Kapitel 2

Die Ausdrucksstärke der Logiken *FO+MTC*, *FO+MLFP* und *MSO*

In diesem Kapitel untersuchen wir die Ausdrucksstärke der im vorherigen Kapitel eingeführten Logiken, die aus der Logik erster Stufe durch Hinzunahme des monadischen TC-Operators ($FO+MTC$), des monadischen LFP-Operators ($FO+MLFP$) sowie des monadischen Mengenquantors (MSO) entstehen.

Wir haben bereits in Lemma 1.4.10 gezeigt, daß ausschließlich reguläre Wort- und Baumsprachen durch Formeln in $FO + MTC$ bzw. $FO + MLFP$ definierbar sind. In [53] wurde gezeigt, daß jede reguläre Wortsprache durch eine Formel der Form $\exists X\ \varphi(X)$ definierbar ist, wobei $\varphi(X)$ keine weiteren Mengenquantoren enthält. Dieses Ergebnis werden wir auf reguläre Baumsprachen und den monadischen LFP-Operator erweitern; jede reguläre Wort- und Baumsprache ist also schon durch eine Formel mit einem Mengenquantor bzw. einem monadischen LFP-Operator definierbar. Für die Logik erster Stufe mit monadischem TC-Operator auf Wörtern werden wir zeigen, daß höchstens zwei geschachtelte MTC-Operatoren notwendig sind, um alle regulären Wortsprachen zu beschreiben. Es ist nicht bekannt, ob diese Aussage verschärft werden kann, d.h. ob es überhaupt eine reguläre Wortsprache gibt, die nicht schon durch eine Formel in $FO + MTC$ ohne geschachtelte MTC-Operatoren beschrieben werden kann.

Verallgemeinert-reguläre Ausdrücke können leicht induktiv in Formeln der Logik erster Stufe mit monadischem TC-Operator übersetzt werden. Jeder *-Operator wird dabei durch einen monadischen TC-Operator ersetzt. Daher hängt die Frage, ob geschachtelte MTC-Operatoren notwendig sind, um alle regulären Wortsprachen zu definieren, sehr eng mit dem bekannten Problem der verallgemeinerten Sternhöhe regulärer Wortsprachen zusammen. Die verallgemeinerte Sternhöhe einer regulären Wortsprache L ist die minimale Anzahl geschachtelter Anwendungen des *-Operators in irgendeinem verallgemeinert-regulären Ausdruck, der L beschreibt. Sternfreie Wortsprachen haben also insbesondere die verallgemeinerte Sternhöhe 0. Seit langer Zeit *ist es ein offenes Problem, ob es reguläre Wortsprachen mit einer verallgemeinerten Sternhöhe größer als 1 gibt.*

Im Fall der regulären Baumsprachen ist unbekannt, ob überhaupt alle regulären Baumsprachen in *FO+MTC* definierbar sind. Wir werden dieses Problem am Ende von Kapitel 4 wieder aufgreifen.

Wir zeigen die angekündigten Ergebnisse zunächst für den Bereich der Wortmodelle.

Satz 2.0.1 *Jede reguläre Wortsprache L wird beschrieben durch*

(a) eine Formel in FO+MTC mit höchstens zwei geschachtelten MTC-Operatoren,

(b) eine Formel in FO+MLFP mit höchstens einem MLFP-Operator,

(c) eine Formel in MSO mit höchstens einem existentiellen Mengenquantor.

Beweis Die Wortsprache L werde erkannt durch einen DEA der Form $\mathcal{A} = (\{1,\ldots,n\},\Sigma,\delta,1,F)$. Sei $w = w_0\ldots w_r$ ein Wort in Σ^*. Wir bezeichnen mit q_i den Zustand $\delta(1,w_0\ldots w_i)$ für $1 \leq i \leq r$. Wir werden zeigen, daß akzeptierende Läufe von $\mathcal{A}$ in geeignet kodierter Form durch Formeln der geforderten Art kodiert werden können. Dabei wird nicht der gesamte Lauf kodiert, sondern nur die Zustände, die an speziellen Stellen des Wortes angenommen werden. Zunächst geben wir einen Überblick. Sei $m = 3n$. Wir kodieren die Zustände an allen Stellen, die Vielfache von m sind, indem für die Stelle lm die Stelle $lm + q_{lm}$ ausgezeichnet wird. Bei der Beschreibung durch einen Mengenquantor bzw. durch den MLFP-Operator werden neben den Stellen $lm + q_{lm}$ auch die Stellen, die Vielfache von m sind, in die konstruierte Menge aufgenommen. Die Wahl von m erlaubt es dann, aus dem Abstand zweier ausgezeichneter Stellen zu erkennen, welche Stelle ein Vielfaches von m kodiert und welche Stelle einen Zustand kodiert. Haben zwei Stellen in einer Menge einen Abstand von höchstens n, so handelt es sich bei der kleineren Stelle um ein Vielfaches von m und bei der größeren Stelle um die Kodierung des Zustandes an dieser Stelle. Ist der Abstand größer als n, aber kleiner als m, so ist die größere Stelle ein Vielfaches von m. Bei der Beschreibung akzeptierender Läufe in der Logik $FO + MTC$ wird der in einer Stelle x kodierte Zustand dadurch dekodiert, indem der Abstand von x zum nächst kleineren Vielfachen von m bestimmt wird. Für die Bestimmung dieser Stelle wird dann ein weiterer MTC-Operator benötigt. Dieser kann auch ersetzt werden durch einen modulo m zählenden Quantor bzw. durch ein numerisches Prädikat $C_{0,m}$, wie es in [26] eingeführt wurde. Dieses Prädikat trifft genau dann auf eine Stelle k zu, wenn k durch m teilbar ist.

Die Kodierungen werden durch die folgende Tabelle beschrieben. Sei w ein Wort der Länge $lm + k$ mit $m \leq k < 2m$. Eine Zahl i in der ersten Zeile unter dem Wort

w bezeichnet die Stelle, die bei der Anwendung des äußeren MTC-Operators nach i Schritten erreicht wird. In der zweiten Zeile sind diejenigen Stellen mit i markiert, die bei der i-ten Iteration des MLFP-Operators neu aufgenommen werden.

w	w_0	w_1	$\cdots$	w_m	$\cdots$	w_{m+q_m}	$\cdots$	w_{2m}	$\cdots$	w_{lm}	$\cdots$	$w_{lm+q_{lm}}$	$\cdots$	w_{lm+k}
$FO + MTC$		0				1						l		$l+1$
$FO + MLFP$	0	0		1		2		3		$2l-1$		$2l$		$2l+1$

Jede endliche Menge von Worten kann durch eine 1.Stufe-Formel beschrieben werden. Daher beschränken wir uns darauf, alle Worte in L mit einer Länge größer oder gleich $2m$ zu definieren. Wir verwenden bei der Angabe der Formeln die folgenden Abkürzungen:

$x \leq y$	:	$x < y \vee x = y$
xS^iy	:	$x = y \vee \exists x_1 \ldots \exists x_{i-1} \, (xSx_1 \wedge \ldots \wedge x_{i-1}Sy)$
$x = 0$	:	$\forall y \; x \leq y$
$x = i$	:	$\exists y \; (y = 0 \wedge yS^ix)$
$x = max$	:	$\forall y \; x \geq y$
xS^Xy	:	$Xx \wedge Xy \wedge x < y \wedge \forall z \, (Xz \rightarrow (z \leq x \vee y \leq z))$
Xi	:	$\exists x \; (x = i \wedge Xx)$

Die Abkürzung xS^iy beschreibt, daß x und y genau den Abstand i haben. Die Formel xS^Xy trifft genau dann auf zwei Stellen zu, wenn zwischen den beiden Stellen keine weitere Stelle zur Menge X gehört. Zuletzt verwenden wir für Zustände $p, q \in \{1, \ldots, n\}$ und $i \in \mathbb{N}$ noch die Formeln $\psi^i_{p,q}(x)$ mit der Bedeutung:

$$(w, k) \models \psi^i_{p,q}(x) \iff \delta(p, w_k \ldots w_{k+i-1}) = q.$$

(w, k) erfüllt also $\psi^i_{p,q}(x)$, wenn an der Stelle k eine Buchstabenfolge der Länge i beginnt, die Zustand p in Zustand q überführt. Da die Anzahl dieser Buchstabenfolgen beschränkt ist, besteht $\psi^i_{p,q}(x)$ also aus einer endlichen Fallunterscheidung über die Beschriftung der Knoten x bis $x + i - 1$ und kann daher durch eine 1.Stufe-Formel beschrieben werden.

Beweis von (a):

Der innere MTC-Operator der zu konstruierenden Formel beschreibt alle Stellen x, die ein Vielfaches von m sind. Dies wird durch die Formel

$$\varphi_1(x) : \exists z \; [z = 0 \wedge TC(y, y', yS^my')(z, x)]$$

erreicht. Der äußere MTC-Operator wird verwendet, um den kodierten Lauf zu beschreiben. Zusammen mit dem inneren MTC-Operator kann dann die Stelle festgelegt werden, an der der kodierte Zustand angenommen wird. Bei den Übergängen des äußeren MTC-Operators sind zwei Fälle zu unterscheiden. Entweder erfolgt der

Übergang von Stelle $im + q_{im}$ zur Stelle $(i+1)m + q_{(i+1)m}$, falls die Länge von w größer oder gleich $(i+2)m$ ist, oder von Stelle $im + q_{im}$ zum Ende des Wortes, falls die Länge von w echt kleiner $(i+2)m$ ist und der Rest des Wortes ab Stelle im mit dem kodierten Zustand q_{im} akzeptiert wird. Der erste Übergang wird durch die Formel $\varphi_{Schritt}(x_1, x_2, x_3)$ beschrieben.

$$\varphi_{Schritt}(x_1, x_2, x_3) : \exists x_4\, [x_1 S^{2m} x_4 \wedge \bigvee_{j=1,\ldots,n} \bigvee_{k=1,\ldots,n} (x_1 S^j x_2 \wedge x_1 S^{m+k} x_3 \wedge \psi^m_{j,k}(x_1))]$$

Es gilt also $(w, k_1, k_2, k_3) \models \varphi_{Schritt}(x_1, x_2, x_3)$ genau dann, wenn folgende Bedingungen zutreffen: k_1 ist wenigstens $2m$ vom Ende des Wortes entfernt, und wenn k_2 die Kodierung des Zustandes an der Stelle k_1 ist, dann ist k_3 die Kodierung des Zustandes an der Stelle $k_1 + m$.

Entspechend wird der zweite Übergang durch die Formel $\varphi_{Max}(x_1, x_2, x_3)$ beschrieben.

$$\varphi_{Max}(x_1, x_2, x_3) : x_3 = max \wedge \bigvee_{i=m,\ldots,2m-1} \bigvee_{j=1,\ldots,n} \bigvee_{k \in F} (x_1 S^i x_3 \wedge x_1 S^j x_2 \wedge \psi^{i+1}_{j,k}(x_1))$$

Es gilt $(w, k_1, k_2, k_3) \models \varphi_{Max}(x_1, x_2, x_3)$ genau dann, wenn k_3 das Ende des Wortes ist, k_1 weniger als $2m$ vom Ende des Wortes entfernt ist und der Rest des Wort ab Stelle k_1 vom Zustand $k_2 - k_1$ akzeptiert wird.

Zusammen erhalten wir die folgende Formel in *FO+MTC* mit zwei geschachtelten MTC-Operatoren für die Menge der Wörter in L einer Länge größer oder gleich $2m$:

$$\exists x \exists x'\, [x = 1 \wedge x' = max \wedge TC(z, z', \varphi(z, z'))(x, x')] \quad \text{mit}$$

$$\varphi(z, z') : \exists z'' [\varphi_1(z'') \wedge (\varphi_{Schritt}(z'', z, z') \vee \varphi_{Max}(z'', z, z'))]$$

Beweis von (b):

Wie oben beschrieben, unterscheiden wir für je zwei Stellen k und k' mit Abstand kleiner als m im Fixpunkt, ob der Abstand kleiner oder gleich n ist, oder größer als n ist. Im ersten Fall beschreibt k' die Kodierung des Zustandes an der Stelle k und entweder wird $k + m$ in den Fixpunkt aufgenommen oder das Ende des Wortes wird aufnommen, falls das Wort eine Länge kleiner als $k + 2m$ hat und ab Stelle k vom Zustand $(k' - k)$ akzeptiert wird. Dies wird durch die folgende Formel $\varphi_{klein}(x_1, x_2, x_3)$ ausgedrückt:

$$\varphi_{klein}(x_1, x_2, x_3) : (\bigvee_{i=1,\ldots,n} x_1 S^i x_2 \wedge x_1 S^m x_3) \vee \varphi_{Max}(x_1, x_2, x_3)$$

Im anderen Fall haben k und k' einen Abstand größer als n. Dann ist k' ein Vielfaches von m und k kodiert den Zustand an der Stelle $k' - m$. In diesem Fall kann die Stelle in den Fixpunkt aufgenommen werden, die den Zustand an der Stelle k' kodiert. Die folgende Formel $\varphi_{groß}(x_1, x_2, x_3)$ beschreibt gerade diesen Übergang.

$$\varphi_{groß}(x_1, x_2, x_3) : (\bigvee_{i=2n,\ldots,m-1} x_1 S^i x_2) \wedge \exists x_4\, [x_4 S^m x_2 \wedge \varphi_{Schritt}(x_4, x_1, x_3)]$$

Zusammen erhalten wir, daß die Menge der Wörter einer Länge größer oder gleich $2m$ durch die Formel $\exists x\, [x = max \wedge LFP(z, X, \varphi(z, X))(x)]$ mit

$$\varphi(z, X) : z = 0 \vee z = 1 \vee \exists y \exists y'\, [Xy \wedge Xy' \wedge (\varphi_{klein}(y, y', z) \vee \varphi_{groß}(y, y', z))]$$

definiert wird.

Beweis von (c):

Die Idee dieser Kodierung ist die gleiche, wie sie bei der MLFP-Formel verwendet wurde. Nur die Eigenschaft, daß die beschriebene Menge die kleinste Menge ist, die gewisse Abschlußeigenschaften besitzt, muß auf andere Weise sichergestellt werden. Dafür darf nun die Mengenvariable X auch negativ vorkommen. Dies führt zu der folgenden Formel:

$$\exists X\, [X0 \wedge X1 \wedge Xmax \wedge \varphi_{abschluß}(X)]$$

mit

$$\varphi_{abschluß}(X) : \forall x \forall y \forall z [(x S^X y \wedge y S^X z) \rightarrow (\varphi_{klein}(x, y, z) \vee \varphi_{groß}(x, y, z))]$$

Die Formel $\varphi_{abschluß}(X)$ drückt aus, daß je drei direkt aufeinanderfolgende Stellen in X einen Übergang analog zur Iteration des MLFP-Operators beschreiben.

Wir hätten auf dieses Ergebnis auch sofort mit Lemma 1.4.10 schließen können. Die Formel hätte in diesem Fall einen universellen Mengenquantor enthalten. □

Wir wollen nun zeigen, wie sich der vorherige Satz für die Logiken $FO + MLFP$ und MSO auf den Fall der Baumsprachen übertragen läßt. Da sich nur in einem Punkt eine Modifikation ergibt, gehen wir auf diesen Punkt genauer ein und verzichten darauf, die Kodierung in allen Einzelheiten in eine Formel zu übertragen.

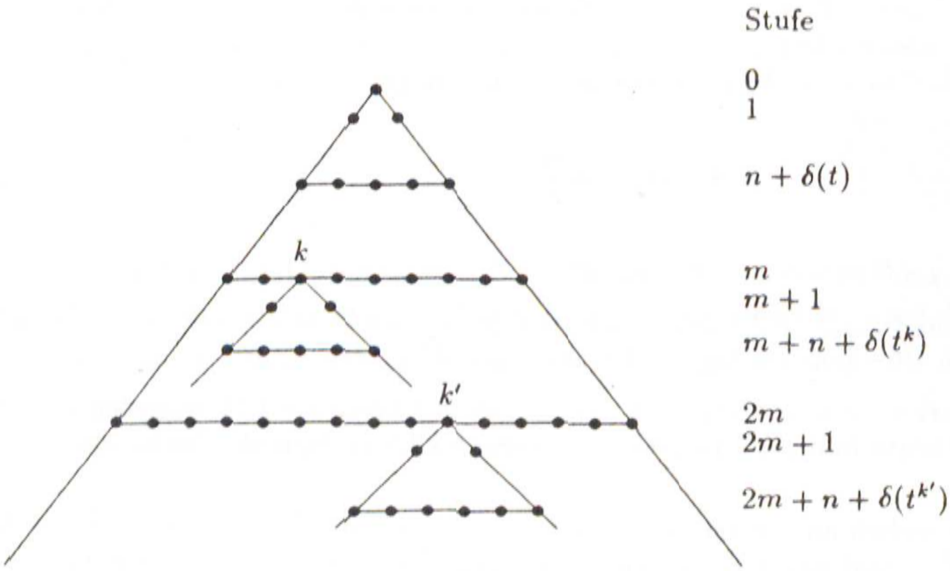
Satz 2.0.2 *Jede reguläre Baumsprache $T \subseteq T_\Sigma$ wird beschrieben durch*

(a) eine Formel in FO+MLFP mit höchstens einem MLFP-Operator,

(b) eine Formel in MSO mit höchstens einem existentiellen Mengenquantor.

Beweis Sei $\Sigma = \Sigma_0 \cup \ldots \cup \Sigma_r$ und sei $\mathcal{A} = (\{1, \ldots, n\}, \Sigma, \delta, F)$ ein T erkennender DBA. Zuerst teilen wir einen Baum in Stufen auf. Stufe i besteht aus allen Knoten, die Abstand i zur Wurzel haben, d.h. aus allen Knoten k mit $|k| = i$. Sei nun wieder

$m = 3n$. Wir wollen wieder die Zustände kodieren, die an den Knoten angenommen werden, die sich auf den Stufen im befinden. Die Kodierung des Zustandes $\delta(t^k)$ an einem Knoten k erfolgt durch Auszeichnung aller Knoten, die unter k auf der Stufe $|k|+n+\delta(t^k)$ liegen. Da ein DBA von den Blättern zur Wurzel arbeitet, die Einteilung in Stufen aber in umgekehrter Reihenfolge definiert ist, kann der vorherige Beweis für den Fall der Wortsprachen nicht vollständig übernommen werden. Wir unterscheiden nun zwei Phasen der Berechnung des Fixpunktes. In der ersten Phase werden alle Knoten in den Fixpunkt aufgenommen, die auf den Stufen im und $im+1$ liegen und für die ein Nachfolger auf der Stufe $(i+1)m$ existiert. Indem jeweils die Knoten auf zwei aufeinanderfolgenden Stufen ausgezeichnet werden, kann leicht zwischen Knoten unterschieden werden, die einen Zustand kodieren bzw. die eine Stufe im markieren. In der zweiten Phase werden dann die Knoten aufgenommen, die einen Zustand kodieren. Das folgende Bild veranschaulicht die Idee dieses Beweises.



Da die beschriebenen Übergänge in beiden Phasen lokaler Natur sind, d.h. immer nur Knoten in einem Subbaum der Höhe $2m$ berücksichtigen, ist offensichtlich, daß diese Übergänge durch eine 1.Stufe-Formel beschrieben werden können. Wir wollen daher auf die einfache, aber notationell aufwendige Konstruktion einer Fixpunktformel verzichten. Die existentielle Formel der monadischen Logik zweiter Stufe ergibt sich dann wieder wie im vorherigen Satz oder mit Lemma 1.4.10, angewendet auf die Fixpunktformel für das Komplement von T und anschließender Negation. □

Für eine Diskussion der Ausdrucksstärke der Logik erster Stufe mit monadischem TC-Operator verweisen wir auf die Abschnitte 4.4 und 5.4.

Kapitel 3

Eine Charakterisierung der sternfreien Baumsprachen

Die Klasse der sternfreien Baumsprachen wurde von Thomas [55] als direkte Verallgemeinerung der Klasse der sternfreien Wortsprachen eingeführt. Ein Hauptergebnis dieser einführenden Arbeit war die Beschreibung der sternfreien Baumsprachen durch ein Fragment der monadischen Logik zweiter Stufe, der sogenannten Antikettenlogik (vgl. Definition 1.4.12).

Satz 3.0.1 *[55] Eine Baumsprache ist genau dann sternfrei, wenn sie in der Antikettenlogik definierbar ist.*

Vergleicht man die Klassen der sternfreien Baumsprachen mit den aperiodischen und 1.Stufe-definierbaren Baumsprachen, so zeigt sich, daß die Ergebnisse von Schützenberger und McNaughton für Wortsprachen nicht vollständig auf Baumsprachen übertragbar sind. So folgt aus dem obigen Satz, daß jede in der Logik erster Stufe definierbare Baumsprache sternfrei ist, andererseits wurde in der oben zitierten Arbeit bereits gezeigt, daß nicht jede sternfreie Baumsprache aperiodisch ist, und damit auch nicht in der Logik erster Stufe definierbar ist. Von Heuter [18, 19] und Niwiński [34] wurde dieses Ergebnis in zwei Richtungen verstärkt. So zeigte Heuter, daß es sogar aperiodische sternfreie Baumsprachen gibt, die nicht in der Logik erster Stufe definierbar sind, und Niwiński bewies, daß jede Baumsprache (über Alphabeten ohne unäre Symbole), die in Logik erster Stufe mit modulo-zählenden Quantoren definierbär ist, auch sternfrei ist. Wir werden sehen, daß die Einschränkung auf Alphabete ohne unäre Symbole notwendig ist. Tatsächlich hängt es ausschließlich von den unären Subbäumen einer Baumsprache ab, ob diese sternfrei ist oder nicht. Wir werden eine Bedingung an die unären Subbäume einer Baumsprache stellen, die hinreichend und notwendig dafür ist, daß eine Baumsprache antikettendefinierbar und sternfrei ist. Zu jeder Baumsprache, die diese Bedingung erfüllt, geben wir sowohl eine definierende Antikettenformel als auch einen definierenden sternfreien Ausdruck an. Daraus ergibt sich insbesondere ein einfacher und konstruktiver Beweis, daß jede antikettendefinierbare Baumsprache sternfrei ist. Dies war der schwierige Teil von Satz 3.0.1. Wir zeigen als Hauptergebnis dieses Kapitels:

Satz 3.0.2 *Eine reguläre Baumsprache $T \subseteq T_\Sigma$ ist sternfrei bzw. antikettendefinierbar genau dann, wenn der unäre Teil von T aperiodisch ist (vgl. Definition 1.5.7).*

Wir teilen den Beweis in drei Abschnitte. Zuerst weisen wir nach, daß jede Baumsprache, deren unärer Teil nicht aperiodisch ist, auch nicht antikettendefinierbar ist. Für die Rückrichtung beginnen wir mit der Konstruktion einer definierenden Antikettenformel. Dazu zeigen wir zuerst, wie Knotenmengen, die nur aus mehrstelligen Knoten bestehen, durch Mengen von Blättern kodiert werden können. Dann geben wir für jede reguläre Baumsprache, deren unärer Teil aperiodisch ist, eine definierende Formel der monadischen Logik zweiter Stufe an, in der Mengenvariablen nur über Mengen mehrstelliger Knoten rangieren. Indem wir auf diese Formel die Kodierung anwenden, erhalten wir eine definierende Antikettenformel der Form $\exists^{ACh} Y_1 \ldots \exists^{ACh} Y_n \; \varphi(Y_1, \ldots, Y_n)$, wobei die Formel $\varphi(Y_1, \ldots, Y_n)$ keine weiteren Mengenquantoren enthält und nur durch paarweise disjunkte Mengen von Blättern erfüllt werden kann. Im letzten Abschnitt übersetzen wir diese Formel dann in zwei Schritten in einen sternfreien Ausdruck. Zuerst werden mögliche Belegungen der Antiketten durch Verkettungsvariablen kodiert. Dadurch erhalten wir eine in der Logik erster Stufe definierbare Baumsprache, die dann durch sternfreie Ausdrücke beschrieben wird.

3.1 Die Notwendigkeit der Bedingung

Sei also T eine Baumsprache, deren unärer Teil nicht aperiodisch ist. Dann gibt es einen unären speziellen Baum $s \in S_\Sigma$ mit $s^n \not\approx_T s^{n+1}$ für alle $n \in \mathbb{N}$. Nach Definition der syntaktischen Kongruenz gibt es $t \in S_\Sigma$ und $t' \in T_\Sigma$ mit $t \cdot s^n \cdot t' \in T$ für unendlich viele $n \in \mathbb{N}$ und $t \cdot s^n \cdot t' \notin T$ für unendlich viele $n \in \mathbb{N}$. Sei U die Menge der unären speziellen Bäume. Offensichtlich ist die Menge aller Bäume der Form $t \cdot U \cdot t'$ 1.Stufe-definierbar. Wir betrachten nun die Baumsprache $T' = T \cap t \cdot U \cdot t'$. Da die antikettendefinierbaren Baumsprachen unter Booleschen Operationen abgeschlossen sind, ist T genau dann antikettendefinierbar, wenn auch T' antikettendefinierbar ist. Nach Wahl von t und t' ist T' eine periodische Baumsprache und daher nicht 1.Stufe-definierbar. Jede Antikette in einem Baum in $t \cdot U \cdot t'$ besteht aus höchstens $N = |front(t)| + |front(t')| - 1$ vielen Knoten. Daher kann in einer Antikettenformel, die eine Teilmenge von $t \cdot U \cdot t'$ definiert, ein Antikettenquantor durch eine Folge von N-vielen 1.Stufe-Quantoren ersetzt werden. Dabei müssen zwei Fälle unterschieden werden, denn eine Antikettenvariable Y rangiert sowohl über nicht-leere Mengen als auch über die leere Menge. Für den zweiten Fall werden in einer Formel alle atomaren Formeln Yy durch eine nicht zu erfüllende Formel ersetzt. Für den ersten Fall ersetzen wir einen Antikettenquantor $\exists Y$ durch die Folge $\exists y_1 \dots \exists y_N \bigwedge_{i \neq j} \neg(y_i < y_j)$ von 1.Stufe-Quantoren und eine atomare Formel Yy durch die Formel $y = y_1 \vee \dots \vee y = y_N$. Eine Disjunktion über die beiden Formeln ist dann äquivalent zur ursprünglichen Formel. Da T' nicht 1.Stufe-definierbar ist, ist T' und damit auch T nicht antikettendefinierbar. Da jeder sternfreie Ausdruck leicht induktiv in eine Antikettenformel übersetzt werden kann, ist T also auch nicht sternfrei.

3.2 Die Konstruktion einer Antikettenformel

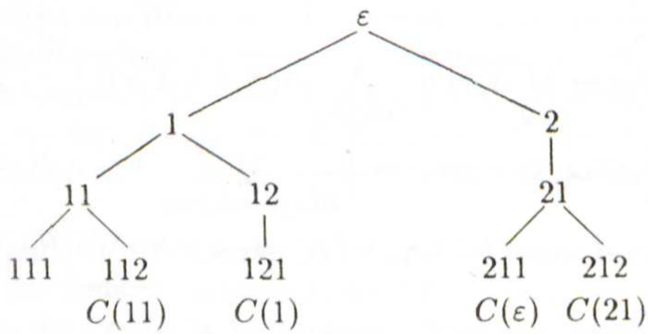
Wir führen den Beweis für die Rückrichtung von Satz 3.0.2 nur für Alphabete der Form $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, in denen also alle Symbole höchstens den Rang 2 besitzen. Die angegebenen Konstruktionen sind direkt auf beliebige Alphabete übertragbar, erfordern dann aber wesentlich aufwendigere Notationen.

Wir beginnen mit der Kodierung von Knotenmengen durch Mengen von Blättern. Wenn ein Rangalphabet unäre Symbole enthält, können nicht alle Knotenmengen eines beliebigen Baumes durch Mengen von Blättern kodiert werden. Wenn man sich aber auf Knotenmengen beschränkt, die nur 2-stellige Knoten enthalten, dann ist sichergestellt, daß mehr Blätter als 2-stellige Knoten im Baum vorhanden sind. Für unsere Konstruktion können nicht alle denkbaren Kodierungen verwendet werden, da die Kodierung zusätzlich in der Antikettenlogik bzw. im Kalkül der sternfreien Ausdrücke beschreibbar sein muß. Wir werden zeigen, daß die Abbildung

$C : \{k \in dom(t) \mid k2 \in dom(t)\} \to front(t)$ mit

$C(k) =$ das eindeutige Blatt in der Menge $k21^*$

diesen Anforderungen genügt. Das folgende Bild zeigt einen Baum mit der Kodierung der 2-stelligen Knoten.



Offensichtlich ist C injektiv auf der Menge der 2-stelligen inneren Knoten und alle Blätter, bis auf das lexikographisch kleinste Blatt, liegen im Bild von C. Wir zeigen nun zuerst, daß C 1.Stufe-definierbar ist, d.h. es gibt eine 1.Stufe-Formel $\varphi_C(x, y)$ mit $(t, k_1, k_2) \models \varphi_C(x, y) \iff C(k_1) = k_2$. Wenn k_2 die Kodierung von k_1 ist, dann ist

k_2 das am weitesten links stehende Blatt unter dem zweiten Nachfolger von k_1. Dieses Blatt kann dadurch beschrieben werden, daß jeder Knoten, der zwischen dem zweiten Nachfolger von k_1 und dem Blatt k_2 liegt, der erste Nachfolger seines Vorgängers ist. Durch Umsetzen dieser Beschreibung in eine Formel erhalten wir:

$$\varphi_C(x,y) : \exists z\,[xS_2z \ \wedge\ \neg\exists z_1\exists z_2(z \leq z_1 < z_2 \leq y \wedge z_1S_2z_2)]$$

Sei nun $T \subseteq T_\Sigma$ eine reguläre Baumsprache, deren unärer Teil aperiodisch ist, und sei $\mathcal{A} = (\{1,\ldots,n\},\Sigma,\delta,F)$ der minimale, T erkennende DBA. Nach Lemma 1.5.8 kann für alle $i,i' \in \{1,\ldots,n\}$ eine 1.Stufe-Formel $\varphi_{i,i'}(x,x')$ konstruiert werden mit

$(t,k,k') \models \varphi_{i,i'}(x,x') \quad \Longleftrightarrow$

- $k \leq k'$,
- alle Knoten l mit $k \leq l < k'$ sind unär,
- wenn $\delta(t^{k'}) = i'$, dann $\delta(t^k) = i$.

Unter Verwendung der Formeln $\varphi_{i,i'}(x,x')$ geben wir nun eine existentielle Formel der monadischen Logik zweiter Stufe an, die die akzeptierenden Läufe von $\mathcal{A}$ beschreibt. Die Mengen $X_1,\ldots,X_n$ legen in dieser Formel die Verteilung der Zustände im Lauf von $\mathcal{A}$ auf einem Baum fest, d.h. X_i enthält alle 2-stelligen Knoten k eines Baumes t mit $\delta(t^k) = i$. Bedingung (i) beschreibt, daß jeder 2-stellige Knoten in genau einer Menge vorkommt, und daß umgekehrt jede Menge nur 2-stellige Knoten enthält. Bedingung (ii) garantiert, daß die Verteilung der Zustände lokal mit der Transitionsfunktion verträglich ist. Die Variablen y_1 und y_2 beschreiben dabei die jeweils minimalen Nachfolger von x_1 und x_2, die entweder ein 2-stelliger Knoten oder ein Blatt sind. Die ersten beiden Bedingungen beschreiben also einfach den Lauf des Automaten und werden durch jeden Baum erfüllt. Die letzte Bedingung (iii) verlangt nun zusätzlich, daß der Zustand an der Wurzel ein Endzustand ist, der beschriebene Lauf also akzeptierend ist.

$$\begin{aligned}
&\exists X_1\ldots\exists X_n \\
&\Big[\forall x\,[(\exists y\, xS_2y \leftrightarrow \bigvee_{i\in Q} X_ix) \wedge \bigwedge_{i,j\in Q, i<j} \neg(X_ix \wedge X_jx)] && (i)\\
\wedge\ \ &\forall x\forall x_1\forall x_2\Big[(xS_1x_1 \ \wedge\ xS_2x_2) \rightarrow \Big[\bigvee_{\delta(b,i,j)=k, i',j'\in Q} X_kx \wedge P_bx && (ii)\\
&\qquad \wedge \exists y_1\,[\varphi_{i,i'}(x_1,y_1) \wedge (X_{i'}y_1 \vee (\forall z\, \neg y_2 < z \wedge \bigvee_{\delta(a)=i'} P_ay_1))]\\
&\qquad \wedge \exists y_2\,[\varphi_{j,j'}(x_2,y_2) \wedge (X_{j'}y_2 \vee (\forall z\, \neg y_2 < z \wedge \bigvee_{\delta(a)=j'} P_ay_2))]\Big]\Big]\\
\wedge\ \ &\exists x\,\Big[\forall z\; x \leq z \wedge && (iii)\\
&\bigvee_{k\in F, k'\in Q} \exists y\,[\varphi_{k,k'}(x,y) \wedge (X_{k'}y \vee (\forall z\, \neg y < z \wedge \bigvee_{\delta(a)=k'} P_ay))]\Big]\Big]
\end{aligned}$$

Wir ersetzen nun jeden Mengenquantor $\exists X_i$ durch einen Antikettenquantor $\exists^{ACh} Y_i$. Dabei beschreibt Y_i die Menge der Kodierungen der Knoten von X_i. Eine atomare Formel $X_i x$ in der obigen Formel wird dabei ersetzt durch die Formel $\exists y\ Y_i y \wedge \varphi_C(x,y)$.

Damit haben wir gezeigt, daß jede Baumsprache, deren unärer Teil aperiodisch ist, durch eine Antikettenformel der Form $\exists^{ACh} Y_1 \ldots \exists^{ACh} Y_n\ \varphi(Y_1,\ldots,Y_n)$ definiert werden kann, wobei in $\varphi(Y_1,\ldots,Y_n)$ nur noch 1.Stufe-Quantoren vorkommen. Zusätzlich kann $\varphi(Y_1,\ldots,Y_n)$ nur durch paarweise disjunkte Antiketten erfüllt werden, deren Elemente Blätter sind.

3.3 Die Konstruktion eines sternfreien Ausdrucks

Wir wollen nun die im obigen Abschnitt konstruierte Formel in den Kalkül der sternfreien Ausdrücke übertragen. Zuerst kodieren wir die Belegung der Antikettenquantoren durch Verkettungsvariablen in der Menge $V_0 = \Sigma_0 \times \{1,\ldots,n\}$. Dabei nutzen wir aus, daß die Formel $\varphi(Y_1,\ldots,Y_n)$ nur durch paarweise disjunkte Mengen von Blättern erfüllt werden kann. Sei $t \in T_\Sigma$ und $K_1,\ldots,K_n$ eine Partition der Blätter von t. Dann definieren wir die Kodierung von t und $K_1,\ldots,K_n$ (notiert durch $cod(t,K_1,\ldots,K_n) \in T_{\Sigma\cup V_0}$) durch

$$cod(t,K_1,\ldots,K_n)(k) = \begin{cases} t(k) & k \notin front(t) \\ (t(k),i) & k \in front(t) \wedge k \in K_i \\ (t(k),j) & k \in front(t) \wedge k \notin K_1 \cup \ldots \cup K_n \\ & \text{für ein beliebiges } j \in \{1,\ldots,n\} \end{cases}$$

Die Menge $T' = \{cod(t,K_1,\ldots,K_n) \mid (t,K_1,\ldots,K_n) \models \varphi(Y_1,\ldots,Y_n)\}$ ist 1.Stufe-definierbar, und wir erhalten T aus T', indem alle Variablen (a,i) durch a substituiert werden. Daher ist T sternfrei, wenn T' sternfrei ist. Letzteres zeigen wir, indem wir T' aus endlichen Mengen von Bäumen durch sternfreie Operationen zusammensetzen.

Neben der Variablenmenge V_0 verwenden wir zusätzlich die Variablenmenge $V_1 = \{c,v_1,v_2,v\}$. Nach Lemma 1.5.8 kann für $i,i' \in \{1,\ldots,n\}$ effektiv ein sternfreier Ausdruck konstruiert werden, der die Menge $T_{i,i'}$ definiert. $T_{i,i'}$ besteht aus allen unären Bäumen, deren einziges Blatt mit c beschriftet ist und die Zustand i' in Zustand i überführen.

Wir definieren zuerst die Mengen T_{-v} und T_v aller Bäume in $T_{\Sigma\cup V_0\cup V_1}$, in denen kein bzw. genau ein Knoten vorkommt, der mit v beschriftet ist. T_{-v} erhalten wir, indem wir in allen Bäumen alle Vorkommen von v durch andere 0-stellige Symbole ersetzen.

$$T_{-v} = (\sim \emptyset) \cdot^{v} (\Sigma_0 \cup V_0 \cup \{c,v_1,v_2\})$$

Das Komplement von T_{-v} besteht dann aus allen Bäumen mit wenigstens einem Vorkommen von v. Um T_v zu erhalten, müßen wir in dieser Menge noch die Bäume ausschließen, die wenigstens zwei Vorkommen von v enthalten. Wenn zwei Knoten mit v beschriftet sind, so gibt es einen maximalen Vorgänger dieser beiden Knoten. Über die Beschriftung dieses Knotens erfolgt eine Fallunterscheidung im folgenden Ausdruck.

$$T_v = (\sim T_{-v}) \setminus \bigcup_{b \in \Sigma_2} [(\sim T_{-v}) \cdot^v b(v,v)] \cdot^v (\sim T_{-v})$$

Mit S_0 bezeichnen wir die Menge aller Bäume, deren Blätter ausschließlich mit Variablen aus V_0 beschriftet sind. Wir erhalten S_0, indem wir das Komplement der Menge der Bäume bilden, in denen wenigstens ein Symbol aus der Menge $\Sigma_0 \cup V_1$ vorkommt.

$$S_0 = \sim (T_v \cdot^v (\Sigma_0 \cup V_1))$$

Wir beschreiben nun die Kodierung C. Mit L_v bezeichnen wir die Menge der Bäume, in denen genau ein v auftritt und zwar am lexikographisch kleinsten Blatt. Dieses Blatt können wir beschreiben, indem wir verbieten, daß v im rechten Teilbaum irgendeines Knotens vorkommt.

$$L_v = T_v \setminus \Big(T_v \cdot^v \big(\bigcup_{b \in \Sigma_2} b(v_1, v_2) \cdot^{v_1} T_{-v} \cdot^{v_2} T_v\big)\Big)$$

Unter Verwendung von L_v kann nun ausgedrückt werden, daß die Kodierung der Mengen festlegt, daß Zustand i an der Wurzel angenommen wird. Dabei wird danach unterschieden, ob ein Baum nur aus einem Knoten besteht, oder ob die Wurzel ein 2-stelliges Symbol ist.

$$T^i = S_0 \cap \Big([\bigcup_{j \in Q, \delta(a)=i} (a,j)] \cup [\bigcup_{b \in \Sigma_2} b(v_1,v_2) \cdot^{v_1} S_0 \cdot^{v_2} (L_v \cdot^v \bigcup_{a \in \Sigma_0} (a,i))]\Big).$$

Analog zu Bedingung (ii) in der im vorherigen Abschnitt angegebenen Formel können wir nun beschreiben, daß die Kodierung der Zustände an den Blättern mit der Transitionsfunktion nicht verträglich ist.

$$S_{inkor} = T_v \cdot^v \bigcup_{\delta(b,i,j) \neq k, i', j' \in Q} \left[T^k \cap b(v_1,v_2) \cdot^{v_1} \left(T_{i,i'} \cdot^c T^{i'}\right) \cdot^{v_2} \left(T_{j,j'} \cdot^c T^{j'}\right)\right]$$

Indem wir die Menge S_{inkor} von der Menge der Bäume abziehen, deren Zustandskodierung einen Endzustand an der Wurzel festlegt, erhalten wir einen sternfreien Ausdruck für die Baumsprache T'.

$$T' = \Big(\bigcup_{k \in F, k' \in Q} T_{k,k'} \cdot^c T^{k'}\Big) \setminus S_{inkor}$$

Damit ist der Beweis von Satz 3.0.2 abgeschlossen.

3.4 Diskussion

Die in Abschnitt 3.2 angegebene Kodierung kann auch verwendet werden, um beliebige Formeln der monadischen Logik zweiter Stufe in Antikettenformeln zu übersetzen.

Die einzige Einschränkung ist, daß die Mengenquantoren über Mengen rangieren, die keine unären Knoten enthalten. Ein Mengenquantor $\exists X$ geht dann über in zwei Antikettenquantoren $\exists^{ACh} Y \exists^{ACh} Z$, die auf Mengen von Blättern eingeschränkt werden. Dabei beschreibt Y die Menge der Kodierungen der inneren Knoten in X und Z die Menge der Blätter in X. Eine atomare Formel Xx wird dann ersetzt durch $(Zx \wedge \forall y \neg x < y) \vee (\exists y\, Yy \wedge \varphi_C(x, y))$. Betrachten wir die Beschreibung der akzeptierenden Läufe eines Baumautomaten durch einen Mengenquantor im Beweis von Satz 2.0.2, so erhalten wir:

Satz 3.4.1 *Sei Σ ein Rangalphabet ohne unäre Symbole. Dann kann jede reguläre Baumsprache über Σ durch eine Antikettenformel mit genau einem existentiellen Antikettenquantor definiert werden.*

Die Ausdrucksstärke der sternfreien Ausdrücke beruht darauf, daß Verkettungsvariablen zur Kodierung von Information verwendet werden können. Daraus ergibt sich sofort die Frage, welche Baumsprachen mit Hilfe einer festen Anzahl von Variablen definiert werden können. Das Komplexitätsmaß "Sternhöhe" wird also ergänzt durch das Komplexitätsmaß "Anzahl der verwendeten Verkettungsvariablen". Bisher ist keine Untersuchung dieser Fragestellung bekannt.

Ein anderer Ansatz wurde von Heuter in [18] gewählt. Dort wird nicht die Zahl der Verkettungsvariablen eingeschränkt, sondern die Anzahl der Vorkommen jeder Verkettungsvariablen in einem Baum. Ein sternfreier Ausdruck heißt dort speziell-sternfrei, wenn in jedem Baum in der definierten Baumsprache jedes Verkettungssymbol höchstens einmal vorkommt. In dieser Arbeit wurde gezeigt, daß diese speziell-sternfreien Ausdrücke genau die 1.Stufe-definierbaren Baumsprachen beschreiben.

Ein weiteres Komplexitätsmaß für sternfreie Wortsprachen ist die Punkttiefe. Die Punkttiefe eines sternfreien Ausdrucks ist die Anzahl der Wechsel zwischen den Booleschen Operationen und der Konkatenation. Die Punkttiefe einer sternfreien Wortsprache ist schließlich die minimale Punkttiefe irgendeines sternfreien Ausdrucks, der diese Sprache definiert. In [4, 55] wurde gezeigt, daß die Punkttiefe eine unendliche Hierarchie innerhalb der sternfreien Wortsprachen definiert. Betrachten wir den in Abschnitt 3.3 konstruierten sternfreien Ausdruck, so ist die Punkttiefe nur abhängig von der Punkttiefe der sternfreien Ausdrücke für die Baumsprachen $T_{i,i'}$. Wenn also T eine Baumsprache über einem Rangalphabet ohne unäre Symbole ist, so ist die Punkttiefe von T kleiner als 6.

Schließlich bleibt festzustellen, daß die Länge des konstruierten sternfreien Ausdrucks von der Länge der Ausdrücke für die Baumsprachen $T_{i,i'}$ dominiert wird. Konstruiert man einen sternfreien Ausdruck für $T_{i,i'}$ nach dem Verfahren von Perrin [35], so ist die Länge dieser Ausdrücke doppelt exponentiell in der Anzahl der Zustände des zugrundeliegenden Wortautomaten. Sind keine unären Symbole vorhanden, so kann der oben angegebene Ausdruck so umgeschrieben werden, daß die Länge des Ausdrucks proportional zur Größe des Baumautomaten ist.