), woraus folgt, daß das Erfüllbarkeitsproblem für $\mathrm{EMITL}^{\mathrm{p}}$ sogar EXPSPACE-schwierig ist.

Die Suche nach effizient handhabbaren Formalismen zur Beschreibung zeitabhängiger Systeme könnte mit zweien, schon in der vorliegenden Abhandlung untersuchten Logiken beginnen: mit TL_Γ und mit der vollen Abstandslogik MSOIId bei Einschränkung auf Intervallwörter beschränkter Variation. Diese Einschränkung ist aus praktischer Sicht nicht unrealistisch: Geht man von der Annahme aus, daß es in einem gegebenen System für jede atomare Aktion eine untere zeitliche Schranke gibt, dann hat das gesamte System, sofern es aus endlich vielen atomaren Aktionen aufgebaut

ist, die Eigenschaft, daß es eine obere Schranke für die Anzahl der in einem Einheitsintervall möglichen Ereignisse gibt, d. h., die Menge der Intervallwörter, die durch das System definiert wird, ist eine Menge beschränkter Variation. Es gilt dagegen offensichtlich nicht — man denke hier nur an sich in ihrem Ablauf überschneidende Aktionen —, daß es für den Abstand zweier Ereignisse eine untere zeitliche Schranke gibt.

Einerseits verursacht die uneingeschränkte Benutzung von monadischen Existenzquantoren die Ineffizienz von $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$, andererseits verdankt $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ den monadischen Existenzquantoren ihre Ausdrucksstärke und ohne Mengenvariablen können in ihr keine zeitlichen Zusammenhänge ausgedrückt werden. Ein sinnvoller Ausgleich zwischen beiden Extremen — keine Mengenvariablen oder existentielle Mengenquantifizierung — könnte in solchen Logiken gelingen, die sich Mengenkonstruktoren bedienen. Denn diese liefern im allgemeinen unmittelbar den Abschluß unter Komplementbildung und bringen einen gewissen Grad von Determinismus mit sich, wie dies z. B. aus der Komplexitätstheorie bekannt ist: Dort studiert man die Klassen P und NP der deterministisch bzw. nichtdeterministisch in Polynomzeit akzeptierbaren Sprachen. Nach allgemeiner Ansicht enthält P ausschließlich effizient lösbare Probleme, während es in NP auch Probleme gibt, die sich einer effizienten Lösung entziehen. Die Klasse NP kann durch eine entsprechende existentielle Logik zweiter Stufe beschrieben werden, während P durch eine Logik charakterisiert werden kann, die sich des LFP-Operators (‚least fixpoint operator') bedient, der eine Konstruktion von Relationen erlaubt (siehe [Imm87a] und [Imm87b]).

Hinsichtlich einer „maximalen" Logik mit entscheidbarem Erfüllbarkeitsproblem, die im Gegensatz zu $\exists\mathrm{MSO}\Pi\overleftarrow{\mathrm{d}}$ auch unter Negation abgeschlossen ist, haben wir Untersuchungen auf dem Gebiet der Uhrenautomaten angestellt und das Modell des eindeutigen Uhrenautomaten, das zwischen dem deterministischen und nichtdeterministischen Modell liegt, studiert. In diesem Zusammenhang wurde die wichtige Frage aufgeworfen, ob die Klasse der eindeutig uhrenerkennbaren Sprachen gegen (effektive) Komplementbildung abgeschlossen ist.

Ein anderer Fragenkomplex betrifft die Verallgemeinerung der Ergebnisse der vorliegenden Arbeit auf den Fall eines verzweigten Zeitmodells nach dem Vorbild des Satzes von Rabin über die Entscheidbarkeit der monadischen Theorie zweiter Stufe des unendlich verzweigten binären Baumes. Erstrebenswert sind Resultate über entsprechende monadische Logiken, die eine Interpretation der schon bekannten Entscheidbarkeitsergebnisse für verzweigte Zeitlogiken, das Modellproblem für einfache zeitabhängige Systeme betreffend (siehe [ADC91] und [HNSY92]), und die Charakterisierung des Verhaltens noch zu definierender „zeitabhängiger Baumautomaten" erlauben.

Literaturverzeichnis

[ACD90] Rajeev Alur, Costas Courcoubetis und David Dill. Model-checking for real-time systems. In „Fifth Annual IEEE Symposium on Logic in Computer Science“, Seiten 414–425, Philadelphia, Pennsylvania (1990). IEEE Computer Society Press.

[ACD91] Rajeev Alur, Costas Courcoubetis und David Dill. Model-checking for probabilistic real-time systems. In J. Leach Albert, B. Monien und M. Rodríguez Artalejo (Herausgeber), „Automata, Languages and Programming: 18th International Colloquium“, Band 510 aus „Lecture Notes in Computer Science“, Seiten 115–126, Madrid, Spanien (1991). European Assoc. Theoret. Comput. Sci., Springer-Verlag.

[ACD+92] Rajeev Alur, Costas Courcoubetis, David Dill, N. Halbwachs und H. Wong-Toi. Minimization of timed transition systems. In W. R. Cleaveland (Herausgeber), „Concur '92: Third International Conference on Concurrency Theory“, Band 630 aus „Lecture Notes in Computer Science“, Seiten 340–354, Stony Brook, New York (August 1992). Springer-Verlag.

[ACD93] Rajeev Alur, Costas Courcoubetis und David Dill. Model-checking in dense real-time. *Inform. and Control* **104**(1), 1–34 (1993).

[ACH93] Rajeev Alur, Costas Courcoubetis und Thomas A. Henzinger. Computing accumulated delays in real-time systems. In Costas Courcoubetis (Herausgeber), „Computer Aided Verification: 5th International Conference“, Band 697 aus „Lecture Notes in Computer Science“, Seiten 181–193, Elounda, Griechenland (1993). Springer-Verlag.

[ACHH93] Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger und Pei-Hsin Ho. Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In Robert L. Grossman, Anil Nerode, Anders P. Ravn und Hans Rischel (Herausgeber), „Hybrid Systems“, Band 736 aus „Lecture Notes in Computer Science“. Springer-Verlag, Berlin (1993).

[AD] Rajeev Alur und David Dill. A theory of timed automata. To appear in Theoret. Comput. Sci.

[AD90] Rajeev Alur und David Dill. Automata for modeling real-time systems. In M. S. Paterson (Herausgeber), „Automata, Languages and Programming: 17th International Colloquium“, Band 443 aus „Lecture Notes in Computer Science“, Seiten 322–335, Warwick, England (1990). European Assoc. Theoret. Comput. Sci., Springer-Verlag.

[AD91] Rajeev Alur und David Dill. The theory of timed automata. In J. W. de Bakker, C. Huizing, W.P. de Roever und G. Rozenberg (Herausgeber), „Real-Time: Theory in Practice“, Band 600 aus „Lecture Notes in Computer Science“, Seiten 45–73, Mook, Niederlande (1991). Springer-Verlag.

[ADC91] Rajeev Alur, David Dill und Costas Courcoubetis. Verifying automata specifications of probabilistic real-time systems. In J. W. de Bakker, C. Huizing, W.P. de Roever und G. Rozenberg (Herausgeber), „Real-Time: Theory in Practice“, Band 600 aus „Lecture Notes in Computer Science“, Seiten 28–44, Mook, Niederlande (1991). Springer-Verlag.

[AFH91] Rajeev Alur, Tomás Feder und Thomas A. Henzinger. The benefits of relaxing punctuality. In „Proceedings of the 10th ACM Symposium on Principles of Distributed Computing“, Seiten 139–152, Montreal, Quebec, Kanada (1991). ACM Press.

[AH89a] Rajeev Alur und Thomas Henzinger. A really temporal logic. In „Proceedings of the 30th Annual Symposium on Foundations of Computer Science“, Seiten 164–169. IEEE Computer Society Press (1989).

[AH89b] Rajeev Alur und Thomas A. Henzinger. A really temporal logic. Bericht STAN-CS-89-1267, Department of Computer Science, Stanford University (Juli 1989).

[AH90] Rajeev Alur und Thomas Henzinger. Real-time logics: complexity and expressiveness. In „Fifth Annual IEEE Symposium on Logic in Computer Science“, Seiten 390–401, Philadelphia, Pennsylvania (1990). IEEE Computer Society Press.

[AH91a] Rajeev Alur und Thomas Henzinger. Time for logic. *SIGACT News* **22**(3), 6–12 (1991).

[AH91b] Rajeev Alur und Thomas A. Henzinger. Logics and models of real-time: a survey. In J. W. de Bakker, C. Huizing, W.P. de Roever und G. Rozenberg (Herausgeber), „Real-Time: Theory in Practice“, Band 600 aus „Lecture Notes in Computer Science“, Seiten 74–106, Mook, Niederlande (1991). Springer-Verlag.

[AH92] Rajeev Alur und Thomas A. Henzinger. Back to the future: towards a theory of timed regular languages. In „33rd Annual Symposium on Foundations of Computer Science“, Seiten 177–186, Pittsburgh, Pennsylvania (1992). IEEE Computer Society Press.

[AH93] Rajeev Alur und Thomas A. Henzinger. Real-time logics: complexity and expressiveness. *Inform. and Control* **104**(1), 35–77 (1993).

[AL91] Martín Abadi und Leslie Lamport. An old-fashioned recipe for real-time. In J. W. de Bakker, C. Huizing, W.P. de Roever und G. Rozenberg (Herausgeber), „Real-Time: Theory in Practice“, Band 600 aus „Lecture Notes in Computer Science“, Seiten 1–27, Mook, Niederlande (1991). Springer-Verlag.

[Alu91] Rajeev Alur. „Techniques for Automatic Verification of Real-time Systems“. Dissertation, Stanford University, Kalifornien (1991).

[Arn83] André Arnold. Rational ω-languages are non-ambiguous. *Theoret. Comput. Sci.* **26**, 221–223 (1983).

[Büc60] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* **6**, 66–92 (1960).

[Büc62] J. Richard Büchi. On a decision method in restricted second-order arithmetic. In E. Nagel, P. Suppes und A. Tarski (Herausgeber), „Logic, Methodology, and Philosophy of Science: Proc. of the 1960 International Congress“, Seiten 1–11. Stanford University Press (1962).

[Čer92] Kārlis Čerāns. Decidability of bisimulation equivalences for parallel timer processes. In G. v. Bochmann und D. K. Probst (Herausgeber), „Computer Aided Verification, Fourth International Workshop, CAV '92“, Band 663 aus „Lecture Notes in Computer Science“, Seiten 302–315, Montreal, Kanada (1992). Springer-Verlag.

[CES86] E. M. Clarke, E. A. Emerson und A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic. *ACM Transactions on Programming Languages and Systems* **8**(2), 244–263 (April 1986).

[CHR91] Zhou Chaochen, C.A.R. Hoare und A.P. Ravn. A calculus of durations. *Information Processing Letters* **40**(5), 269–276 (1991).

[Don70] John Doner. Tree acceptors and some of their applications. *J. Comput. System Sci.* **4**(5), 406–451 (Oktober 1970).

[EH86] Allen A. Emerson und Joe Y. Halpern. "Sometimes" and "not never" revisited: On branching time versus linear time. *J. Assoc. Comput. Mach.* **33**, 151–178 (1986).

[EH93] Joost Engelfriet und Hendrik Jan Hoogeboom. X-automata on ω-words. *Theoret. Comput. Sci.* **110**(1), 1–51 (1993).

[Elg61] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Amer. Math. Soc.* **98**, 21–51 (Januar 1961).

[EM93] Werner Ebinger und Anca Muscholl. Logical definability on infinite traces. In Andrzej Lingas, Rolf Karlsson und Svante Carlsson (Herausgeber), „Automata, Languages and Programming: 20th International Colloquium", Band 700 aus „Lecture Notes in Computer Science", Seiten 335–346, Lund, Schweden (1993). European Assoc. Theoret. Comput. Sci., Springer-Verlag.

[GNRR93] Robert L. Grossman, Anil Nerode, Anders P. Ravn und Hans Rischel (Herausgeber). „Hybrid Systems", Band 736 aus „Lecture Notes in Computer Science". Springer-Verlag, Berlin (1993).

[GR92] Dora Giammaresi und Antonio Restivo. Recognizable picture languages. *Internat. J. Pattern Recognition and Artificial Intelligence* **6**(2,3), 241–256 (1992).

[GRST94] Dora Giammaresi, Antonio Restivo, Sebastian Seibert und Wolfgang Thomas. Monadic second-order logic over rectangular pictures and recognizability by tiling systems. In P. Enjalbert und E. W. Mayr (Herausgeber), „STACS 94: 11th Annual Symposium on Theoretical Aspects of Computer Science", Band 775 aus „Lecture Notes in Computer Science", Seiten 365–375, Caen, Frankreich (1994). Springer-Verlag.

[Har87] David Harel. Statecharts: a visual formalism for complex systems. *Sci. Comput. Programming* **8**, 231–274 (1987).

[Har88] Eyal Harel. Temporal analysis of real-time systems. Diplomarbeit, The Weizmann Institute of Science, Rehovot, Israel (1988).

[HC91] Michael R. Hansen und Zhou Chaochen. Semantics and completeness of duration calculus. In J. W. de Bakker, C. Huizing, W. P. de Roever und G. Rozenberg (Herausgeber), „Real-Time: Theory in Practice“, Band 600 aus „Lecture Notes in Computer Science“, Seiten 176–194, Mook, Niederlande (1991). Springer-Verlag.

[HLP91] Eyal Harel, Orna Lichtenstein und Amir Pnueli. Explicit clock temporal logic. In „Fifth Annual IEEE Symposium on Logic in Computer Science“, Seiten 402–413, Philadelphia, Pennsylvania (1991). IEEE Computer Society Press.

[HMP91] Thomas A. Henzinger, Zohar Manna und Amir Pnueli. Temporal proof methodologies for real-time systems. In „Conference Record of the 18th Annual ACM Symposium on Principles of Programming Languages“, Seiten 353–366, Orlando, Florida (1991). ACM Press.

[HNSY92] Thomas A. Henzinger, Xavier Nicollin, Joseph Sifakis und Sergio Yovine. Symbolic model checking for real-time systems. In „Seventh Annual IEEE Symposium on Locic in Computer Science“, Seiten 394–406, Santa Cruz, Kalifornien (1992). IEEE Computer Society Press.

[Ho93] Pei-Hsin Ho. Automatic symbolic verification of hybrid systems (August 1993). Vortrag über eine gemeinsame Arbeit mit Rajeev Alur und Thomas A. Henzinger anläßlich der Sommerschule „Summerschool in Logical Methods in Concurrency“, Comput. Sci. Dept., Aarhus University, Dänemark.

[Hod93] Wilfrid Hodges. „Model Theory“, Band 42 aus „Encyclopedia of Mathematics and its Applications“. Cambridge University Press, Cambridge (1993).

[HU79] John E. Hopcroft und Jeffrey D. Ullman. „Introduction to Automata Theory, Languages and Computation“. Addison-Wesley (1979).

[Imm87a] Neil Immerman. Expressibility as a complexity measure: results and directions. In „2nd Conference on Structure in Complexity Theory“, Seiten 194–202 (1987).

[Imm87b] Neil Immerman. Languages which capture complexity classes. *SIAM J. Comput.* **16**, 760–778 (1987).

[JM86] Farnam Jahanian und Aloysius Ka-Lau Mok. Safety analysis of timing properties in real-time systems. *IEEE Transactions on Software Engineering* **SE-12**(9), 890–904 (September 1986).

[Kam68] JOHAN ANTHONY WILLEM KAMP. „Tense Logic and the Theory of Linear Order“. Dissertation, University of California (1968).

[KdR86] RON KOYMANS UND WILLEM PAUL DE ROEVER. Examples of real-time temporal logic specification. In B. T. DENVIR, W. T. HARWOOD, M. I. JACKSON UND M. J. WRAY (Herausgeber), „The Analysis of Concurrent Systems“, Band 207 aus „Lecture Notes in Computer Science“, Seiten 231–251. Springer-Verlag (1986).

[Koy89] RON KOYMANS. „Specifying Message Passing and Time-critical Systems with Temporal Logic“. Dissertation, Eindhoven Univ. of Techn. (1989).

[Koy90] RON KOYMANS. Specifying real-time properties with metric temporal logic. *Real-time Systems* **2**(4), 255–299 (1990).

[KPSY93] Y. KESTEN, AMIR PNUELI, JOSEPH SIFAKIS UND SERGIO YOVINE. Integration graphs: a class of decidable hybrid systems. In ROBERT L. GROSSMAN, ANIL NERODE, ANDERS P. RAVN UND HANS RISCHEL (Herausgeber), „Hybrid Systems“, Band 736 aus „Lecture Notes in Computer Science“, Seiten 179–208. Springer-Verlag, Berlin (1993).

[Kro93] HENNING KROPF. Vergleich zweier Automatenmodelle für Spezifikationen mit kontinuierlicher Zeit. Diplomarbeit, Christian-Albrechts-Universität zu Kiel, Kiel (1993).

[KVdR83] RON KOYMANS, JAN VYTOPIL UND WILLEM PAUL DE ROEVER. Real-time programming and asynchronous message passing. In „Proceedings of the Second Annual ACM Symposium on Principles of Distributed Computing“, Seiten 187–197. ACM Press (1983).

[Lam93] LESLIE LAMPORT. Hybrid systems in TLA$^+$. In ROBERT L. GROSSMAN, ANIL NERODE, ANDERS P. RAVN UND HANS RISCHEL (Herausgeber), „Hybrid Systems“, Band 736 aus „Lecture Notes in Computer Science“, Seiten 77–102, Berlin (1993). Springer-Verlag.

[Lew90] HARRY R. LEWIS. A logic of concrete time intervals. In „Fifth Annual IEEE Symposium on Logic in Computer Science“, Seiten 380–389, Philadelphia, Pennsylvania (1990). IEEE Computer Society Press.

[McN66] ROBERT MCNAUGHTON. Testing and generating infinite sequences by a finite automaton. *Information and Control* **9**, 521–530 (1966).

[MP93] ZOHAR MANNA UND AMIR PNUELI. Models for reactivity. *Acta Informatica* **30**(2), 609–678 (Oktober 1993).

[NSY91] Xavier Nicollin, Joseph Sifakis und Sergio Yovine. From ATP to timed graphs and hybrid systems. In J. W. de Bakker, C. Huizing, W. P. de Roever und G. Rozenberg (Herausgeber), „Real-Time: Theory in Practice“, Band 600 aus „Lecture Notes in Computer Science“, Seiten 549–572, Mook, Niederlande (1991). Springer-Verlag.

[NSY93] Xavier Nicollin, Joseph Sifakis und Sergio Yovine. From ATP to timed graphs and hybrid systems. *Acta Informatica* **30**(2), 181–202 (1993).

[Ost87] Jonathan S. Ostroff. „Temporal Logic of Real-time Systems“. Dissertation, University of Toronto (1987).

[Ost90] Jonathan S. Ostroff. „Temporal logic of real-time systems“. Research Studies Press, Taunton, Somerset, England (1990).

[PK92] Amir Pnueli und Y. Kesten. Timed and hybrid statecharts and their textual representation. In Jan Vytopil (Herausgeber), „Formal Techniques in Real-time and Fault-tolerant Systems: Second International Symposium, Nijmegen, The Netherlands“, Band 571 aus „Lecture Notes in Computer Science“, Seiten 591–620, Nijmegen, Niederlande (Januar 1992). Springer-Verlag.

[Pnu77] Amir Pnueli. The temporal logic of programs. In „18th Annual Symposium on Foundations of Computer Science“, Seiten 46–57, Rhode Island, Providence (1977). IEEE Computer Society Press.

[Rab69] Michael Ozer Rabin. Decidability of second-order theories and finite automata on infinite trees. *Trans. Amer. Math. Soc.* **141**, 1–35 (1969).

[RRH93] A. P. Ravn, H. Rischel und K. M. Hansen. Specifying and verifying requirements of real-time systems. *IEEE Transactions on Software Engineering* **SE-19**(1), 41–55 (Januar 1993).

[Saf88] Shmuel Safra. On the complexity of ω-automata. In „29th Annual Symposium on Foundations of Computer Science“, Seiten 319–327, White Plains, New York (1988). IEEE Computer Society Press.

[Sco67] Dana Scott. Some definitional suggestions for automata theory. *J. Comput. System Sci.* **1**, 187–212 (1967).

[She75] Saharon Shelah. The monadic theory of order. *Ann. of Math.* **102**, 379–419 (1975).

[Tho82] Wolfgang Thomas. Classifying regular events in symbolic logic. *J. Comput. System Sci.* **25**, 360–376 (1982).

[Tho90a] Wolfgang Thomas. Automata on infinite objects. In Jan van Leeuwen (Herausgeber), „Handbook of Theoretical Computer Science", Band B: Formal Methods and Semantics, Seiten 134–191. Elsevier Science Publishers B. V. (1990).

[Tho90b] Wolfgang Thomas. On logical definability of regular trace languages. In Volker Diekert (Herausgeber), „Proceedings of a Workshop of the ESPRIT Basic Research Action No. 3166: Algebraic and Syntactic Methods in Computer Science (ASMICS)", Seiten 172–182 (1990). Bericht TUM-I9002, Technische Universität München.

[TM51] Alfred Tarski und J. C. C. McKinsey. „A decision method for elementary algebra and geometry". University of California Press, Berkeley, 2. Auflage (1951).

[TW68] J. W. Thatcher und J. B. Wright. Generalized finite automata theory with an application to a decision problem of second-order arithmetic. *Math. Systems Theory* **2**(1), 57–81 (1968).

[Wol83] Pierre Wolper. Temporal logic can be more expressive. *Inform. and Control* **56**, 72–99 (1983).

[YL93] Mihalis Yannakakis und David Lee. An efficient algorithm for minimizing real-time transition systems. In Costas Courcoubetis (Herausgeber), „Computer Aided Verification: 5th International Conference", Band 697 aus „Lecture Notes in Computer Science", Seiten 210–224, Elounda, Griechenland (1993). Springer-Verlag.

[Zho93] Ping Zhou. „Clocks, Communication, and Correctness". Dissertation, Eindhoven Univ. of Tech. (1993).