Ruprecht-Karls-Universität Heidelberg
Fakultät für Mathematik und Informatik

Maß in Polynomialzeit

Diplomarbeit
vorgelegt von Rupert M. Hölzl
Betreuer: Prof. Dr. Klaus Ambos-Spies
Heidelberg, im Oktober 2006.

AMS Classification: 03D15, 03D55, 68Q15, 68Q30

Inhaltsverzeichnis

Kapitel 1

Einleitung

Wir wollen im folgenden verschiedene Maßbegriffe für die Komplexitätsklasse **P** untersuchen. Die Gründe dafür, überhaupt ein Maß auf Komplexitätsklassen einzuführen, sind vielfältig, z.B. erlaubt uns ein Maßbegriff, eine Variante der probabilitischen Methode anzuwenden, d.h. die Existenz einer Sprache mit bestimmten Eigenschaften in **P** dadurch zu beweisen, dass man zeigt, dass die Menge der Sprachen mit dieser Eigenschaft in **P** Maß 1 hat.

Der erste Ansatz für ein Maß auf einer Komplexitätsklasse, nämlich auf **E** bzw. **EXP** stammt von Lutz [L]. Hier soll es darum gehen sein Konzept auf **P** zu übertragen. Dabei tritt ein charakteristisches Problem auf (siehe 4.1), für das es mehrere mögliche Lösungsansätze gibt. Wir werden drei bereits bekannte Lösungen vorstellen, eine eigene erarbeiten, und sie alle auf Vor- und Nachteile untersuchen.

In Abschnitt 6 werden wir dann noch einen Schritt weitergehen, und versuchen, verschiedene Resultate im Zusammenhang mit « Komplexitätsschichten » zu zeigen. Dabei bedeutet « Schicht » dass wir Mengen von Sprachen betrachten werden, die sich statt durch Überdeckungen beliebiger polylogarithmischer Zeitschranken durch solche mit Schranken von der Gestalt $\log^k |w|$ mit festem k überdecken lassen.

Herrn Prof. Dr. Klaus Ambos-Spies danke ich für die Themenstellung und die laufende Beratung bei einer Reihe von schwierigen Stellen. Mein Dank gilt ebenfalls Herrn PD Dr. Wolfgang Merkle für seine umfangreichen Korrekturvorschläge, die mir halfen, eine Reihe von Fehlern und Unklarheiten zu beseitigen.

Kapitel 2

Definitionen und Notationen

Im folgenden soll es darum gehen, Komplexitätsklassen mit Maßen zu versehen – zunächst allgemein, später speziell die Klasse **P**. Wir werden dabei das aus der Wahrscheinlichkeitstheorie stammende Konzept der Martingale verwenden. Ein Martingal d ist (in unserem Spezialfall) eine Abbildung $\{0,1\}^* \longrightarrow \mathbb{R}_0^+$ mit der Eigenschaft

$$d(w) = \frac{d(w0) + d(w1)}{2}. \tag{2.1}$$

Für eine (endliche oder unendliche) 0/1-Folge w führen wir folgende Notationen ein: $w[i...j]$ seien die Stellen i bis j von w, $w[i]$ entsprechend die i-te Stelle. Die Stellen werden ab 0 gezählt. Wir können eine Sprache L mit einer unendlichen 0/1-Folge w identifizieren, indem wir alle möglichen Wörter in $\{0,1\}^*$ zuerst nach Länge und dann lexikographisch aufsteigend durchnummerieren, und dann $w[i]$ auf 1 setzen, falls das i-te Wort in L ist, und auf 0 sonst. Dieses w nennen wir dann die charakteristische Folge von L. Wenn ein Wort x in der lexikographischen Ordnung die i-te Position erhält, so schreiben wir $\mathrm{pos}(x) = i$. Wir werden Sprache und charakteristische Folge miteinander identifizieren; dies erlaubt es uns z.B. Mengenoperatoren auf w anzuwenden. Wir schreiben $L \upharpoonright x$ für $w[0...(\mathrm{pos}(x) - 1)]$.

Wenn wir nun eine Sprache L auf diese Weise mit ihrer charakteristischen Folge identifizieren, so können wir auf Anfangsstücke $L \upharpoonright x$ dieser Folge das Martingal anwenden. Ein Martingal kann man sich dann als eine Funktion vorstellen, die den Verlauf des Kontostandes eines Spielers angibt, der auf Bits der charakteristischen Folge wettet. Dabei erhält es immer die schon gesehenen Bits als Eingabe, denn davon hängt natürlich i. Allg. der Kontostand ab.

Wir sagen ein Martingal d *überdeckt* eine Sprache L, wenn

$$\limsup_{n\to\infty} d(L \upharpoonright x) = \infty$$

gilt und schreiben dafür $L \in S^\infty[d]$. Eine *Klasse* von Sprachen wird bedeckt, wenn es *ein* Martingal gibt, dass *alle* Sprachen in der Klasse bedeckt. Wir definieren: Eine bedeckte Menge hat das Maß 0. Eine Menge deren Komplement das Maß 0 hat, habe das Maß 1.

Einem Martingal liegt eine bestimmte Strategie zu Grunde, diese gibt immer an, welcher Anteil des aktuell vorhandenen Geldes in der nächsten Wette auf das Ergebnis « 0 » gewettet wird; der Rest wird (der hier benutzten Konvention nach) immer auf « 1 » gesetzt. Eine Strategie erhält wie das Martingal den bisherigen Wettverlauf als Eingabe. Eine geschickt gewählte Strategie würde dann versuchen, aus diesen Informationen auf die zukünftige Entwicklung zu schließen, und dadurch Gewinn zu machen. Wenn eine Strategie an einer bestimmten Stelle $\frac{1}{2}$ ausgibt, so bedeutet das, dass der Wert des Martingals sich im folgenden Schritt nicht ändert.

Wir werden im Folgenden davon sprechen, dass eine Strategie (oder das zugehörige Martingal) an einer bestimmten Stelle wettet. Das bedeutet dass der Wert der Strategie an dieser Stelle $\neq \frac{1}{2}$ ist. Natürlich kann es vom bisherigen Verlauf abhängig sein, ob eine Strategie auf einem bestimmten Wort x wettet oder nicht.

Es gilt

$$|L \upharpoonright x| \approx 2^{|x|}$$

und wir schreiben im folgenden w für $L \upharpoonright x$ wenn wir über das Verhalten eines Martingals an einer bestimmten « Stelle » (d.h. bei einem bestimmten Wort) x reden.

Wenn wir später die verschiedenen Maßansätze einführen, so werden wir von « Γ », « $\Gamma^{\geq}$ », « $\mathscr{Q}$ », « $\mathscr{F}$ » bzw. « Σ » sprechen, um klarzumachen, auf welchen der verschiedenen Maßansätze wir uns jeweils beziehen.

Wenn von einer « Überdeckung » gesprochen wird, so kann damit ein Martingal oder auch eine Strategie mit den im jeweiligen Kontext verlangten Eigenschaften gemeint sein.

Für alle Zeitklassen, über die wir sprechen, gelte $\mathbf{DTIME}(.) = \mathbf{DTIME}(\mathcal{O}(.))$.

Kapitel 3

Ressourcenbeschränktes Maß in Exponentialzeit

Bisher haben wir beliebige Martingale betrachtet. Um diesen allgemeinen Ansatz auf $\mathbf{E}$ im Speziellen anwenden zu können, betrachten wir nach Lutz [L] nur noch bestimmte Martingale, und zwar solche die in Zeit

$$\mathcal{O}(|w|^{\mathcal{O}(1)}) = \mathcal{O}((2^{|x|})^{\mathcal{O}(1)}) = \mathcal{O}(2^{|x|\cdot\mathcal{O}(1)})$$

berechenbar sind. Das Maß, das wir auf diese Weise erhalten, bezeichnen wir aus naheliegenden Gründen mit μ_{poly}. Die Motivation, für die beschriebene Beschränkung der Laufzeit der Martingale besteht darin, dass wir – wenn wir auf $\mathbf{E}$ ein Maß einführen – auch erreichen wollen, dass $\mu_{\text{poly}}(\mathbf{E}) \neq 0$ gilt. Dies können wir unter Ausnutzung der Beschränkung folgendermaßen zeigen.

Satz 1 (Lutz [L])**.** *Es gilt* $\mu_{\text{poly}}(\mathbf{E}) \neq 0$.

Beweis. Sei d gegeben. Wir konstruieren eine Sprache $L \in \mathbf{E}$, welche von d nicht überdeckt wird: Wähle die Bits von L (induktiv!) so, dass d entlang L *nie Gewinn macht* (Diagonalisierung). Es bleibt zu zeigen, dass die Sprache L auch in $\mathbf{E}$ ist. Zeige also: Der Zeitaufwand, um $x \in L$ zu prüfen, ist linear exponentiell. Wegen

$$x \in L \overset{\text{Def. } L}{\Longleftrightarrow} d((L \upharpoonright x)L[x]) \leq d(L \upharpoonright x)$$

müssen wir dafür $L \upharpoonright x$ kennen.[1] Diesen String rekonstruieren wir nun Bit für Bit von Anfang an. Dafür benötigen wir maximal $\mathcal{O}(2^{|x|})$ Aufrufe des Martingals (für alle vor x kommenden Wörter) und jeder solche Aufruf dauert $2^{|x|\cdot\mathcal{O}(1)}$ lang, da wir die Laufzeit des Martingals ja so beschränkt hatten. Insgesamt benötigen wir also

$$\mathcal{O}(2^{|x|\cdot\mathcal{O}(1)}) \cdot \mathcal{O}(2^{|x|})$$

für die Berechnung, also ist $L \in \mathbf{E}$. □

[1] Im Allgemeinen könnte der Wert von d an der jeweiligen Stelle von *allen* Bits in $L \upharpoonright x$ abhängen; genauso möglich wäre aber, dass dem Martingal nur bestimmte Bits von $L \upharpoonright x$ bekannt sein müssen, oder – in ganz einfachen Fällen – sogar kein einziges (z.B. bei einem Martingal, das *nie* wettet). Je mehr Bits erforderlich sind, umso aufwändiger die Rekursion. In Kürze werden wir genau an dieser Stelle ansetzen, um die Rekursionen einfach zu halten.

Würden wir statt der exponentiellen Beschränkung in $|x|$ z.B. eine exponentielle in $|w|$ wählen, so würden wir eine Laufzeit von

$$\mathcal{O}(2^{|w|\cdot\mathcal{O}(1)}) \cdot \mathcal{O}(2^{|x|}) = \mathcal{O}(2^{2^{|x|}\cdot\mathcal{O}(1)} \cdot 2^{|x|})$$

erhalten und die durch die Diagonalisierung entstandene Sprache läge i. Allg. nicht mehr in $\mathbf{E}$. Außerdem lässt sich sogar leicht zeigen, dass mit dieser Schranke $\mathbf{E}$ das Mass 0 erhalten würde.

Auf $\mathbf{E}$ gilt, dass die Vereinigung $\mathbf{C}$ einer Familie $(\mathbf{C}_i)_i$ von Nullmengen wieder eine Nullmenge ist, falls es eine Maschine gibt, die uniform und mit einer Zeitschranke der Form $2^{\mathcal{O}(n)}$ die Überdeckungen d_i für die $\mathbf{C}_i$ berechnet. Wir werden im Folgenden unsere Maße auf $\mathbf{P}$ auf ihr Verhalten bei Vereinigungen untersuchen, und legen dabei die Situation auf $\mathbf{E}$ als Maßstab an, um zu beurteilen, ob sich die betrachteten Maße bezüglich Vereinigungen « gut » verhalten.

Kapitel 4

Verschiedene Ansätze für ein Maß in Polynomialzeit

4.1 Diagonalisierungsprobleme in Polynomialzeit

Versuchen wir auf **P** genauso wie im letzten Abschnitt auf **E** vorzugehen, so müssen wir aus dem selben Grund (also damit die Diagonalisierung klappt) die Laufzeit der Martingale mit

$$\log^{\mathcal{O}(1)} |w| = (\log 2^{|x|})^{\mathcal{O}(1)} = |x|^{\mathcal{O}(1)}$$

beschränken. Man beachte, dass polylogarithmische Zeit in $|w|$ das selbe ist wie polynomielle Zeit in $|x|$. Entsprechend werden wir im Folgenden abwechselnd von « $|w|$-polylogarithmisch » oder « $|x|$-polynomiell » sprechen.

Nach Konvention bekommt die Maschine, die ein Martingal berechnet, als wirkliche Eingabe nicht das Anfangsstück der charakteristischen Folge, sondern nur seine Länge (logarithmisch kodiert), sowie wahlfreien Zugriff (*random access*) auf die einzelnen Bits des Strings. Das bedeutet, dass die Maschine ein Orakelband besitzt, auf das sie die Adresse eines Eingabebits schreiben kann; der Eingabemechanismus liefert dann das entsprechende Bit, das an dieser Stelle steht, zurück. Diese Konstruktion ist erforderlich, da wir sonst ja alleine für das Durchlaufen aller Bits bereits mehr Zeit benötigen würden als erlaubt.

Bei der Diagonalisierung erhalten wir zur Berechnung von $L(x)$ nun eine Laufzeit von

$$\mathcal{O}(|x|^{\mathcal{O}(1)}) \cdot \mathcal{O}(2^{|x|}).$$

Denn jeder Aufruf des Martingals dauert zwar jetzt weniger lang, aber die *Anzahl* der benötigten Aufrufe bleibt gleich groß. Das Produkt ist nicht wie gewünscht polynomiell; die durch Diagonalisierung entstehende Sprache läge also im Allgemeinen nicht wie gewünscht in **P**.

4.2 Schranken auf Abhängigkeitsmengen

Dieser Ansatz stammt von Allender und Strauss [AlS] und im Mittelpunkt steht dabei das folgende Konzept.

Definition 2. *Für eine Maschine M und eine Eingabelänge l sei*

$$H_{M,l} := \{i \mid \exists w : |w| = l \wedge \text{bei Berechnung von } M(w) \text{ wird das Bit mit Adresse } i \text{ gelesen}\}$$

Weiter definieren wir induktiv für $l = 0, 1, 2, \ldots$ die Abhängigkeitsmengen $G_{M,l}$ als den transitiven Abschluss der $H_{M,l}$ in dem Sinne, dass gilt

- $H_{M,l} \subseteq G_{M,l}$
- $\mathrm{pos}(x) \in G_{M,l} \Longrightarrow G_{M,\mathrm{pos}(x)} \subseteq G_{M,l}$
- *$G_{M,l}$ ist die kleinste Teilmenge von $\mathbb{N}$ mit diesen Eigenschaften.*

Wir werden im Folgenden für ein Martingal d nur noch von Abhängigkeitsmengen $G_{d,l}$ sprechen. Dabei sei implizit eine Maschine M gewählt, die d berechnet.

Die Bits in der Abhängigkeitsmenge $G_{d,|w|}$ legen also für alle w einer bestimmten Länge den Wert von $d(w)$ fest. Dabei werden diese Bits über *alle* Eingaben w der entsprechenden Länge betrachtet, und es wird der transitive Abschluss betrachtet, also zu jeder Stelle i in der Abhängigkeitsmenge sind wiederum die Stellen in der Abhängigkeitsmenge, von denen das Kapital an Stelle i abhängt.

Alle Wörter, auf denen d wettet, sind in der Abhängigkeitsmenge an der entsprechenden Stelle und an allen folgenden Stellen.

Satz 3. *Falls bezüglich einer Vorgeschichte w ein Kapital $d(w) \neq 0$ vorhanden ist und auf die nächste Stelle $i = |w|$ ein Betrag $s(w) \neq \frac{1}{2}$ gewettet wird, so ist i in der Abhängigkeitsmenge $G_{d,j}$ aller Stellen $j \geq i$.*

Beweis. Wir beweisen dies per Induktion über l mit $i + l = j$.

Induktionsanfang: Sei $l = 0$. Nehme an, auf i wird (bzgl. w) gewettet. Da $d(w) \neq 0$ ist, folgt

$$d(w0) \neq d(w1),$$

was wiederum impliziert, dass, um $d(w\,L[x])$ zu berechnen, $L[x]$ bekannt sein muss.

Damit ist dann auch $x \in G_{d,i+0}$.

Induktionsschritt $l - 1 \to l$: Sei $b \in \{0, 1\}$ der Ausgang der Wette an Stelle i.

Angenommen $\nexists v : |v| = l - 1 \wedge s(wbv) \neq \frac{1}{2}$. Dann gilt für alle v

$$d(wbv0) = d(wbv1) = d(wbv).$$

Das Martingal hat also an der Stelle $|w| + 1 + |v| + 1$ immer den gleichen Wert wie an Stelle $|w| + 1 + |v|$. Nach Induktionsannahme hängt letzterer Wert vom Ausgang der Wette an Stelle i ab. Also hängt auch der neue Wert hiervon ab.

Es gelte also o.E. $\exists v : s(wbv) \neq \frac{1}{2}$. Das (vor der Wette) vorhandene Kapital $d(wbv)$ hängt nach Induktionsannahme vom Ausgang der Wette auf i ab, sagen wir also z.B.

$$d(wbv) < d(w(1-b)v).$$

Damit $i \notin G_{d,|w|+1+|v|+1}$ gelten kann, muss gelten:

$$d(wbv0) = d(w(1-b)v0) \quad \wedge \quad d(wbv1) = d(w(1-b)v1)$$

Dann wäre aber

$$d(wbv) = \frac{d(wbv0) + d(wbv1)}{2} = \frac{d(w(1-b)v0) + d(w(1-b)v1)}{2} = d(w(1-b)v) \quad \lightning$$

Also muss $i \in G_{d,|w|+1+|v|+1}$ gelten.

□

Die Umkehrung dieses Satzes gilt nicht, wie folgendes Beispiel zeigt.

Beispiel 4. $s(\lambda) = \frac{1}{2}, \quad s(0) = 1, \quad s(1) = 0.$
Dann benötigt man, um $d[s]$ für Wörter der Länge 2 zu berechnen, sowohl das erste als auch das zweite Bit, obwohl auf dem ersten nicht gewettet wurde:
$d(00) = 2, \quad d(01) = 0, \quad d(10) = 0, \quad d(11) = 2.$

Definition 5. *Eine Folge von Mengen $(A_i)_i$ heiße n-polylogarithmisch druckbar, wenn es eine Maschine M und eine Konstante k gibt, so dass M bei Eingabe 1^n alle Elemente $x \in A_n$ in Zeit $\log^k n$ ausgibt.*

Wenn wir die Abhängigkeitsmengen eines Martingal so beschränken, dass sie $|x|$-polynomiell druckbar sind, klappt die Diagonalisierung, da dann rekursiv nur wenige Rechenschritte erforderlich sind. Dies liefert uns einen Maßansatz, den wir im Folgenden mit Γ-Maß bezeichnen wollen.

Definition 6. *Ein Γ-Martingal ist ein Martingal, dessen Abhängigkeitsmenge $|x|$-polynomiell druckbar ist.*

Wie in Abschnitt 4.1 erwähnt, werden wir abwechselnd die beiden äquivalenten Begriffe $|x|$-polynomiell druckbar und $|w|$-polylogarithmisch druckbar verwenden.

Definition 7. *Wir sagen, eine Menge* **C** *habe Γ-Maß 0, wenn es ein Γ-Martingal gibt, das* **C** *überdeckt, und schreiben dafür $\mu^\Gamma(\mathbf{C}) \neq 0$.*

Definition 8. *Die Klasse der dünnen Mengen sei definiert als*

$$\mathbf{SPARSE} := \{A \subset \{0,1\}^* \mid \exists \text{ Polynom } p \; \forall n \; \#\{x \mid x \in A \wedge |x| = n\} < p(n)\}$$

Leider bedeutet die Beschränkung der Abhängigkeitsmengen eine starke Einschränkung der Rechenkraft der Martingale; z.B. kann man zeigen, dass sich **SPARSE** nicht mit solchen Martingalen bedecken lässt:

Satz 9. *Es gilt $\mu^\Gamma(\mathbf{SPARSE}) \neq 0$.*

Beweis. Sei d ein Γ-Martingal. Wir konstruieren eine Sprache $L \in \mathbf{SPARSE}$, die nicht von d bedeckt wird wie folgt: Wo das Martingal nicht wettet setze $x \notin L$. Wo das Martingal wettet, diagonalisiere gegen d. Da das Martingal nur polynomiell oft (relativ zu $|x|$) wetten kann (siehe Satz 3), sind pro Wortlänge nur polynomiell viele Worte in $L \Longrightarrow L \in \mathbf{SPARSE}$. □

SPARSE wird also von diesem Maß nicht bedeckt. Die Sprachen darin sind aber sehr einfach, es reicht, immer mit einem festen Anteil des Kapitals auf « 0 » zu tippen, um unbeschränkten Gewinn zu erzielen. **SPARSE** sollte also – intuitiv betrachtet – ohne weiteres zu bedecken sein. Allgemeiner gesprochen: Wir sind mit dem Ergebnis $\mu^\Gamma(\mathbf{SPARSE}) \neq 0$ nicht zufrieden, da im

klassischen Lebesgue-Sinn die Sprachen mit Dichte $\varepsilon < \frac{1}{2}$ das Maß 0 haben. Falls möglich, hätten wir diese Eigenschaft auch gerne für unser Maß auf $\mathbf{P}$.

Dass dies hier nicht funktioniert, liegt daran, dass das Martingal nach Definition immer das aktuelle Kapital ausgibt; dieses hängt aber von sehr vielen vorherigen Ereignissen ab. Für ein Martingal, das *viele* Wetten tätigt, wäre die Abhängigkeitsmenge also groß, und das Martingal damit unzulässig – selbst wenn die Wetten sehr « einfach » wären. In Abschnitt 4.6 werden wir einen alternativen Maßansatz betrachten, der sich in genau dieser Hinsicht von dem hier betrachteten unterscheidet.

Satz 10. *Es gilt für alle Sprachen $L \in \mathbf{SPARSE}$, dass $\mu^{\Gamma}(\{L\}) = 0$.*

Beweis. Da $L \in \mathbf{SPARSE}$, existiert ein Polynom p, so dass für alle n gilt $\#L^{=n} \leq p(n)$. Definiere ein Martingal wie folgt: Auf den ersten $3p(|x|)$ Wörtern der Länge $|x|$ wird mit festem Einsatz auf 0 gewettet. Im « worst case » können $p(|x|)$ dieser Wörter in L liegen, das Martingal wird also unbeschränkt. Die Größe der Abhängigkeitsmenge des Martingals ist ebenfalls durch $\mathcal{O}(3p(|x|))$ beschränkt, also polylogarithmisch in $|w|$. □

Abschließend wollen wir untersuchen, welche zufälligen Folgen dieser Maßansatz generiert. Das folgende Resultat zeigt, dass sich eine Sprache konstruieren lässt, die einen im Grenzwert nicht verschwindenden Überschuss an 0en aufweist, und die dennoch relativ zum Γ-Maß zufällig ist.

Satz 11. *Zu jedem $\varepsilon > 0$ gibt es ein L, so dass gilt*

$$L \text{ } \Gamma\text{-zufällig } \wedge \lim_{n\to\infty} \frac{\#\{i \leq n \mid L[i] = 0\}}{n} > 1 - \varepsilon.$$

Beweis. Wir erzeugen in längenanwachsender, lexikographischer Reihenfolge die ersten $\lfloor \log |w| \rfloor$ Binärwörter. Wir interpretieren diese Wörter als Kodierungen für Turingmaschinen und lassen nun eine universelle Maschine diese Codes ausführen, normieren die simulierten Funktionen aber so, dass d_i für jedes i ein Γ-Martingal mit Schranke $\log^{\log i} |w|$ bei Eingabe w ist und die Eigenschaft

$$d_i(X \upharpoonright x) := 2^{-i} \text{ für } \log |X \upharpoonright x| \leq i$$

hat (da wir keine Laufzeitschranke einzuhalten haben, stellt diese Normierung kein Problem dar). Die Konstanz des Martingals auf dem Anfangsstück wird uns gleich die Summation der d_i erleichtern; der mit 2^{-i} gewählte Startwert wird dafür sorgen, dass die Summe der Martingale überhaupt konvergiert.

Wir definieren nun ein Martingal

$$d(X \upharpoonright x) = \sum_{i=0}^{\infty} d_i(X \upharpoonright x) = \sum_{i=0}^{\lceil \log |X \upharpoonright x| \rceil - 1} d_i(X \upharpoonright x) + \sum_{\lceil \log |X \upharpoonright x| \rceil}^{\infty} 2^{-i}$$

Da für $n \to \infty$ auch $\log i \to \infty$ gilt, taucht jedes Γ-Martingal früher oder später in der Aufzählung auf. Wir haben also durch die Konstruktion ein Martingal erhalten, das auf allen Mengen gewinnt, die von einem Γ-Martingal bedeckt werden.

Der Berechnungsaufwand der Abhängigkeitsmenge eines d_i ist durch $\log^{\log i} |w|$ beschränkt, so dass die Größe der vereinigten Abhängigkeitsmenge der ersten $\log |w|$ Martingale durch

$$\log |w| \cdot \log^{\log i} |w| \leq \log |w| \cdot \log^{\log \log |w|} |w| \leq |x|^{\log |x| + 1}$$

beschränkt wird. Im Limes wird der Anteil an den jeweils 2^n Wörtern der Länge n, den die Abhängigkeitsmengen ausmachen, also verschwinden.

Diagonalisiert man nun gegen alle Stellen in der Abhängigkeitsmenge des entstandenen Martingals, und belegt alle anderen Stellen der erhaltenen Sprache mit 0en, so erhält man eine Sprache, die im Limes einen beliebig großen Überschuss an 0en aufweist, und die dennoch Γ-zufällig ist. □

4.3 Supermartingale vs. Martingale

Bei folgendem Ansatz von Strauss [S] gehen wir beinahe genauso vor wie in Abschnitt 4.2, allerdings betrachten wir statt Martingalen sogenannte Supermartingale, d.h. an Stelle der Gleichheit in der Fairnessbedingung (2.1) gilt nur

$$d(w) \geq \frac{d(w0) + d(w1)}{2}.$$

Intuitiv bedeutet das, dass das Martingal Geld « wegwerfen » darf. Durch diese Änderung wird die Forderung nach einer beschränkten Abhängigkeitsmenge wesentlich weniger restriktiv, denn wir können nun folgenden Trick anwenden:

Man kann auf einem Block von Wörtern auf die verschiedenste Weise (unterschiedlich hohe) Gewinne machen. In bestimmten Fällen kann man eine geschickte Konstruktion durchführen, und es dadurch schaffen, dass in bestimmten Blöcken auf bestimmten Mengen von Sprachen auf jedem möglichen Weg ein gewisser Mindestgewinn erzielt wird. Am Ende des Blockes kann man danach das « überschüssige » Geld « wegwerfen ». Dadurch kann man es schaffen, für die betrachtete Menge von Sprachen ein bedeckendes Supermartingal zu konstruieren, dessen Abhängigkeitsmenge nie über die Grenzen des aktuell betrachteten Blocks hinausreicht.

Andererseits gilt deshalb bei Supermartingalen anders als bei den fairen Martingalen nicht, dass aus einer $\neq \frac{1}{2}$-Wette an einer bestimmten Stelle folgt, dass diese Stelle in allen folgenden Abhängigkeitsmengen enthalten ist. Dies sehen wir am folgenden Gegenbeispiel.

Beispiel 12. *Die Strategie*[1] *s setze an jeder Stelle der Form* $2i$ *den Anteil* $\frac{3}{4}$ *des vorhandenen Geldes auf « 0 », und an der Stelle* $2i+1$ *verhalte sich s wie folgt: Falls an Stelle* $2i$ *(anders als gewettet) das Bit « 1 » zu finden war, wette an Stelle* $2i+1$ *nicht (es ist wegen der falschen Wette an Stelle i noch* $2 \cdot \frac{1}{4} = \frac{1}{2}$ *des ursprünglichen Geldes vorhanden); falls an Stelle* $2i$ *(wie gewettet) das Bit « 0 » zu finden war, wirf das vorhandene Geld* $\frac{3}{2}$ *bis auf* $\frac{1}{2}$ *weg.*

Es ist klar, dass s an $\frac{2^n}{2}$ *der Wörter der Länge n einen Einsatz* $\neq \frac{1}{2}$ *wettet. Aber: Das Martingal hat an den Stellen der Form* $2i$ *nur ein Bit in der Abhängigkeitsmenge (gerade das Bit* $2i$*), an den Stellen der Form* $2i+1$ *hat das Martingal sogar leere Abhängigkeitsmengen.*

Dieser Unterschied zwischen fairen Martingalen und Supermartingalen wird sich später auswirken, wenn wir versuchen werden, aus Strategien Martingale zu berechnen.

Wenn wir im Maßansatz aus dem letzten Abschnitt statt Martingalen Supermartingale betrachten, so erhalten wir einen Maßansatz, den wir im Folgenden als $\Gamma^{\geq}$-Maß bezeichnen werden.

[1] Eine Strategie, die einem Supermartingal zugrunde liegt, ist natürlich ein 2-dimensionales Vektorfeld $s(w) = (s_0(w), s_1(w))$ mit $s_0(w) + s_1(w) \leq 1$ für alle w.

Definition 13. *Ein $\Gamma^{\geq}$-Martingal ist ein Supermartingal, dessen Abhängigkeitsmenge sich in $|x|$-polynomieller Zeit ausgeben lässt.*

Definition 14. *Wir sagen, eine Menge* $\mathbf{C}$ *habe $\Gamma^{\geq}$-Maß 0, wenn es ein $\Gamma^{\geq}$-Martingal gibt, das* $\mathbf{C}$ *überdeckt, und schreiben dafür $\mu^{\Gamma^{\geq}}(\mathbf{C}) = 0$.*

Wir wenden nun die oben beschriebene Möglichkeit, Abhängigkeitsketten « aufzutrennen » an, um folgendes Resultat zu erhalten.

Satz 15 (Strauss [S])**.** *Es gilt $\mu^{\Gamma^{\geq}}(\mathbf{SPARSE}) = 0$.*

Beweis. Sei $\varepsilon < 1/2$. Teile Σ^* in aufeinanderfolgende Regionen $R_0, R_1, ...$ auf, mit $R_0 := \{w : |w| < 16\}$ und alle weiteren Regionen so, dass die 2^n Wörter der Länge n in Regionen der Größe n^2 geteilt werden (falls die Einteilung nicht aufgeht, ignoriere die restlichen Wörter). Dann enthält R_j Wörter der Länge $n_j > \log j$.

X_j seien die Sprachen mit Dichte kleiner als ε auf der j-ten Region. Mit Hilfe der Chernoff-Ungleichung (z.B. bei Papadimitriou [P], Lemma 11.9) folgt für geeignetes c

$$\mu^{\Gamma^{\geq}}(X_j) \leq e^{-cn^2} \leq 2^{-3n} \leq \frac{1}{j^3}. \quad (\star)$$

Wir definieren nun ein Martingal nur für die Region R_j, das dort sein Kapital von 1 auf j^3 vervielfacht, seine Abhängigkeitsmenge nur in R_j hat (so dass sie also nur polynomiell groß ist), und in n-polynomieller Zeit läuft:

$$d_j(w) = \underbrace{\frac{\sum_{i<\varepsilon n^2 - a} \binom{n^2 - |w[R_j]|}{i}}{2^{n^2-|w[R_j]|}}}_{=:P} \cdot j^3,$$

wobei a die Zahl der 1en in $w[R_j]$ ist. Der Term P gibt dann gerade die Wahrscheinlichkeit dafür an, dass bei bereits bekanntem w bis zu einer bestimmten Stelle die restlichen Stellen in der betrachteten Region noch so belegt werden, dass wir noch eine Sprache in X_j erhalten. Am Anfang einer Region wissen wir noch gar nichts über $w[R_j]$. Hier gilt also obige Abschätzung $(\star)$, so dass wir $d_j(w) \leq 1/j^3 \cdot j^3 = 1$ erhalten. Am Ende einer Region ist $P = 1$ und wir haben $d_j(w) = j^3$.

Auf jedem R_j wetten wir dann mit der Strategie d_j/j^2, riskieren also $1/j^2$ und auf einer Sprache in X_j gewinnen wir damit $1/j^2 \cdot j^3 = j$. Die Reihe $\sum 1/j^2$ konvergiert, wir riskieren also *über alle Regionen gerechnet* endlich viel (siehe auch nächsten Absatz).

Am Ende jeder Region werfen wir alles Geld bis auf $1/j$ weg. Damit wir das tun können, müssen wir mindestens so viel haben. Nehme induktiv an, wir hätten aus der vorherigen Region $1/(j-1)$. Im schlimmsten Fall verlieren wir das gesamte eingesetzte Kapital $1/j^2$. Dann bleiben uns immer noch $1/(j-1) - 1/j^2 \geq 1/j$.

Durch das « Wegwerfen » wird das Gesamtmartingal d, das wir jetzt aus den d_j konstruieren, eine Abhängigkeitsmenge haben, die nie über die aktuellen Region hinausreicht, denn der Gewinn jeder vorherigen Region lässt sich direkt durch Einsetzen von j in $1/j$ *ohne jegliche Rekursion* errechnen. Definiere das Martingal d wie folgt:

$$d(w) = \frac{1}{j} + \frac{d_j(w)}{j^2}$$

für das jeweils « zuständige » j. Eine Sprache deren Dichte kleiner als ε ist (also insbesondere eine Sprache in **SPARSE**), ist in unendlich vielen X_j; am Ende dieser Regionen gilt jeweils

$$d_j(w)/j^2 = j \xrightarrow{j \to \infty} \infty,$$

$d(w)$ wird also unbeschränkt. □

Ein $\Gamma^{\geq}$-Supermartingal kann sein Kapital nicht beliebig oft verdoppeln, wie folgender Satz zeigt.

Satz 16 (Allender und Strauss [AlS2])**.** *Ein $\Gamma^{\geq}$-Supermartingal kann sein Kapital nicht häufiger als $|x|$-polynomiell oft verdoppeln.*

Beweis. Seien $i < j$ zwei *direkt* aufeinanderfolgende Elemente der Abhängigkeitsmenge $G_{d,|w|}$ eines Supermartingals d. Nach Definition einer Abhängigkeitsmenge gilt

$$\forall \big((z, z') : |z| = |z'| = j \leq |w| \wedge z[0...i] = z'[0...i] \wedge z[j] = z'[j]\big) : d(z) = d(z').$$

Gemäß der Supermartingaleigenschaft gilt, dass $d(w[0...i])$ mindestens so groß ist wie der Mittelwert über alle $d(z)$ mit z derart, dass

$$|z| = j \ \wedge \ w[0...i] \sqsubseteq z.$$

Über all diese z's werden höchstens zwei verschiedene Werte $d(z)$ angenommen, abhängig davon, ob $z[j]$ den Wert 0 oder 1 hat (da ja das Bit j das letzte « freie » Bit in der Abhängigkeitsmenge ist). Da nicht die Hälfte der $d(z)$ größer sein kann, als zwei mal der Mittelwert, folgt $d(w[0...j]) \leq 2d(w[0...i])$.

Eine Abhängigkeitsmenge kann höchstens $\text{polylog}(|w|) = \text{poly}(|x|)$ viele Paare (i, j) enthalten, also kann sich der Supermartingalwert nur entsprechend oft verdoppeln. □

Leider bringt auch das Konzept des $\mu^{\Gamma^{\geq}}$-Maßes Probleme mit sich, und zwar mit der Vereinigung von Nullmengen. Beliebige abzählbare Vereinigungen dürfen wir nie erlauben (denn sowohl **E** als auch **P** sind abzählbar, und jedes $\{L\}$ hat Maß 0). Lutz musste deshalb, wie in Abschnitt 3 erwähnt, für sein Maß auf **E** uniforme Vereinigungen betrachten.

Bei dem hier betrachteten Maß können wir aber bereits *nur zwei* Nullmengen finden, die vereinigt das Maß 1 haben (siehe Strauss [S], Theorem 15). Dadurch ist dieser Maßansatz kaum zu gebrauchen.

4.4 Quotientenformulierung

Um das Problem mit der Vereinigung von Nullmengen bei Supermartingalen vom Ende des letzten Abschnitts zu lösen, betrachten wir nun einen weiteren Ansatz von Strauss [S], der vom Supermartingalansatz abgeleitet ist. Wir führen dazu den folgenden *Quotienten* einer Sprache

$$L/x := \{y \,|\, yx \in L\}$$

ein und definieren dann Nullmengen wie folgt.

Definition 17. *Eine Menge* $\mathbf{C}$ *ist* $\mathcal{Q}$*-null (Notation:* $\mu^{\mathcal{Q}}(\mathbf{C}) = 0$*), wenn sie Teilmenge einer* nummerierten Vereinigung *von* Subbasis-Nullmengen *ist, wobei die verwendeten Begriffe wie folgt definiert sind:*

nummerierte Vereinigung: *Für zu vereinigende Mengen* $\{A_i\}$ *existiert eine Maschine* M*, die polynomiell in* $|i| + \log|w|$ *arbeitet, so dass* $d_i(w) = M(i, w)$ *für ein Supermartingal* d_i *gilt, welches* A_i *bedeckt, wobei die Abhängigkeitsmengen der* d_i *eine gemeinsame Schranke haben, die polynomiell in* $|i| + \log|w|$ *ist.*

Subbasis-Nullmenge: *Menge, die unter Quotienten abgeschlossen ist, und von einem Abhängigkeitsmengen-beschränkten Supermartingal bedeckt wird.*

Das Vereinigungsaxiom wird dadurch trivial erfüllbar, denn jede der zu vereinigenden Mengen besitzt eine nummerierte Vereinigung; und eine zulässige Vereinigung war ja eine derartige, dass eine universelle Maschine für die einzelnen Überdeckungen existiert. Wir hätten also eine universelle Maschine M die universelle Maschinen M_i berechnet, d.h.

$$M(\langle i, j, w\rangle) = M_i(\langle j, w\rangle)$$

Sortiere die neue Vereinigung dann so, dass jedes der Martingale, die von den M_i berechnet werden, irgendwann in der Liste auftaucht.

Durch einen Widerspruchsbeweis erhalten wir das folgende Resultat.

Satz 18 (Strauss [S])**.** *Es gilt* $\mu^{\mathcal{Q}}(\mathbf{P}) \neq 0$.

Beweis. Angenommen es gäbe $(A_i)_i$ als nummerierte Vereinigung von Subbasis-Nullmengen, so dass

$$\mathbf{P} \subset \bigcup_i A_i$$

gilt. Dies würde implizieren, dass

$$\exists (d_i)_i, c\ \forall i\ (d_i \text{ bedeckt } A_i \text{ und hat Schranke } n^c).$$

Wähle für jedes d_i durch Diagonalisierung eine Sprache $L_i \in \mathbf{DTIME}(n^{2c})$ (die Schranke gilt, denn die Rekursion, um gegen das Martingal zu diagonalisieren, läuft innerhalb dieser Zeit), die nicht von d_i bedeckt wird, und setze

$$L = \bigotimes_i L_i := \{x10^{i-1} \mid x \in L_i\}$$

L ist in $\mathbf{P}$, denn:

> Wenn $L(y)$ für ein Wort y berechnet werden soll, gehe wie folgt vor: Prüfe zunächst, ob y von der Gestalt $x10^{i-1}$ ist. Falls nicht, verwerfe. Falls ja, so bestimme erst i. Dies dauert linear lang, und ist somit kein Problem (wir reden hier ja über Worte, nicht über charakteristische Folgen!). Dank der Uniformität der nummerierten Aufzählung können wir d_i bestimmen; wir diagonalisieren dann dagegen, um $L_i(x)$ zu errechnen. Die hierfür nötige Rekursion läuft wieder in $|x|$-polynomieller Zeit.

Jedoch gilt für alle i, dass $L/10^{i-1} = L_i \notin A_i$, was für alle i impliziert, dass $L \notin A_i$, denn A_i war unter Quotienten abgeschlossen. $\lightning$ □

Auch dieser Maßansatz bringt leider Probleme mit sich: Bei Lutz' Maß auf **E** galt, dass fast alle Sprachen in **E** **P**-*bi-immun* sind, d.h. dass gilt

$$\mu_{\text{poly}}(\{L \mid \nexists L_P \in \mathbf{P} : \#L_P = \infty \wedge (L_P \subseteq L \vee L_P \subseteq L^c)\}) = 1.$$

Das hier betrachtete Maß dagegen weist im Zusammenhang mit Bi-Immunität unnatürliche Eigenschaften auf, denn:

Beispiel 19. *Sei L mit $\mu^{\mathscr{Q}}(\{L\}) \neq 0$ und setze $L' := \{x1 \mid x \in L\} \cup 0^*$. Nehme an $\{L'\}$ sei null $\xrightarrow{\exists \text{ numm. Vereinig.}}$ $\exists A_i : L' \in A_i$ $\xrightarrow{A_i \text{ unter Quot. abgeschl.}}$ $L'/1 = L \in A_i \Longrightarrow$ L wird von einem Martingal bedeckt.* $\lightning$

Also ist $\{L'\}$ nicht null, obwohl L' eine unendliche, einfache Teilmenge enthält.

Um uns im Folgenden auf diesen Maßansatz zu beziehen, sprechen wir von Überdeckung durch $\mathscr{Q}$-Martingale.

4.5 Martingalfamilien

Eines ist klar: Damit die Diagonalisierung klappt, dürfen die rekursiven Abhängigkeiten für ein Wort nicht zu groß werden. Es gibt aber verschiedene Möglichkeiten, diese Beschränkung der Abhängigkeiten zu erreichen. Die bisher betrachteten Ansätze hatten folgende Nachteile: Die « ausgelassenen » Wörter sind Wörter auf denen *gar nicht* gewettet werden kann (bei Martingalen) bzw. wir können das Kapital nur polynomiell oft verdoppeln (bei Supermartingalen; wie in Satz 16 gesehen).

Jetzt wollen wir einen Ansatz von Moser [Mo] betrachten, der dieses Problem behebt. Dazu betrachten wir Mengen von Wörtern Q_i, so dass $\#Q_i^{=n}$ nur polynomiell in n wächst. Da $\#\{0,1\}^n$ exponentiell wächst, können wir mit einer festen Anzahl solcher Q_i natürlich $\{0,1\}^*$ nicht überdecken. Deswegen nehmen wir bei Bedarf (d.h. sobald die bisher betrachteten Q_i nicht mehr ausreichen) einfach immer weitere Q_i hinzu. Die rekursiven Abhängigkeiten beschränken wir dann so, dass zur Berechnung von $d(L \upharpoonright x)$ mit $x \in Q_j$ nur rekursiv auf Martingalwerte an Stellen, die *ebenfalls* in Q_j sind, zugegriffen werden darf.

Bisher hatten wir es so gehandhabt, dass ein Martingal das aktuell vorhandene Kapital angibt. Jetzt gehen wir so vor, dass wir unterscheiden zwischen einem Gesamtkapital und einem Kapital auf jedem der Blöcke Q_i. *Das Gesamtkapital ist das « Erfolgskriterium » für Martingalfamilien.* Dieses können wir jedoch nicht ohne weiteres berechnen, da das Kapital über die Blockgrenzen hinweg voneinander abhängt, so dass dies i. Allg. eine Rechnung wäre, deren Aufwand die gegebenen Ressourcenschranken übersteigt. Stattdessen verlangen wir nur, dass das Kapital auf jedem der Blöcke berechenbar sei. Wir können dafür zu einem der beiden folgenden äquivalenten Ansätze greifen:

Entweder wir betrachten für jeden der Blöcke sogenannte *Ratenmartingale*, d.h. Martingale, die statt des tatsächlichen Kapitals nur einen Faktor angeben, um den sich das Kapital ändert. Oder wir betrachten « normale » Martingale, die das Kapital angeben. Diese sind dann so gewählt, dass jedes der Martingale das Startkapital 1 erhält, und für eines der Q_i « zuständig » ist, d.h. nur auf diesen wettet, und als Kapital nur ein Kapital « für diesem Zuständigkeitsbereich » ausgibt (dies ist äquivalent dazu, die Abhängigkeitsmenge jedes Martingals zu beschränken). Um das *Gesamt*kapital zu errechnen, müssen dann die entsprechenden Werte multipliziert werden; also

entweder die Ratenmartingale über alle Wörter, oder die Martingale über alle Q_i. Diese beiden Konzepte sind *hier* äquivalent, weil die Martingale aus den polynomiell vielen Ratenmartingal-Faktoren in polynomieller Zeit errechnet werden können (normalerweise wäre ein Martingal natürlich wesentlich aufwendiger zu berechnen, als ein Ratenmartingal).

Beachte: Γ-Martingale und (einzelne) $\mathscr{F}$-Martingale sind im Wesentlichen das gleiche Konzept. Hier eine formale Definition.

Definition 20. *Eine* **P***-Familie von Ratenmartingalen ist ein Tripel* $(\{D_i\}_i, \{Q_i\}_i, \mathrm{ind})$*, so dass die* $\{D_i\}_i$ *Ratenmartingale sind und so dass*

- $Q_i : \mathbb{N} \to \mathcal{P}(\{0,1\}^*)$ *disjunkte Mengen sind mit* $Q_i(n) \subseteq Q_i(m)$ für $n \leq m$*, so dass es eine universelle Maschine gibt, die bei Eingabe* $(i, 1^n)$ *die Strings in* $Q_i(n)$ *in n-polynomieller Zeit ausgibt,*
- *die Wörter in* $Q_i(n)$ *höchstens Länge n haben,*
- *jedes Wort x in genau einem* $\bigcup_n Q_i(n)$ *enthalten ist,*
- $\mathrm{ind} : \{0,1\}^* \to \mathbb{N}$ *eine in* $|x|$*-polynomieller Zeit berechenbare Funktion ist, so dass*

$$x \in Q_j(|x|) \Rightarrow j = \mathrm{ind}(x),$$

- $D_i(L \upharpoonright x) = M^{L\upharpoonright x}(x,i)$*, wobei M eine (relativ zu* $|x|$*) in polynomieller Zeit arbeitende Maschine ist, die ihr Orakel nur nach Strings y in* $Q_i(|x|)$ *befragt, mit* $y \leq x$*.*

Die Nummerierung der Q_i's ist bei Moser [Mo] implizit so gewählt, dass ihre Reihenfolge der Reihenfolge der jeweiligen minimalen Wörter in jedem Q_i entspricht. Dies schlägt sich auch in der folgenden Definition der Gewinnbedingung für Martingalfamilien nieder.

Definition 21. *Wir sagen, eine Klasse* **C** *hat* $\mathscr{F}$*-Maß 0, falls es eine* **P***-Familie von Ratenmartingalen* $(\{D_i\}_i, \{Q_i\}_i, \mathrm{ind})$ *gibt, so dass für alle* $L \in \mathbf{C}$ *gilt:*

$$\limsup_{n\to\infty} W_D(L \upharpoonright n) := \limsup_{x\to\infty} \prod_{i\leq 2^{|x|}} \prod_{y\leq x} D_i(L \upharpoonright y) = \infty$$

Satz 22. *Es gilt* $\mu^{\mathscr{F}}(\mathbf{P}) \neq 0$*.*

Beweis. Wir diagonalisieren einfach, um eine nicht bedeckte Sprache zu erhalten, was ja jetzt kein Problem mehr ist. Beachte dabei nur, dass gerade immer gegen das Martingal in der Familie diagonalisiert wird, das für das aktuell betrachtete Wort « zuständig » ist. □

Satz 23. *Es gilt* $\mu^{\mathscr{F}}(\mathbf{SPARSE}) = 0$*.*

Beweis. Wette einfach auf jedem String z.B. $\frac{3}{4}$ auf « 0 », was ja jetzt kein Problem mehr ist, da wir z.B. *jedes* Wort in einen eigenen Zuständigkeitsbereich packen können. □

Betrachten wir das Verhalten bzgl. Bi-Immunität.

Beispiel 24. *Die Konstruktion aus Beispiel 19 stellt für dieses Maß kein Problem dar: Wir können $L' := \{x1 \mid x \in L\} \cup 0^*$ bedecken, indem wir alle Strings in 0^* in einen Block stecken, auf diesem immer auf « 1 » wetten, und die übrigen Strings in andere Blöcke verteilen, auf denen wir aber nicht wetten.*

Eine Möglichkeit die uns dieser Maßansatz eröffnet, ist die Definition der Hausdorff-Dimension. Dabei handelt es sich um die Zahl

$$\inf\{s \in [0,1] \mid \exists D \ \mathbf{P}\text{-Martingalfamilie: } \limsup_{n\to\infty} 2^{n(s-1)} W_D(L \upharpoonright n) = \infty\}$$

Anders formuliert: Wir erheben für jeden Wettschritt eine « Steuer » von 2^{s-1}, und untersuchen, wie wir s wählen können, so dass das Martingal immer noch unbeschränkt ist. Offensichtlich würde $s = 0$ bedeuten, dass *auf jeden Fall* jeglicher Gewinn von der Steuer aufgefressen wird. Eine Hausdorff-Dimension von 0 bedeutet also, dass wir an diese « 100% Steuer » *beliebig nahe* herankommen dürfen, und das Martingal immer noch unbeschränkten Gewinn macht. Eine Menge mit Hausdorff-Dimension 0 ist also sehr einfach.

Die Definition der Hausdorff-Dimension ist mit den anderen Maßansätzen nicht möglich, da sich dort das Kapital höchstens polynomiell oft verdoppeln kann. Eine *exponentiell oft* erhobene « Steuer » würde diesen Gewinn also immer auffressen.

Der Maßansatz hat leider auch einen Nachteil: Die Vereinigung von Nullmengen ist i. Allg. nur dann null, wenn es Martingale gibt, die die zu vereinigenden Mengen bedecken, und die alle die selben « Zuständigkeitsbereiche » Q_i haben (siehe Moser [Mo], Theorem 4.1).

Um uns im Folgenden auf diesen Maßansatz zu beziehen, sprechen wir von $\mathscr{F}$-Martingalen bzw. Martingalfamilien.

4.6 Abhängigkeitsmengenbeschränkung auf Strategien

Wir wollen nun folgenden alternativen Maßansatz betrachten.

Definition 25. *Eine Menge $\mathbf{C}$ habe Σ-Maß 0 (Notation: $\mu^\Sigma(\mathbf{C}) = 0$), falls eine Σ-Strategie s bzw. ein Σ-Martingal $d[s]$ existieren, die $\mathbf{C}$ bedecken, d.h. falls es eine Konstante k und eine Strategie s gibt, so dass*

$$G_{s,|w|} \text{ ist } \log^k |w|\text{-druckbar } \wedge s \in \mathbf{DTIME}(\log^k |w|) \wedge \mathbf{C} \subseteq S^\infty[d[s]]$$

Die Abhängigkeitsmengenbeschränkung wird also über die Abhängigkeitsmenge der Strategie, nicht über die des Martingals, verhängt. Dieser Maßansatz ist natürlich nicht mehr äquivalent zu den bereits betrachteten Maßansätzen, denn die Laufzeit der Martingale, die sich aus den Strategien errechnen ließen, wäre unter Umständen sehr hoch.

Intuitiv kann man sagen, dass dieser Maßansatz es erlaubt, *viele* aber nicht besonders « gut informierte » Wetten durchzuführen: Denn die *Anzahl* der Wetten vergrößert i. Allg. nur die Abhängigkeitsmenge des induzierten *Martingals*. Dagegen sind « gut informierte » Wetten, also solche, die aus den bereits aufgetretenen Bits geschickte Schlußfolgerungen für die Zukunft ziehen,

nur eingeschränkt möglich, da ja nur ein Teil der Bits abgefragt werden darf, um die Strategie zu berechnen.

Wir wollen nun untersuchen, inwieweit dieser Maßansatz die grundlegenden maßtheoretischen Axiome erfüllt.

Satz 26. *Für alle $A \in \mathbf{P}$ wird die Einermenge $\{A\}$ wird von einem Σ-Martingal bedeckt.*

Beweis. Ohne Einschränkung sei $A \in \mathbf{DTIME}(n^k) \subset \mathbf{P}$. Wir definieren folgende Strategie:

$$s_A(X \upharpoonright x) := 1 - A(x)$$

Dann gilt wegen $A(x) \in \mathbf{DTIME}(|x|^k)$ auch

$$s_A(X \upharpoonright x) \in \mathbf{DTIME}(|x|^k) = \mathbf{DTIME}(\log^k |w|).$$

Es ist klar, dass A von s_A bedeckt wird, und dass s_A leere Abhängigkeitsmengen hat (ganz im Gegensatz zu $d[s_A]$!). □

Satz 27. *Es gilt $\mu^{\Sigma}(\mathbf{P}) \neq 0$.*

Beweis. Dazu konstruieren wir zu jeder Strategie s eine Sprache A_s, die von s nicht bedeckt wird, wie folgt:

$$x \in A_s :\Leftrightarrow s(X \upharpoonright x) \geq \frac{1}{2}$$

Beachte, dass dies eine induktive Definition ist, denn der bisherige String $X \upharpoonright x$ geht in die Definition mit ein.

Dass A_s von s nicht bedeckt wird, ist klar. Wir müssen noch zeigen, dass $A_s \in \mathbf{P}$ gilt, also dass sich $A_s(x)$ in polynomieller Zeit berechnen lässt. Dies geht wie folgt:

Berechne $s(X \upharpoonright x)$. Dies ist nach Annahme in Zeit $\mathcal{O}(\log^k |w|)$ möglich. Falls Bits aus $X \upharpoonright x$ zur Berechnung abgefragt werden, berechne diese rekursiv mit dem selben Aufwand. Da die Abhängigkeitsmenge von s $\mathcal{O}(\log^k |w|)$-druckbar ist, beträgt der Gesamtaufwand maximal $\mathcal{O}(\log^{2k} |w|) = \mathcal{O}(\log^{2k} 2^{|x|}) = \mathcal{O}(|x|^{2k})$. □

Satz 28. *Teilmengen von Σ-Nullmengen sind wieder Σ-Nullmengen*

Beweis. Die Aussage des Satzes folgt trivialerweise aus der Definition von Überdeckungen. □

4.7 Vergleich von $\mathscr{F}$-Maß und Σ-Maß

In diesem Abschnitt wollen wir untersuchen, wie sich $\mathscr{F}$- und Σ-Maß zueinander verhalten.

Satz 29. *Jede Martingalfamilie lässt sich auch als Σ-Martingal auffassen.*

Beweis. Sei $M = (d_i, Q_i, \mathrm{ind})$ eine Martingalfamilie. Jedes der Mitgliedsmartingale d_i hat eine Abhängigkeitsmengenbeschränkung, die durch $\#Q_i(n)$ gegeben ist. Die gleiche Schranke gilt dann auch für die entsprechende Mitgliedstrategie s_i, die d_i zu Grunde liegt (siehe dazu Satz 37 unten). Wir definieren nun eine Strategie s wie folgt:

$$s(w) := s_i(w) \text{ mit } i = \mathrm{ind}(|w|)$$

Bei der Konstruktion dieser neuen Strategie haben wir ausgenutzt, dass die Mitgliedsmartingale (und damit die Mitgliedsstrategien) einer Martingalfamilie uniform berechenbar sind, und dass uns die Funktion ind zu jeder Stelle angibt, welche Mitgliedsstrategie an dieser Stelle benötigt wird.

Diese Strategie ist nun nach Konstruktion in zulässiger Laufzeit berechenbar und ihre Abhängigkeitsmengen sind wie gewünscht beschränkt (denn die Mitgliedsmartingale einer Familie sind nach Definition durch *eine einheitliche* Laufzeit- und Abhängigkeitsmengenschranke beschränkt). □

Dies bedeutet natürlich auch, dass gilt

$$\mu^{\mathscr{F}}(\mathbf{C}) = 0 \Rightarrow \mu^{\Sigma}(\mathbf{C}) = 0.$$

Nun wollen wir noch untersuchen, ob diese Inklusion echt ist. Zunächst betrachten wir der Anschauung halber das Beispiel eines Σ-Martingals, das sich *nicht* als Martingalfamilie auffassen lässt:

Beispiel 30. *Wir definieren eine Strategie wie folgt:*

$$s(w) := \sum_{i \in I(w)} w(i) \bmod 2$$

wobei die Indexmenge als $I(\lambda ... x) := \{\lambda, 0, 00, ..., 0^{|x|}\}$ *definiert sei.*

Diese Strategie hat offensichtlich eine geeignete Abhängigkeitsmengen- und Laufzeitschranke, um für das Σ-Maß zulässig zu sein. In eine Martingalfamilie lässt sie sich dagegen nicht packen, da es nur einen einzigen Zuständigkeitsbereich gäbe, der dann alle Wörter umfassen würde und damit zu groß wäre.

Dies bedeutet natürlich *noch nicht*, dass es tatsächlich Mengen mit $\mu^{\Sigma}(\mathbf{C}) = 0 \ \wedge \ \mu^{\mathscr{F}}(\mathbf{C}) \neq 0$ gibt. Dazu betrachte jedoch folgenden Satz.

Satz 31. *Es gibt eine Klasse* $\mathbf{C}$*, so dass gilt* $\mu^{\Sigma}(\mathbf{C}) = 0$ *und* $\mu^{\mathscr{F}}(\mathbf{C}) \neq 0$*.*

Beweis. Setze

$$\mathbf{C} := \left\{ L^* \;\middle|\; \begin{array}{l} L \text{ ML-zufällig} \wedge L^* \text{ entsteht aus } L \text{ wie folgt: Die ersten } n \text{ Bits} \\ \text{jedes Blocks der } 2^n \text{ Wörter mit Länge } n \text{ werden als Adresse} \\ \text{innerhalb des restlichen Blocks interpretiert. Das dort befindli-} \\ \text{che Bit wird dann an die } n+1\text{-te Stelle innerhalb des Blocks} \\ \text{geschrieben (das bisher dort stehende Bit wird überschrieben).} \end{array} \right\}$$

Eine Σ-Strategie kann auf $\mathbf{C}$ unbeschränkt gewinnen, indem sie innerhalb jedes Blocks immer die Adresse und das $n+1$-te Bit ausliest, prüft, ob die aktuelle Stelle die adressierte ist, und gegebenenfalls gemäß des $n+1$-ten Bits wettet. Die Abhängigkeitsmenge umfasst jeweils pro Block nur die ersten $n+1$ Bits.

Eine Martingalfamilie **F** kann dagegen auf dieser Klasse nicht gewinnen: Sei $p(n)$ die gemeinsame Schranke von **F**. Wir betrachten für n beliebig den Block der Wörter mit Länge n. Sei $i_0^{(n)}$ die $(n+1)$-te Stelle in diesem Block und $i_1^{(n)}$ die Stelle, die von den ersten n Bits des Blocks adressiert wird.

1. Es kann nur endlich viele n geben, so dass $i_0^{(n)}$ und $i_1^{(n)}$ im selben Zuständigkeitsbereich von **F** liegen. Denn sonst wären auf jedem Block nur $p(n)$ Belegungen für die ersten n Bits des Blocks möglich, der die Adresse von $i_1^{(n)}$ beschreibt, was bedeuten würde, dass sich ein eingesetztes Kapital um den Faktor $\frac{2^n}{p(n)}$ vervielfachen ließe. Würde man dann auf jedem Block z.B. den Einsatz $\frac{1}{n^2}$ setzen, so ließe sich – nur unter Verwendung des Adressblocks, also auch auf der ursprünglichen ML-zufälligen Folge – unbeschränkter Gewinn für $n \to \infty$ erzielen. $\lightning$

2. Nur die Wetten auf die $i_1^{(n)}$ und $i_0^{(n)}$ erzielen für $n \to \infty$ bereits unbeschränkten Gewinn. Denn wäre das Produkt der entsprechenden Änderungsfaktoren endlich, könnten die entsprechenden Wetten weggelassen werden, und man würde dennoch unbegrenzten Gewinn machen. Die entstehende Martingalfamilie würde auf der ursprünglichen, ML-zufälligen Menge ebenfalls gewinnen. $\lightning$

3. Nur die Wetten auf die $i_1^{(.)}$ *oder* die auf die $i_0^{(.)}$ erzielen bereits unbeschränkten Gewinn. Denn wegen Punkt 1 sind die Wetten auf $i_1^{(n)}$ und $i_0^{(n)}$ für fast alle n voneinander unabhängig. Also muss entweder das Produkt über alle Änderungsfaktoren der Wetten auf die $i_1^{(.)}$ oder das Produkt über alle Änderungsfaktoren der Wetten auf die $i_0^{(.)}$ unendlich sein.

4. Angenommen das Produkt über alle Wetten auf die $i_1^{(.)}$ ist unendlich. Dann erhielten wir eine berechenbare Strategie, die auf der ursprünglichen, ML-zufälligen Menge gewinnt, wie folgt: Für eine Eingabe w bestimmt die berechnende Maschine zunächst, ob das nächste Bit von der Form $i_1^{(.)}$ ist. Falls nein, wette nicht. Falls ja, simuliere die Funktion ind der Martingalfamilie, um zu bestimmen, welches Mitgliedsmartingal der Familie an dieser Stelle wetten würde. Simuliere dann dieses Mitgliedsmartingal, um den Einsatz zu erhalten, den das Mitgliedsmartingal auf die Stelle $i_1^{(.)}$ wetten würde. Gib das Ergebnis der Simulation aus.

5. Angenommen das Produkt über alle Wetten auf die $i_0^{(.)}$ ist unendlich. Dann erhielten wir eine berechenbare Strategie, die auf der ursprünglichen, ML-zufälligen Menge gewinnt, wie folgt: Für eine Eingabe w bestimmt die berechnende Maschine zunächst, ob das nächste Bit von der Form $i_1^{(.)}$ ist. Falls nein, wette nicht. Falls ja, simuliere die Funktion ind der Martingalfamilie, um zu bestimmen, welches Mitgliedsmartingal der Familie an *der letzten Stelle der Form* $i_0^{(.)}$ gewettet hätte. Simuliere dann dieses Mitgliedsmartingal für ein geeignetes Anfangsstück von w, um den Einsatz an dieser Stelle $i_0^{(.)}$ zu erhalten. Gib das Ergebnis der Simulation als Einsatz an der aktuellen Stelle der Form $i_1^{(.)}$ aus.

Eine berechenbare Strategie, die auf der ursprünglichen, ML-zufälligen Menge gewinnt, kann es nicht geben. Damit haben wir einen Widerspruch erhalten, und der Satz ist bewiesen. □

Unser neuer Maßansatz ist also tatsächlich stärker als die Martingalfamilien bei Moser.

Kapitel 5

Gemeinsame Eigenschaften der Maßbegriffe

Wir möchten im nächsten Kapitel Aussagen über einzelne Schichten der Klasse der polylogarithmischen Überdeckungen machen, d.h. über Überdeckungen deren Schranke von der Form $\mathcal{O}(\log^k |w|)$ für festes k ist. Als Vorbereitung darauf werden wir hier einige Untersuchungen durchführen, die uns später nützlich sein werden.

In verschiedenen Konstruktionen ist es hilfreich, die erwähnten Schichten nicht über die Laufzeit von bedeckenden Martingalen zu definieren, sondern über die von Strategien, deren zugehörige Martingale dann eine Menge bedecken. Dies ermöglicht es uns, Resultate, die für den Fall exponentiellen Maßes gelten, auf den Fall polynomiellen Maßes zu übertragen. In den Artikeln [AlS], [Mo], denen unsere Maßbegriffe entstammen, wurden keine derartigen Schichten untersucht, so dass es uns freisteht, sie so zu definieren. Natürlich wollen wir vermeiden, dass der Fall auftritt, dass für eine Klasse eine polylogarithmische, überdeckende Strategie existiert, aber kein polylogarithmisches, überdeckendes Martingal. Wir müssen also zeigen, dass die Existenz einer (irgendwie) polylogarithmisch beschränkten Strategie die Existenz eines polylogarithmisch beschränkten Martingals (wie in den Artikeln verlangt) impliziert. Dass dies zumindest beim Ansatz der Abhängigkeitsmengenbeschränkung von Allender und Strauss [AlS] gilt, sehen wir in Satz 39.

Die von uns betrachteten Schranken müssen jedoch so beschaffen sein, dass sie nicht nur Laufzeiten, sondern auch die Abhängigkeitsmengen der betrachteten Überdeckungen beschränken. Um mit den zitierten Ansätzen verträglich zu sein, müssen wir die Abhängigkeitsmengen der Martingale, nicht der Strategien beschränken: Wie wir in Abschnitt 5.1 sehen, kann nämlich bei beschränkter Abhängigkeitsmenge einer Strategie die Abhängigkeitsmenge des zugehörigen Martingals dennoch beliebig größer sein, so dass uns keine andere Wahl bleibt, als die Abhängigkeitsmengenbeschränkung direkt auf das von einer Strategie induzierte Martingal zu verhängen.

Dies wiederum stellt uns vor die nächste Herausforderung: Damit die Umrechnung von Strategie zu Martingal in Abschnitt 5.2 funktioniert, darf nur auf wenigen Stellen der Strategie ein Betrag $\neq \frac{1}{2}$ gewettet werden, denn bei der Umrechnung müssen die Faktoren an diesen Stellen miteinander multipliziert werden: Die Abhängigkeitsmengenbeschränkung des Martingals soll sicherstellen, dass es nur « wenige » solche Stellen gibt; die der Strategie, dass jeder einzelne der Faktoren leicht zu berechnen ist. In Satz 37 werden wir sehen, dass ersteres bei fairen Martingalen automatisch gegeben ist. Bei Supermartingalen lässt sich eine derartige Argumentation aber nicht durchführen, wie Gegenbeispiel 12 gezeigt hat.

Deshalb müssen wir für den Quotientenansatz einzelne « Schichten » doch über Martingale definieren. Wir werden trotzdem Überdeckungen mit Hilfe von Strategien definieren, müssen dann aber eben die Laufzeit des zugehörigen Martingals untersuchen, um sicherzustellen, dass es zulässig ist.

Wenn von einer $\mathcal{O}(\log^k n)$-Schranke auf eine Strategie s die Rede ist, ist also – außer bei der Quotientenformulierung – damit gemeint, dass die Schranke $\mathcal{O}(\log^k n)$ auf die Laufzeit von s und auf die Abhängigkeitsmenge *des von s induzierten Martingals* $d[s]$ verhängt wird – *nicht* auf die Abhängigkeitsmenge der Strategie selbst.

Definition 32. *Für eine Konstante k und den Maßbegriffe μ^Γ möge $\mu^\Gamma_{\log^k}$ den Maßbegriff bezeichnen, den wir erhalten, wenn wir nur Γ-Martingale mit der Schranke $\mathcal{O}(\log^k |w|)$ zulassen, d.h. Martingale, deren Abhängigkeitsmenge $\mathcal{O}(\log^k n)$-druckbar ist und deren zugrundeliegende Strategie in* $\mathbf{DTIME}(\log^k n)$ *liegt.*

Definition 33. *Für eine Konstante k und den Maßbegriffe $\mu^{\mathscr{Q}}$ möge $\mu^{\mathscr{Q}}_{\log^k}$ den Maßbegriff bezeichnen, den wir erhalten, wenn wir nur $\mathscr{Q}$-Überdeckungen mit der Schranke $\mathcal{O}(\log^k |w|)$ zulassen, d.h. nummerierte Aufzählungen, für deren Mitgliedsmartingale gilt, dass ihre Abhängigkeitsmenge $\mathcal{O}(\log^k n)$-druckbar ist und dass sie in* $\mathbf{DTIME}(\log^k n)$ *liegen.*

Definition 34. *Für eine Konstante k und den Maßbegriffe $\mu^{\mathscr{F}}$ möge $\mu^{\mathscr{F}}_{\log^k}$ den Maßbegriff bezeichnen, den wir erhalten, wenn wir nur Martingalfamilien mit der Schranke $\mathcal{O}(\log^k |w|)$ zulassen, d.h. Martingalfamilien für deren Mitglieder $\{d_i\}_i$ und Zuständigkeitsbereiche $\{Q_i\}_i$ gilt:*

$$\big(\exists s_i\ \mathscr{F}\text{-Strategie} : s_i \in \mathbf{DTIME}(\log^k n) \wedge d_i = d[s_i]\big) \wedge \big(Q_i(n) \text{ ist } \mathcal{O}(\log^k n)\text{-druckbar}\big)$$

Definition 35. *Für eine Konstante k und den Maßbegriffe μ^Σ möge $\mu^\Sigma_{\log^k}$ den Maßbegriff bezeichnen, den wir erhalten, wenn wir nur Σ-Strategien mit der Schranke $\mathcal{O}(\log^k |w|)$ zulassen, d.h. Strategien, die in* $\mathbf{DTIME}(\log^k n)$ *liegen, und deren Abhängigkeitsmenge $\mathcal{O}(\log^k n)$-druckbar sind.*

5.1 Abhängigkeitsmengen von Martingalen und Strategien

5.1.1 Einfache Abhängigkeitsmengenbeschränkung

Der Begriff der Abhängigkeitsmenge kann sowohl auf Martingale als auch auf Strategien angewendet werden, wie wir das schon in Abschnitt 4.6 getan haben; dabei entstehen auch tatsächlich zwei verschiedene Begriffe. Das heißt, die Abhängigkeitsmenge eines Martingals kann wesentlich größer sein, als die Abhängigkeitsmenge der ihm zugrunde liegenden Strategie, wie folgendes Beispiel zeigt.

Beispiel 36. *Die Strategie s wette immer auf 0, d.h. $s \equiv 1$. Diese Strategie hat leere Abhängigkeitsmengen. Das daraus entstehende Martingal $d[s]$ hat dagegen maximale Abhängigkeitsmengen, denn um $d[s](w)$ zu berechnen muss jedes Bit von w abgefragt werden (ist ein beliebiges Bit gleich 1, so folgt $d[s](w) = 0$, sonst $d[s](w) = 2^{|w|}$).*

Wir wollen zeigen, dass, wenn wir eine Schranke über die Abhängigkeitsmenge eines Martingals verhängen, auch die Abhängigkeitsmenge der Strategie beschränkt ist.

Satz 37. *Es gilt* $G_{s,k} \subseteq G_{d[s],k+1}$.

Beweis. Sei w ein beliebiger String und nehme an, Stelle $i := |w|+1$ sei in der Abhängigkeitsmenge $G_{s,k}$ für eine Länge $k \geq i$, d.h. $\exists v : |w| + 1 + |v| = k \wedge s(w0v) \neq s(w1v)$.
Wir möchten zeigen, dass entweder $d[s](w0v0) \neq d[s](w1v0)$ oder $d[s](w0v1) \neq d[s](w1v1)$, denn dann wäre i in der Abhängigkeitsmenge $G_{d[s],k+1}$. Wir zeigen dies durch Widerspruch: Nimm an $d[s](w0v0) = d[s](w1v0) \wedge d[s](w0v1) = d[s](w1v1)$. Wir folgern $s(w0v) = s(w1v)$, ein Widerspruch, wie folgt: Allgemein gilt

$$\begin{aligned}
d[s](w0v1) &= 2 \cdot (1 - s(w0v)) \cdot d[s](w0v) && (5.1)\\
d[s](w0v0) &= 2 \cdot s(w0v) \cdot d[s](w0v) && (5.2)\\
d[s](w1v1) &= 2 \cdot (1 - s(w1v)) \cdot d[s](w1v) && (5.3)\\
d[s](w1v0) &= 2 \cdot s(w1v) \cdot d[s](w1v) && (5.4)
\end{aligned}$$

Aus den oben angenommenen Gleichheiten folgt:

$$\begin{aligned}
(1 - s(w0v)) \cdot d[s](w0v) &= (1 - s(w1v)) \cdot d[s](w1v) && (5.5)\\
s(w0v) \cdot d[s](w0v) &= s(w1v) \cdot d[s](w1v) && (5.6)
\end{aligned}$$

Addition liefert $d[s](w0v) = d[s](w1v)$; o.E. können wir annehmen, dass dieser Wert $\neq 0$ ist; denn wäre dies (für alle v wie oben gewählt) der Fall, so würde der Wert der Strategie an dieser Stelle irrelevant, und wir könnten ihn so wählen, dass $i \notin G_{s,k}$. Einsetzen und Gleichsetzen von (5.2) & (5.4) liefert

$$s(w0v) = s(w1v) \;\lightning$$

□

5.1.2 Quotientenformulierung

Es stellt sich nun die Frage, ob auch für Supermartingale gilt, dass Abhängigkeitsmengen von Strategien in den Abhängigkeitsmengen der zugehörigen Martingale enthalten sind. Intuitiv würde man sagen, dass dem nicht so ist, da Supermartingale durch das Wegwerfen von « Geld » Abhängigkeitsketten zerstören können. Diese Intuition spiegelt sich in dem folgenden Gegenbeispiel wieder.

Beispiel 38. *Die Strategie s wette nicht, außer an folgenden beiden Stellen: An Stelle i setze s $\frac{3}{4}$ des vorhandenen Geldes auf « 0 », und an der Stelle $k > i$ verhalte sich s wie folgt: Falls an Stelle i (anders als gewettet) das Bit 1 zu finden war, wette an Stelle k nicht (es ist wegen der falschen Wette an Stelle i noch $2 \cdot \frac{1}{4} = \frac{1}{2}$ des ursprünglichen Geldes vorhanden); falls an Stelle i (wie gewettet) das Bit 0 zu finden war, wirf das vorhandene Geld $\frac{3}{2}$ bis auf $\frac{1}{2}$ weg.*

Es ist klar, dass $i \in G_{s,k}$, da die Strategie sich – je nach dem Bit an Stelle i – an der Stelle k unterschiedlich verhält. Aber: Das Martingal hat an Stelle $k+1$ (und an allen folgenden Stellen) unabhängig vom Bit an Stelle i immer den Wert $\frac{1}{2}$; das Bit an Stelle i wird zur Berechnung also nicht benötigt. Also gilt $G_{s,k} \not\subseteq G_{d[s],k+1}$.

Dennoch gilt aber, dass eine Abhängigkeitsmengen*beschränkung* auf das Martingal auch eine auf die Strategie impliziert: Denn um die Strategie an der Stelle w zu bestimmen, müssen wir die Werte $d(w1)$, $d(w0)$ und $d(w)$ kennen. Die Abhängigkeitsmenge der Strategie ist also höchstens doppelt so groß, wie die des Martingals.

Beachte jedoch: Wir haben im Falle der Supermartingale keine Schranke mehr auf die Anzahl der Stellen, an denen $\neq \frac{1}{2}$ gewettet wird – anders als bei den fairen Martingalen. Dies hatten wir in Beispiel 12 gesehen.

5.1.3 Martingalfamilien

Da Γ-Martingale und (einzelne) $\mathscr{F}$-Martingale äquivalent sind, gilt die Aussage aus Satz 37 – nämlich, dass die Abhängigkeitsmengen von Strategien in den Abhängigkeitsmengen der zugehörigen Martingale enthalten sind – auch hier.

5.1.4 Σ-Maß

In Beispiel 36 hatten wir gesehen, dass bei beschränkten Abhängigkeitsmengen einer Strategie die Abhängigkeitsmenge des zugehörigen Martingals beliebig groß werden kann. Gerade deshalb war das Σ-Maß auch ein stärkeres Maß als das Γ-Maß. Die Aussage von Satz 37 gilt also für Σ-Maß nicht.

5.2 Martingale und Strategien

5.2.1 Einfache Abhängigkeitsmengenbeschränkung

Wir setzen hier voraus, dass eine polylogarithmische Schranke über die Abhängigkeitsmenge eines Martingals verhängt ist, und zeigen, dass es für Maßbegriffe gleichbedeutend ist, ob man (zusätzlich) eine polylogarithmische Schranke $\mathcal{O}(\log^k |w|)$ über die Laufzeit des Martingal oder der ihm zu Grunde liegenden Strategie verhängt: Die Umrechnung ist in polylogarithmischer Zeit möglich. Die genaue Zeitschranke bleibt dabei jedoch nicht erhalten.

Satz 39. *s sei eine Γ-Strategie mit Schranke $\log^k |w|$, d.h. s lasse sich in Laufzeit $\log^k |w|$ berechnen und die Abhängigkeitsmenge von $d[s]$ sei $\mathcal{O}(\log^k |w|)$-druckbar. Dann hat $d[s]$ Laufzeit $\mathcal{O}(\log^{6k} |w|)$.*

Beweis. Untersuche den Aufwand zur Errechnung eines Martingals aus einer Strategie s mit Schranke $\mathcal{O}(\log^k |w|)$: Um das Martingal zu errechnen, werden die von der Strategie festgelegten Änderungsfaktoren des Kapitals multipliziert.

- Um das Martingal zu berechnen, müssen wir alle in der Vergangenheit aufgetretenen Wetten $\neq \frac{1}{2}$ betrachten, und die entsprechenden Faktoren multiplizieren. Dabei ensteht Aufwand aus zwei Gründen:
  - Die Anzahl der zu betrachteten Stellen. Nach Satz 3 bewirkt die Beschränkung der Abhängigkeitsmenge von $d[s]$ auch eine Beschränkung auf die Anzahl der Stellen mit Wetten $\neq \frac{1}{2}$.

∗ Der Berechnungsaufwand für die Strategien, der an jeder dieser Stellen auftritt. Nach Satz 37 bewirkt die Beschränkung der Abhängigkeitsmenge von $d[s]$ auch eine Beschränkung der Abhängigkeitsmenge der Strategie. Damit ist auch diese in der zulässigen Zeit berechenbar.

Die Abhängigkeitsmengenbeschränkung der Strategie sorgt also dafür, dass jede Stelle der Strategie « leicht » zu berechnen ist; die wenigen Stellen mit « echten Wetten » dafür, dass wir nur « wenige » solche Stellen betrachten müssen.

– Das zweite Problem ist der Berechnungsaufwand: Jede einzelne Multiplikation hat Rechenaufwand $\mathcal{O}(|s(.)|^2)$ (unter Verwendung des « Schulalgorithmus »). Es stellt sich nun die Frage wie man $|s(.)|$ in Abhängigkeit von $|w|$ abschätzen kann.

Im Gegensatz zur üblichen Konvention für Berechenbarkeit durch sublineare Maschinen, wird in den zitierten Artikeln unter « polylogarithmischer Berechenbarkeit » ein Berechnungsmodell verstanden, bei der die Ausgabelänge von der Laufzeit der berechnenden Maschine beschränkt wird. Eine triviale obere Schranke ist somit $\mathcal{O}(\log^k |w|)$, denn da sich die Strategie in dieser Zeit berechnen lässt, kann auch die Kodierungslänge der Ausgabe höchstens so lang sein. Damit bleiben wir bereits im polylogarithmischen Bereich.

Bei dem Maßansatz aus [AlS] können wir also Strategien und Martingale als äquivalent betrachten; allerdings hat das Martingal die folgende (höhere) Schranke für die Laufzeit:

$$\mathcal{O}(\underbrace{\log^k |w|}_{①} \cdot \underbrace{\log^k |w|}_{②} \cdot \underbrace{\log^k |w|}_{③} \cdot \underbrace{\log^{2k} |w|}_{④} \cdot \underbrace{\log^k |w|}_{⑤}) = \mathcal{O}(\log^{6k} |w|)$$

① Berechnungsaufwand einer Stelle der Strategie

② Rekursionsaufwand zur Berechnung einer Stelle der Strategie

③ Aufwand zur Bestimmung der zu berechnenden Stellen

④ Aufwand pro Multiplikation

⑤ Anzahl Multiplikationen

Die Größe der Abhängigkeitsmenge bleibt bei der Umrechnung erhalten, da die Schranke ja nach Definition bereits für $d[s]$ galt. □

Wie bereits erwähnt werden wir im Folgenden so vorgehen, dass wir die Laufzeitschranken über Strategien definieren, wenn wir « schichtweise » Schranken (also Schranken $\mathcal{O}(\log^k |w|)$ für festes k) betrachten wollen. Falls wir nur *irgendeine* polylogarithmische Schranke auferlegen wollen, so werden Strategie- und Martingalschranken gleichwertig.

5.2.2 Quotientenformulierung

Eine Konstruktion analog zu Satz 39 – mit dem Ziel, Supermartingale aus Strategien errechnen zu können –, können wir nicht durchführen, denn im Falle von $\mathcal{Q}$-Martingalen ist, wie wir in Beispiel 12 gesehen haben, nicht sichergestellt, dass nur auf wenigen Stellen gewettet wird. Dies ist auch nicht verwunderlich, denn die Motivation hinter der Einführung der Supermartingale

war ja gerade, häufiger als polynomiell oft wetten zu können, ohne für das Supermartingal zu große Abhängigkeitsmengen zu erhalten; bei der Umrechnung müssten wir also i. Allg. zu viele Strategiefaktoren multiplizieren.

5.2.3 Martingalfamilien

Zu beachten ist hier: Bei Mosers Maßansatz der Martingalfamilien können immer nur *einzelne* der Martingale in einer Familie aus den zugehörigen Strategien errechnet werden. Die Berechnung der gesamten Familie wäre zu aufwändig.

5.2.4 Σ-Maß

Die Umrechnung von Strategie zu Martingal ist in den zulässigen Zeitschranken nicht möglich, da (offensichtlicherweise) sowohl Laufzeit als auch Abhängigkeitsmenge des Martingals zu einer zulässigen Strategie zu groß werden können. Dies sieht man an folgendem einfachen Gegenbeispiel.

Beispiel 40. *Sei $s \equiv 1$. Diese Strategie hat offensichtlich minimale Laufzeit und leere Abhängigkeitsmengen. Das zugehörige Martingal $d[s]$ dagegen muss alle bisherigen Wettergebnisse abfragen, um das aktuelle Kapital ausgeben zu können, so dass sowohl Laufzeit als auch Abhängigkeitsmenge zu groß werden.*

Kapitel 6

Komplexitätsschichten und ihre Eigenschaften

6.1 Konstruktion eines universellen Martingals

6.1.1 Einfache Abhängigkeitsmengenbeschränkung

Leider ist es uns nicht gelungen, zu zeigen, dass es zu gegebenem k ein Γ-Martingal gibt, das die Vereinigung aller Mengen überdeckt, die von Γ-Martingalen mit der Schranke $\log^k |w|$ bedeckt werden. Versucht man, solch ein Martingal zu konstruieren, treten Probleme damit auf, sicherzustellen, dass dieses Martingal an jeder Stelle die Fairnessbedingung erfüllt. Im nächsten Abschnitt werden wir den gleichen Versuch für die Quotientenformulierung unternehmen. Dort steht uns mit der Überdeckung durch nummerierte Aufzählungen ein mächtigeres Werkzeug zur Verfügung.

6.1.2 Quotientenformulierung

Hier wollen wir versuchen, zu zeigen, dass es für ein festes k eine $\mathscr{Q}$-Überdeckung gibt, die die Vereinigung aller Mengen überdeckt, die von $\mathscr{Q}$-Überdeckungen mit der Schranke $\log^k |w|$ bedeckt werden. Ein naheliegender Ansatz wäre es, Martingale zu summieren. Dies bedeutet jedoch einen zu großen Rechenaufwand, da wir – damit die (Ungleichungs-)Fairness des entstehenden Supermartingals sichergestellt ist – die Fairness jedes summierten Supermartingals an allen Stellen überprüfen müssten.

Wir gehen also anders vor: Statt die Supermartingale zu summieren, benutzen wir das Konzept der nummerierten Aufzählung. Mit Hilfe dieses Konzept ist es kein Problem, wenn wir Fehler bezüglich der Fairnessbedingung erst zu spät feststellen – wichtig ist nur, *dass* wir sie feststellen. Genauer:

Satz 41. *Es existiert für jedes $k > 0$ eine nummerierte Aufzählung mit Schranke $\mathcal{O}(\log^{2k+2} |w|)$, welche alle Mengen überdeckt, die von nummerierten Aufzählungen mit $\mathcal{O}(\log^k |w|)$-$\mathscr{Q}$-Martingalen überdeckt werden.*

Beweis. Wir benutzen eine universelle Maschine. Diese erhält als Eingabe einen Binärcode und intepretiert ihn als ein Paar aus einer Konstante c und der Kodierung einer Maschine, die ein Supermartingal berechnet. Die Laufzeit der simulierten Maschine wird wie gewünscht

mit $c \cdot \log^k |w|$ beschränkt (der Vorfaktor ist erforderlich, da alle unsere Zeitschranken von der Form $\mathcal{O}(...)$ waren). Es wird immer auf den ersten $\log |w|$ Präfixen von w geprüft, ob das Supermartingal dort (Ungleichungs-)fair ist. Falls nicht, gibt die Maschine sofort 0 aus. Dadurch wird eine Normierung der Funktion zu einem Supermartingal erreicht.

Außerdem müssen wir sicherstellen, dass die simulierte Funktion zulässige Abhängigkeitsmengen hat. Dazu berechnen wir $d(v)$ für alle v mit $|v| \leq \log\log|w|$. Dies sind $\mathcal{O}(2^{\log\log|w|}) = \mathcal{O}(\log|w|)$ viele Berechnungen; für jede davon wird die Laufzeit mit $c \cdot \log^k |v| = c \cdot \log^k \big(\log\log|w|\big)$ beschränkt. Die Berechnungen dauern also insgesamt höchstens

$$\mathcal{O}(\log|w| \cdot c \cdot \log^k \log\log|w|) \subseteq \mathcal{O}(c \cdot \log^{k+1}|w|)$$

Falls bei der Berechnung eines der $d(v)$ mehr als $c \cdot \log^k |v| = c \cdot \log^k \big(\log\log|w|\big)$ viele Bits abgefragt werden, so ist die Abhängigkeitsmenge für dieses v zu groß. Gib sofort 0 aus.

Die konstruierte Maschine stellt gerade eine nummerierte Aufzählung dar. Nur Ungleichungsfaire Supermartingale können dabei Mengen bedecken, denn die unfairen werden (wenn auch mit Verzögerung) auf « 0 » gesetzt. Die gelisteten Supermartingale bedecken auf jeden Fall irgendeine unter Quotienten abgeschlossene Menge (im Zweifelsfall $\emptyset$), sind also zulässig. Alle Supermartingale mit der gewünschten Zeitschranke werden aufgezählt. Wir bedecken also alle Mengen, die von Supermartingalen mit Schranke $\log^k |w|$ bedeckt werden.

Die Laufzeit der nummerierten Aufzählung ist im Wesentlichen bestimmt durch den quadratischen Zeitverlust, der durch die Verwendung der universellen Maschine entsteht. Unsere nummerierte Aufzählung hat also die Schranke $\log^{2k+2}|w|$. □

Zu dieser nummerierten Aufzählung werden wir nun eine zufällige Sprache definieren. Wir gehen dabei analog vor zu Satz 18:

Satz 42. *Es existiert für jedes $k > 0$ eine Sprache $A \in \mathbf{DTIME}(n^{4k+4})$, die von keinem $\mathcal{O}(\log^k |w|)$-$\mathcal{Q}$-Martingal überdeckt wird.*

Beweis. Sei $\{(A_i, d_i)\}$ unsere nummerierte Aufzählung aus Satz 41, wobei die A_i jeweils als die größtmöglichen Mengen gewählt seien, so dass A_i von d_i bedeckt wird, und so dass $\{(A_i, d_i)\}$ eine nummerierte Aufzählung ist. Solche größten A_i existieren, da die unter Quotienten abgeschlossenen Mengen gegen Vereinigung abgeschlossen sind. Nach der Konstruktion aus Satz 41 hat $\{(A_i, d_i)\}$ dann die Zeitschranke $\log^{2k+2}|w|$.

Wähle für jedes d_i durch Diagonalisierung eine Sprache $L_i \in \mathbf{DTIME}(\log^{4k+4}|w|)$, die nicht von d_i bedeckt wird (die Schranke gilt, denn die Rekursion, um gegen das Martingal zu diagonalisieren, dauert genau so lange). Setze

$$L = \bigotimes_i L_i := \{x10^{i-1} \mid x \in L_i\}$$

L ist in $\mathbf{DTIME}(|x|^{4k+4})$, denn:

> Wenn $L(y)$ für ein Wort y berechnet werden soll, gehe wie folgt vor: Prüfe zunächst, ob y von der Gestalt $x10^{i-1}$ ist. Falls nicht, verwerfe. Falls ja, so bestimme erst die Länge des Suffixes der Form 10^{i-1}. Dies dauert linear lang, ist also vernachlässigbar. Der Aufwand, um dann $L_i(x)$ zu errechnen ist dann – wie gerade gesehen – $\mathcal{O}(\log^{4k+4}|w|) = \mathcal{O}(|x|^{4k+4})$ (dank der Uniformität der nummerierten Aufzählung).

Es bleibt die Zufälligkeit zu zeigen: $\forall i : L/10^{i-1} = L_i \notin A_i \implies \forall i : L \notin A_i$, denn A_i war unter Quotienten abgeschlossen. Also wird L von keinem der d_i bedeckt, und ist somit bzgl. der betrachteten « universellen » nummerierten Aufzählung zufällig, also auch bezüglich aller nummerierten Aufzählungen mit Schranke $\log^k |w|$. □

6.1.3 Martingalfamilien

Versucht man, für Martingalfamilien universelle Überdeckungen zu konstruieren, so stößt man auf Probleme. Würden wir versuchen, eine universelle Martingalfamilie zu konstruieren, so könnten wir transitiv einen einzigen großen Zuständigkeitsbereich $\{0,1\}^*$ erhalten, was unzulässig wäre. Die Konstruktion einer universellen Martingalfamilie erscheint also (im Allgemeinen) unmöglich.

6.1.4 Σ-Maß

Es ist völlig unklar, wie man für Σ-Maß eine universelle Überdeckung konstruieren soll: Würde man die zu den Strategien zugehörigen Martingale summieren, so hätte das summierte Martingal große Abhängigkeitsmengen.

Würde man stattdessen über die Strategien vorgehen, ist unklar, welche Rechenoperation man auf sie anwenden sollte: Denn Martingale haben den Vorteil, dass sie, wenn sie ungeeignet sind, eine bestimmte Folge zu bedecken, in einer Summe zumindest keinen „schädlichen“ Beitrag leisten – sie werden eben einfach den Wert 0 annehmen. Deswegen bietet sich die Summation für Martingale an. Strategien dagegen kosten i. Allg. Kapital, wenn Sie falsch wetten; betrachte z.B. folgendes einfache Beispiel.

Beispiel 43. *Es seien*

$$s_0(w) := \begin{cases} 1 & \text{falls } |w| \equiv 0 \pmod 2 \\ \frac{1}{2} & \text{sonst} \end{cases} \quad \textit{und} \quad s_1(w) := \begin{cases} 0 & \text{falls } |w| \equiv 1 \pmod 2 \\ \frac{1}{2} & \text{sonst} \end{cases}$$

Dann überdeckt s_0 die Folge 0^∞ und s_1 die Folge 1^∞. Eine beliebige Konvexkombination von s_0 und s_1 kann jedoch nur entweder 0^∞ oder 1^∞ überdecken.

Eine Konvexkombination von Strategien eignet sich a priori also nicht um eine universelle Überdeckung zu erhalten.

6.2 Definitionen für schichtweise Zufälligkeit

6.2.1 Einfache Abhängigkeitsmengenbeschränkung

Definition 44. *Eine Sprache A sei $\log^k n$-Γ-zufällig, falls keine Γ-Strategie $s \in \mathbf{DTIME}(\log^k n)$ existiert, so dass A von $d[s]$ bedeckt wird und die Abhängigkeitsmenge von $d[s]$ $\mathcal{O}(\log^k n)$-druckbar ist.*

6.2.2 Quotientenformulierung

Definition 45. *Eine Sprache A sei $\log^k n$-$\mathcal{Q}$-zufällig, falls kein $\mathcal{O}(\log^k n)$-Abhängigkeitsmengenbeschränktes $\mathcal{Q}$-Martingal $d \in \mathbf{DTIME}(\log^k n)$ existiert, so dass d eine unter Quotienten abgeschlossene Menge C mit $A \in C$ bedeckt.*

6.2.3 Martingalfamilien

Definition 46. *Eine Sprache A sei $\log^k n$-$\mathcal{F}$-zufällig, falls keine $\mathbf{P}$-Martingalfamilie existiert, die A bedeckt und für deren Mitglieder $\{d_i\}_i$ und Zuständigkeitsbereiche $\{Q_i\}_i$ gilt:*

$$\big(\exists s_i\ \mathcal{F}\text{-Strategie} : s_i \in \mathbf{DTIME}(\log^k n) \wedge d_i = d[s_i]\big) \wedge \big(Q_i(n) \text{ ist } \mathcal{O}(\log^k n)\text{-druckbar}\big)$$

6.2.4 Σ-Maß

Definition 47. *Eine Sprache A sei $\log^k n$-Σ-zufällig, falls keine Σ-Strategie $s \in \mathbf{DTIME}(\log^k n)$ existiert, so dass $d[s]$ A bedeckt und die Abhängigkeitsmenge von s $\mathcal{O}(\log^k n)$-druckbar ist.*

6.3 Maß und Zufälligkeit

6.3.1 Einfache Abhängigkeitsmengenbeschränkung

Satz 48. *Es gilt*

$$\mu^{\Gamma}(\mathbf{C}) = 0 \implies \exists k : \mathbf{RAND}^{\Gamma}(\log^k n) \cap \mathbf{C} = \varnothing$$

Beweis. Der Beweis ist trivial: $\mu^{\Gamma}(\mathbf{C}) = 0 \Longrightarrow \exists(k,d) : d \in \mathbf{DTIME}(\log^k n) \wedge \mathbf{C} \subseteq S^{\infty}[d] \Longrightarrow \exists(k,s) : s \in \mathbf{DTIME}(\log^k n) \wedge \mathbf{C} \subseteq S^{\infty}[d[s]] \Longrightarrow \mathbf{C}$ kann keine $\log^k n$-Γ-zufällige Sprache enthalten. □

Die Umkehrung der Implikation aus Satz 47 zu zeigen war uns dagegen nicht möglich, da wir in unseren Untersuchungen zu Abschnitt 6.1.1 keinen Erfolg bei der Konstruktion eines universellen Γ-Martingals hatten.

6.3.2 Quotientenformulierung

Satz 49. *Es gilt*

$$\mu^{\mathcal{Q}}(\mathbf{C}) = 0 \Leftrightarrow \exists k : \mathbf{RAND}^{\mathcal{Q}}(\log^k n) \cap \mathbf{C} = \varnothing$$

Beweis. Vorwärtsrichtung: $\mu^{\mathcal{Q}}(\mathbf{C}) = 0 \Longrightarrow \exists(k, M, \{\mathbf{C}_i\}) : M(i,w) \in \mathbf{DTIME}((\log|w| + i)^k) \wedge \forall i\, \mathbf{C}_i \in S^{\infty}[M(i,.)] \wedge \mathbf{C} \subseteq \bigcup \mathbf{C}_i$. Da i für jedes einzelne der durch $M(i,.)$ berechneten $\mathcal{Q}$-Martingale d_i konstant ist, läuft jedes dieser Martingale in $\mathcal{O}(\log^k |w|)$, also kann in keinem der $\mathbf{C}_i$ eine $\log^k |w|$-$\mathcal{Q}$-zufällige Sprache enthalten sein, also auch nicht in $\mathbf{C}$.

Rückrichtung: $\mathbf{RAND}^{\mathscr{Q}}(\log^k n) \cap \mathbf{C} = \varnothing \Longrightarrow \forall A \in \mathbf{C} \quad \mu^{\mathscr{Q}}_{\log^k n}(\{A\}) = 0$. Wir konstruieren nun nach Satz 41 eine nummerierte Aufzählung für die die Singletons bedeckenden Martingale, welche dann in Laufzeit $\mathcal{O}(\log^{2k+2} |w|)$ läuft, und $\mu^{\mathscr{Q}}(\mathbf{C}) = 0$ bezeugt. □

6.3.3 Martingalfamilien

Satz 50. *Es gilt*

$$\mu^{\mathscr{F}}(\mathbf{C}) = 0 \implies \exists k : \mathbf{RAND}^{\mathscr{F}}(\log^k n) \cap \mathbf{C} = \varnothing$$

Beweis. Sei $\mathbf{F}$ eine bedeckende $\mathbf{P}$-Martingalfamilie. Nach Mosers Definition von Martingalfamilien gibt es eine Maschine M_1, die alle Martingale in $\mathbf{F}$ mit einer gemeinsamen Zeitschranke $\mathcal{O}(\log^{k_1} |w|)$ berechnet. Eine weitere solche Maschine berechnet alle Zuständigkeitsbereiche Q_i, sagen wir in Zeitschranke $\mathcal{O}(\log^{k_2} |w|)$. Setze $k := \max\{k_1, k_2\}$, dann haben wir eine Martingalfamilie mit Schranke $\mathcal{O}(\log^k |w|)$ gefunden, die also die Konklusion bezeugt. □

Die Rückrichtung zu zeigen dürfte dagegen schwierig werden, da die Diskussion in 6.1.3 zeigt, dass nicht damit zu rechnen ist, dass die Konstruktion universeller Martingalfamilien möglich ist.

6.3.4 Σ-Maß

Satz 51. *Es gilt*

$$\mu^{\Sigma}(\mathbf{C}) = 0 \implies \exists k : \mathbf{RAND}^{\Sigma}(\log^k n) \cap \mathbf{C} = \varnothing$$

Beweis. Der Beweis ist trivial: $\mu^{\Sigma}(\mathbf{C}) = 0 \Longrightarrow \exists (k, s) : s \in \mathbf{DTIME}(\log^k n) \wedge \mathbf{C} \subseteq S^{\infty}[d[s]] \Longrightarrow$ $\mathbf{C}$ kann keine $\log^k n$-Σ-zufällige Sprache enthalten. □

Die Rückrichtung ist unklar, da wir für den Strategieansatz ja kein Konzept einer universellen Überdeckung finden konnten. Da dieser Ansatz ein allgemeinerer Ansatz ist, als es die Martingalfamilien sind, passt es durchaus ins Bild, dass wir hier die gleichen Probleme haben, wie bei den Martingalfamilien.

6.4 Zeitkomplexität zufälliger Sprachen I

6.4.1 Einfache Abhängigkeitsmengenbeschränkung

Satz 52. *Es gilt*

$$\mathbf{DTIME}(n^k) \cap \mathbf{RAND}^{\Gamma}(\log^k n) = \varnothing$$

Beweis. Als Vorüberlegung nehme an, $A \in \mathbf{DTIME}(\log^k(2^n - 1))$.
Dann definiere eine Strategie wie folgt:

$$s(X \upharpoonright x) = \begin{cases} 1 - A(x) & \text{falls } x \in 0^* \\ \frac{1}{2} & \text{sonst} \end{cases}$$

$\Longrightarrow s(X \upharpoonright x)$ lässt sich in $\mathcal{O}(\log^k(2^{|x|}-1)) = \mathcal{O}(\log^k(2^{\log|X\upharpoonright x|}-1)) = \mathcal{O}(\log^k(|X \upharpoonright x|))$ berechnen.
$\Longrightarrow \exists\mathcal{O}(\log^k|w|)$-Strategie, die auf A Erfolg hat (die Strategie ist zulässig, denn die Abhängigkeitsmenge des induzierten Martingals umfasst nur 0^* und ist damit leicht polylog-druckbar).
Sei nun A $\log^k n$-Γ-zufällig. $\xrightarrow{\text{Vorüberlegung}} A \notin \mathbf{DTIME}(\log^k(2^n-1)) = \mathbf{DTIME}(n^k)$. □

6.4.2 Quotientenformulierung

Wir würden gerne auch für die Quotientenformulierung ein Ergebnis erhalten, das analog zu der Aussage von Satz 52 ist. Um dies zu zeigen, eignet sich der dort gewählte Ansatz nicht, denn wir können nicht einfach eine Strategie definieren, die zu einer bestimmten Sprache passt – schließlich müssen wir nicht nur auf dieser Sprache Erfolg haben, sondern auch auf allen ihren Quotienten, damit eine Überdeckung überhaupt zulässig ist. Wir wählen deshalb einen anderen Ansatz: Wir zeigen zunächst, dass zu einer $\mathbf{DTIME}(n^k)$-Sprache auch alle Quotienten in $\mathbf{DTIME}(n^k)$ sind, und kontruieren dann eine Strategie, die auf allen n^k-Sprachen Erfolg hat.

Sei M eine Maschine, die $A \in \mathbf{DTIME}(\log^k(2^n-1))$ in der angegebenen Zeitschranke erkennt. Wir können nun wie folgt für jedes y eine Maschine M_y konstruieren, die A/y in der gleichen Zeitschranke erkennt: M_y läuft zunächst ans Ende seiner Eingabe, schreibt an das Ende das Anhängsel y, läuft wieder zum Anfang und verhält sich wie M. Der Zeitaufwand für diesen « Vorspann » ist linear in der Eingabelänge, also vernachlässigbar. Die restliche Laufzeit von M_y ist durch $\mathcal{O}((|x|+|y|)^k) \subseteq \mathcal{O}(|x|^k)$ beschränkt. Somit folgt für alle Quotienten A/y von A: $A/y \in \mathbf{DTIME}(\log^k(2^n-1))$.

Es sei nun eine Aufzählung $M_1, M_2, ...$ von Turingmaschinen gegeben, die alle Maschinen enthält, die Sprachen erkennen und die in $\mathcal{O}(|x|^k)$ laufen. Diese erhält man z.B. indem man alle Binärwörter nacheinander aufzählt, sie als Kodierungen für Turingmaschinen interpretiert, deren Laufzeit durch $|x|\cdot|x|^k$ beschränkt, und sie simuliert. Durch die Simulation entsteht nochmals quadratischer Zeitverlust.

Es sei nun eine Eingabe w gegeben, und wir wollen dazu bestimmen, welchen Anteil $s(w)$ wir auf das nächste Bit x_j wetten. Setze $l = 1$. Für $0 \leq j \leq \lfloor\log|w|\rfloor$ betrachten wir jeweils das Wort u mit $\text{pos}(u) = j$ und prüfen ob $M_l(u)$ mit $w[j]$ übereinstimmt.

Falls ja, so haben wir keinen Grund M_l zu verwerfen, betrachten also im nächsten Schritt Stelle $j+1$ weiterhin mit Maschine M_l.

Falls nein, so ist M_l nicht geeignet, um die Eingabe w zu beschreiben, und wir betrachten im nächsten Schritt Stelle $j+1$ mit der nächsten Maschine. Setze also $l := l+1$.

Sobald wir alle $0 \leq j \leq \lfloor\log|w|\rfloor$ durchlaufen haben, haben wir höchstens $\lfloor\log|w|\rfloor$ Maschinen als solche erkannt, die sich auf unserer Eingabe w schlecht verhalten. Es bleibt eine Maschine M_l übrig, die – nach unserem momentanen Informationsstand – sehr wohl eine Beschreibung für w liefern könnte. Es kann durchaus sein, dass auch M_l sich auf w schlecht verhält, dies überprüfen wir aber um den Rechenaufwand zu begrenzen nicht mehr.[1]

Nun müssen wir noch den Einsatz ausgeben. Es sei

$$s(X \upharpoonright x) := \begin{cases} \frac{1}{2} & \text{falls } x \notin 0^* \\ \frac{1}{4} & \text{falls } x \in 0^* \wedge M_l(x) = 1 \\ \frac{3}{4} & \text{falls } x \in 0^* \wedge M_l(x) = 0 \end{cases}$$

Auf diese Weise ist sichergestellt, dass wir nie das gesamte Kapital verlieren, und – da wir, falls

w (im Limes) eine $\mathbf{DTIME}(n^k)$-Sprache beschreibt, irgendwann die „richtige“ Maschine M_l erreichen werden – erwirtschaften wir unendlichen Gewinn auf allen $\mathbf{DTIME}(n^k)$-Sprachen.

Die Strategie hat Laufzeit $\mathcal{O}(\log|w| \cdot \log^2|w| \cdot \log^{2k}|w|)$, die Abhängigkeitsmengen des Martingals sind in 0^* enthalten. Da es eine faire Strategie ist, ist die Umrechenbarkeit in ein Martingal kein Problem (es ergibt sich nach Satz 39 die Laufzeit $\log^{12k+18}|w|$), und nach Konstruktion werden zu jeder Sprache auch alle Quotienten bedeckt.

Nun können wir den selben Schluß anwenden, wie im Beweis von Satz 52.

Satz 53. *Es gilt*

$$\mathbf{DTIME}(n^k) \cap \mathbf{RAND}^{\mathscr{Q}}(\log^{12k+18} n) = \varnothing$$

Beweis. Für A mit der Eigenschaft $\log^{12k+18}|w|$-$\mathscr{Q}$-zufällig gilt $A \notin \mathbf{DTIME}(n^k)$ nach der Vorüberlegung. □

6.4.3 Martingalfamilien

Satz 54. *Es gilt*

$$\mathbf{DTIME}(n^k) \cap \mathbf{RAND}^{\mathscr{F}}(\log^k n) = \varnothing$$

Beweis. Hier funktioniert der Beweisansatz wie in Satz 52, denn wir können die konstruierte $\mathcal{O}(\log^k|w|)$-Strategie direkt in eine Martingalfamilie hinüberretten: Dazu kommt jedes einzelne Wort x in einen eigenen Zuständigkeitsbereich $Q_{\mathrm{pos}(x)}$ und das zugehörige Ratenmartingal $D_{\mathrm{pos}(x)}$ ist überall 1, außer auf x, wo wir setzen $D_{\mathrm{pos}(x)}((X \upharpoonright x)0) := 2 \cdot s(X \upharpoonright x)$ sowie $D_{\mathrm{pos}(x)}((X \upharpoonright x)1) := 2 \cdot (1 - s(X \upharpoonright x))$. □

6.4.4 Σ-Maß

Satz 55. *Es gilt*

$$\mathbf{DTIME}(n^k) \cap \mathbf{RAND}^{\Sigma}(\log^k n) = \varnothing$$

Beweis. Als Vorüberlegung nehme an, $A \in \mathbf{DTIME}(\log^k(2^n - 1))$.

Dann definiere eine Strategie wie folgt:

$$s(X \upharpoonright x) = \begin{cases} 1 - A(x) & \text{falls } x \in 0^* \\ \frac{1}{2} & \text{sonst} \end{cases}$$

$\Longrightarrow s(X \upharpoonright x)$ lässt sich in $\mathcal{O}(\log^k(2^{|x|} - 1)) = \mathcal{O}(\log^k(2^{\log|X\upharpoonright|x|} - 1)) = \mathcal{O}(\log^k(|X \upharpoonright x|))$ berechnen. $\Longrightarrow \exists\mathcal{O}(\log^k|w|)$-Strategie, die auf A Erfolg hat (die Strategie ist zulässig, denn ihre Abhängigkeitsmenge umfasst immer nur das aktuelle Bit und ist damit leicht polylog-druckbar).

Sei nun A $\log^k n$-Σ-zufällig. $\xrightarrow{\text{Vorüberlegung}}$ $A \notin \mathbf{DTIME}(\log^k(2^n - 1)) = \mathbf{DTIME}(n^k)$. □

[1] Es ist also nicht so, dass wir für die Eingabe w so lange nach einer Maschine suchen, bis wir eine gefunden haben, die sich zumindest auf w gut verhält. Jedoch ist es so, dass wir – mit Verspätung – jede « schlechte » Maschine aussortieren, und irgendwann die richtige finden werden.

6.5 Zeitkomplexität zufälliger Sprachen II

6.5.1 Einfache Abhängigkeitsmengenbeschränkung

Es ist uns nicht gelungen, zu zeigen, dass für jedes k sichergestellt ist, dass wir eine Sprache $L \in \mathbf{DTIME}(n^{g(k)})$, mit $g(k)$ nur von k abhängig, finden können, so dass sich L nicht von $\log^k |w|$-Martingalen bedecken lässt. Dass uns dies nicht möglich war, resultiert daraus, dass wir in Abschnitt 6.1.1 keinen Erfolg bei der Konstruktion eines universellen Γ-Martingals hatten, so dass auch die Konstruktion einer entsprechenden zufälligen Sprache nicht möglich erschien.

6.5.2 Quotientenformulierung

Satz 56. *Es gilt*

$$\mathbf{DTIME}(n^{4k+4}) \cap \mathbf{RAND}^{\mathcal{Q}}(\log^k n) \neq \varnothing$$

Beweis. Satz 42 liefert die gewünschte zufällige Sprache. □

6.5.3 Martingalfamilien

Eine Aussage analog zu Satz 56 für Martingalfamilien zu zeigen, dürfte schwierig sein. Dies liegt daran, dass die Diskussion in Abschnitt 6.1.3 zeigt, dass nicht damit zu rechnen ist, dass die Konstruktion universeller Martingalfamilien möglich ist, so dass auch die Konstruktion einer zufälligen Sprache schwierig wird.

6.5.4 Σ-Maß

Eine Aussage analog zu Satz 56 für Σ-Strategien zu zeigen, dürfte schwierig sein. Dies liegt daran, dass – wie in Abschnitt 6.1.4 beschrieben – unklar ist, wie man bei der Konstruktion einer universellen Σ-Strategie vorgehen sollte. Dies macht auch die Konstruktion einer zufälligen Sprache schwierig.

6.6 $\leq_{\mathrm{m}}^{\mathrm{polylog}}$-Reduzierbarkeit

Wir definieren nun ein Analogon zur $\leq_{\mathrm{m}}^{\mathrm{p}}$-Reduzierbarkeit für den Fall polylogarithmischer Zeitschranken. Dabei müssen wir einige Eigenheiten in Kauf nehmen.

Definition 57. *Wir sagen, es gelte $A \leq_{\mathrm{m}}^{\mathrm{polylog}} B$ genau dann, wenn es eine Reduktionsfunktion f gibt, derart dass*

- *Es existiert eine Turingmaschine M, die f in polylogarithmischer Zeit relativ zur Eingabelänge $|x|$ berechnet.*
- *Als tatsächliche Eingabe erhält die Maschine M statt des Wortes x nur $|x|$ (logarithmisch groß kodiert).*
- *Um auf die einzelnen Bits von x zuzugreifen, schreibt die Maschine M eine « Adresse » auf ein Orakelband, und erhalt als Rückgabe das dort stehende Bit von x.*

– *Der Funktionswert wird* nicht *auf ein Ausgabeband geschrieben, stattdessen geht man wie folgt vor: Der Maschine M werden zusätzlich zu $|x|$ als Eingabe und x als Orakel noch zwei Werte i und b übergeben. Die Maschine akzeptiert die Eingabe, falls der zu berechnende Funktionswert an Adresse i das Bit b hat. Die Maschine kann also auch sehr lange, aber dafür einfach aufgebaute Wörter « berechnen ».*

– *Damit diese Reduktion ein Spezialfall der $\leq^{\mathrm{p}}_{\mathrm{m}}$-Reduktion bleibt, beschränken wir die Ausgabelänge zusätzlich polynomiell (nicht polylogarithmisch!). Dies ist erforderlich, da $|i| \approx \log n$, so dass die « Ausgabe » ohne diese zusätzliche Einschränkung z.B. die Länge $n^{\log n}$ erreichen könnte.*

6.7 Konstruktion zufälliger Sprachen

6.7.1 Einfache Abhängigkeitsmengenbeschränkung

Satz 58. *Es gilt*

$$A \ \log^2 n\text{-}\Gamma\text{-zufällig} \Rightarrow \forall k \geq 2 : \exists A_k : A_k \ \log^k n\text{-}\Gamma\text{-zufällig} \wedge A_k \leq^{\mathrm{polylog}}_{\mathrm{m}} A$$

Beweis. Wir versuchen, eine analoge Konstruktion zu der im Beweis von Satz 6.14 bei Ambos-Spies und Mayordomo [ASMa] durchzuführen. Das heißt, wir wählen einen Teil der Wörter von A aus, deren « Informationsgehalt » relativ gering ist, und « komprimieren » diese Information in kürzere Wörter. Wir argumentieren dann, dass die so entstehenden Sprachen A_k $\log^k |w|$-zufällig sein müssen. Denn: Wären sie es nicht, so gäbe es eine $\mathcal{O}(\log^k |w|)$-Γ-Strategie, die Erfolg auf A_k hat, und wir könnten daraus sehr einfach eine $\mathcal{O}(\log^2 |w|)$-Strategie konstruieren, die auf A Erfolg hat, was im Widerspruch zur Annahme stünde.

Durch die « Kompression » der Wörter wird die Komplexität der Sprache relativ zur Wortlänge größer, dies erlaubt die Konstruktion der erwähnten Strategie. Zunächst überlegen wir, wie stark wir die « Information » in A komprimieren müssen, damit wir den Sprung von $\log^k |w|$ zu $\log^2 |w|$ schaffen. Wir betrachten dazu die folgenden Formeln mit der gesuchten neuen Länge $|g_k(x)|$:

$$\begin{aligned} \log^k(2^{|x|}) &\leq \log^2(2^{|g_k(x)|}) && (6.1)\\ &\Updownarrow \\ |x|^k &\leq |g_k(x)|^2 && (6.2) \end{aligned}$$

Damit die Konstruktion analog zum exponentiellen Fall funktioniert, muss die unten stehende Bedingung erfüllt werden, z.B. indem wir für alle k

$$g_k(x) := 0^{|x|^k} x$$

wählen und die Mengen A_k wie folgt:

$$A_k := \{x \mid g_k(x) \in A\}$$

Wir nehmen nun an, es gebe eine $\mathcal{O}(\log^k n)$-Strategie s, die A_k bedeckt. Die neue Strategie zu definieren, die A bedeckt, ist dann sehr einfach:

$$\widehat{s}(X \upharpoonright x) = \begin{cases} s(X[g_k(\lambda)]\, X[g_k(0)] \ \ldots\ X[g_k(s_{\mathrm{pos}(y)-1})]) & \text{falls } \exists y : x = g_k(y) \\ \frac{1}{2} & \text{sonst} \end{cases}$$

wobei s_i für jedes i das i-te Wort in der längenaufsteigenden, lexikographischen Ordnung aller Wörter bezeichne.

Dass die Schranke $\log^2 |w|$ für die Laufzeit gilt, ist klar. Durch die Konstruktion aus [ASMa] bleiben die Abhängigkeitsmengen « gleich groß », werden aber zu größeren Wortlängen « verschoben »; relativ dazu werden sie also (in Bezug auf ihre Kardinalität) kleiner. Jedoch ist die Länge jedes Elements *in* den Abhängigkeitsmengen durch die zusätzlichen Nullen deutlich gewachsen, und wir müssen zeigen, dass die Abhängigkeitsmengen trotzdem weiter polylog-druckbar sind und die Schranke $\log^2 |w|$ dazu ausreicht. Da der Aufwand zur Berechnung des Wortteils x von $0^{|x|^k}x$ gleich groß bleibt, und nur genau die Ausgabe der $|x|^k$ Nullen für *jedes* der höchstens $\mathcal{O}\big(\log^k(2^{|x|})\big)$ Elemente in der Abhängigkeitsmenge zusätzlich Laufzeit verbraucht, liefert folgende Rechnung das gewünschte Ergebnis.

$$\mathcal{O}\big(\log^k(2^{|x|})\big) + \underbrace{|x|^k \cdot \mathcal{O}\big(\log^k(2^{|x|})\big)}_{\text{Zusätzlicher Aufwand}} = \mathcal{O}\big(|x|^{2k}\big) = \mathcal{O}\big(\log^2(2^{|x|^k+|x|})\big)$$

Es bleiben die Reduktionen zu zeigen. Das oben angegebene g_k ist die gesuchte Reduktionsfunktion, die nun in polylogarithmischer Zeit berechenbar sein soll. Der Ausgabewert der Funktion ist zwar polynomiell lang. Aber dies ist nach oben gewählter Definition für polylogarithmische Berechenbarkeit kein Problem, da nur einzelne Bits akzeptiert werden müssen. Ein geeignetes Berechnungsverfahren für g_k sieht so aus:

- Der Maschine zur Berechnung von g_k wird als Eingabe $(|x|, i, b)$ und als Orakel x übergeben.
- Es muss nun aus $|x|$ der Wert $|x|^k$ berechnet werden. Für die Kodierungslänge $||x||$ von $|x|$ gilt: $||x|| \approx \log(|x|)$; $k-1$ Multiplikationen einer solchen Zahl lassen sich dann in

  $$\mathcal{O}\left(\underbrace{\sum_{i=1}^{k-1} i^2}_{\text{konstant}} \cdot \log^2 |x|\right)$$

  Schritten durchführen.
- Nun muss $i \leq |x|^k$ überprüft werden. Da

  $$|i| \leq ||0^{|x|^k}x|| = ||x|^{k+1}| \approx \log(|x|^{k+1}) = (k+1) \cdot \log(|x|)$$

  sowie

  $$||x|^k| \approx \log(|x|^k) = k \cdot \log|x|$$

  gilt, ist auch dies in polylogarithmischer Zeit möglich (man vergleiche zunächst die Länge von i und $|x|^k$, und – falls diese gleich ist – dann die beiden Zahlen bitweise, am Anfang beginnend, bis ggfs. ein Unterschied auftritt).

  * Falls $i \leq |x|^k$: Akzeptiere falls $b = 0$, verwerfe sonst.
  * Falls $i > |x|^k$: Berechne $i - |x|^k$. Da nach obigen Überlegungen$|i|$ und $||x|^k|$ polylogarithmisch in $|x|$ sind, ist dies in polylogarithmischer Zeit möglich. Übergib $i - |x|^k$ an das Orakel und vergleiche den Rückgabewert mit b. Bei Gleichheit akzeptiere, bei Ungleichheit verwerfe.

□

6.7.2 Quotientenformulierung

Es ist uns nicht gelungen ein Äquivalent des Satzes 58 für die Quotientenformulierung zu zeigen. Die Schwierigkeit bestand darin, aus einer $\log^k n$-Überdeckung eine $\log^{\text{konst}} n$-Überdeckung zu konstruieren und dabei Verträglichkeit mit Quotienten sicherzustellen.

6.7.3 Martingalfamilien

Satz 59. *Es gilt*

$$A \ \log^2 n\text{-}\mathscr{F}\text{-zufällig} \Rightarrow \forall k \geq 2 : \exists A_k : A_k \ \log^k n\text{-}\mathscr{F}\text{-zufällig} \wedge A_k \leq_{\text{m}}^{\text{polylog}} A$$

Beweis. Wir möchten den selben Widerspruchsbeweis verwenden, wie im Beweis von Satz 58; dazu seien die Funktionen g_k und die Mengen A_k wie dort definiert. Wir müssen nun zeigen, dass es unter der Annahme, es gebe eine $\log^k |w|$-Überdeckung für A_k, auch eine $\log^2 |w|$-Überdeckung für A gibt.

Sei $F_k = (Q_i^{(k)}, D_j^{(k)}, \text{ind}_k)$ die Martingalfamilie mit Schranke $\mathcal{O}(\log^k |w|)$ die A_k bedeckt. Wir definieren dann die Martingalfamilie $F = (Q_i, D_j, \text{ind})$, die A bedeckt, wie folgt:

Sei $Q_i(n^k + n) := \{g_k(x) \,|\, x \in Q_i^{(k)}(n)\}$, und der verbleibende « Platz » werde mit beliebigen weiteren, ausreichend kleinen Zuständigkeitsbereichen R_i gefüllt. Also ist

$$\text{ind}(y) = \begin{cases} (\text{« } Q \text{ »}, \text{ind}_k(x)) & \text{falls } \exists x : y = g_k(x) \\ (\text{« } R \text{ »}, c) & \text{für geeignetes } c \text{ sonst} \end{cases}$$

(wir schreiben der Einfachheit halber die Indizes der Zuständigkeitsbereiche in dieser zweistelligen Form, natürlich könnte man sie bei Bedarf wie üblich einstellig durchnummerieren).

Es gilt dann $\#Q_i(n^k + n) = \#Q_i^{(k)}(n)$. Jedes Element $0^{|x|^k} x \in Q_i$ ist aber verglichen mit x deutlich länger geworden, und zwar so, dass $|x|^k$ zusätzliche Zeichen (die Nullen) ausgegeben werden müssen. Bleiben die Zuständigkeitsbereiche damit noch polylog-druckbar und reicht die Schranke $\log^2 |w|$ noch aus? Ja, denn der Aufwand zur Ausgabe des Wortteils x von $0^{|x|^k} x$ bleibt gleich groß, es kommt nur genau der Aufwand zur Ausgabe der $|x|^k$ Nullen für jedes der höchstens $\mathcal{O}\big(\log^k(2^{|x|})\big)$ Elemente in der Abhängigkeitsmenge dazu, und somit liefert folgende Rechnung das gewünschte Ergebnis.

$$\mathcal{O}\big(\log^k(2^{|x|})\big) + \underbrace{|x|^k \cdot \mathcal{O}\big(\log^k(2^{|x|})\big)}_{\text{Zusätzlicher Aufwand}} = \mathcal{O}\big(|x|^{2k}\big) = \mathcal{O}\big(\log^2(2^{|x|^k + |x|})\big)$$

Die Ratenmartingale werden als

$$D_{\text{ind}(y)}(X \upharpoonright x) = \begin{cases} D^{(k)}_{\text{ind}_k(x)}(X[g_k(\lambda)]\, X[g_k(0)]\, \ldots\, X[g_k(s_{\text{pos}(y)-1})]) & \text{falls } \exists y : x = g_k(y) \\ 1 & \text{sonst} \end{cases}$$

definiert, wobei s_i für jedes i das i-te Wort in der längenaufsteigenden, lexikographischen Ordnung aller Wörter bezeichne.

Das selbe Argument wie in Satz 58 zeigt, dass die Zeitschranke $\mathcal{O}(\log^2 |w|)$ dann ausreicht. □

6.7.4 Σ-Maß

Satz 60. *Es gilt*

$$A \ \log^2 n\text{-}\Sigma\text{-zufällig} \Rightarrow \forall k \geq 2 : \exists A_k : A_k \ \log^k n\text{-}\Sigma\text{-zufällig} \wedge A_k \leq_{\mathrm{m}}^{\mathrm{polylog}} A$$

Beweis. Der Beweis funktioniert im Wesentlichen wie für Satz 58.
Zu beachten ist nur: Während wir dort eine Abhängigkeitsmengenschranke $\log^k n$ auf $G_{d[s],n}$ verhängt hatten, die durch die Konstruktion in die Schranke $\log^2 n$ auf $G_{d[\widehat{s}],n}$ überführt wird, haben wir jetzt die Situation, dass die Schranke $\log^k n$ auf $G_{s,n}$ in die Schranke $\log^2 n$ auf $G_{\widehat{s},n}$ überführt wird. □

Kapitel 7

Zusammenfassung

Während der Untersuchung verschiedener Ansätze für ein Maß auf **P** kam man zu dem Schluss, dass keiner der bisher vorgeschlagenen Ansätze ideal ist. Keines der Maße weist alle guten Eigenschaften auf, die wir uns wünschen; man muss Kompromisse machen. Hier seien nochmal die wesentlichen Eigenschaften der Maße zusammengefasst.

Die Γ-Martingale, der erste Ansatz von Allender und Strauss, setzt an der Quelle der Probleme an: Um zeigen zu können, dass **P** ein Maß hat, das von 0 verschieden ist, darf die Rekursion, die bei der Diagonalisierung gegen ein Martingal auftritt nicht zu aufwändig werden. Der Γ-Ansatz verhindert genau dies – nicht mehr und nicht weniger. So gesehen ist dieser Ansatz äußerst natürlich. Leider bewirkt er, dass ein Problem auftaucht, das bei weniger restriktiven Zeitschranken nie vorhanden war: Ein Martingal muss nach Definition nicht nur Rechenaufwand in die Vorhersage der nächsten Bits investieren – es entsteht auch Aufwand dabei, den bisherigen Kapitalverlauf nachzuverfolgen. Eine Beschränkung der Abhängigkeitsmengen in diesem Ansatz sorgt dann dafür, dass Klassen wie **SPARSE**, deren Vorhersagbarkeit sehr einfach ist, nicht mehr überdeckt werden können, da die Verwaltung des aktuellen Kapitals zu aufwändig ist.

Der $\Gamma^{\geq}$-Ansatz mit Supermartingalen ist ähnlich natürlich wie der gerade betrachtete. Hier haben wir die Möglichkeit, Abhängigkeitsketten zu durchtrennen. Dies macht die Supermartingale wesentlich mächtiger, so dass auch **SPARSE** überdeckbar wird. Leider ist die andere Seite der selben Medaille jedoch, dass Situationen entstehen können, in denen eine « echte » Wette (also eine Wette mit Einsatz $\neq \frac{1}{2}$) an einer bestimmten Stelle im weiteren Verlauf des Kapitals aus der Abhängigkeitsmenge verschwindet. Dies widerspricht der Intuition und verursacht allerlei Probleme, wie wir gesehen haben. Außerdem hat dieser Ansatz die unschöne Eigenschaft, dass man zwei Nullmengen finden kann, deren Vereinigung ein Maß $\neq 0$ hat.

Die vom $\Gamma^{\geq}$-Ansatz abgeleitete Quotientenformulierung ist der sicherlich technischste der betrachteten Ansätze. Die dahinter stehende Idee ist, dass man eine Möglichkeit schafft, aus einzelnen Überdeckungen auf einfache Weise eine universelle Überdeckung zu konstruieren. Die Schwierigkeit, bei der Konstruktion solcher Überdeckungen ist, die Fairness der Überdeckung sicherzustellen. In der Quotientenformulierung wird deshalb die nummerierte Aufzählung eingeführt, und die Möglichkeit, dagegen zu diagonalisieren, indem gegen jedes einzelne Martingal der Aufzählung diagonalisiert wird, und dann aus den enstandenen Diagonalsprachen das direkte Produkt gebildet wird.

Der Vorteil dieses Ansatzes ist, dass man in einer nummerierten Aufzählung Verletzungen der Fairnessbedingung verzögert erkennen kann, und der hohe Aufwand für diese Überprüfung damit kein Problem mehr ist. Dies wäre z.B. bei der Summation der an den Überdeckungen beteiligten Supermartingale nicht möglich gewesen, denn hier kann i. Allg. bereits eine einzige unfaire Stelle

das gesamte konstruierte universelle Supermartingal ungültig machen.

Die zusätzliche Forderung nach der Überdeckung von unter Quotienten abgeschlossenen Mengen beim $\mathscr{Q}$-Ansatz spielt beim Beweis, dass **P** nicht Maß 0 hat, eine wichtige Rolle. Leider ist es jedoch diese Eigenschaft, die ein äußerst unnatürliches Verhalten bezüglich der Bi-Immunität verursacht.

Der Ansatz der Martingalfamilien ist wiederum relativ natürlich. Im Wesentlichen beruht er darauf, dass man mehrere der Γ-Martingale « parallel » laufen lässt, und zwar so, dass sich ihre Abhängigkeitsmengen nicht überlappen. Dies ermöglicht es, auf jeder Stelle zu wetten, und damit Mengen wie **SPARSE** überdecken zu können. Der Ansatz erlaubt viele einfach und elegante Beweise. Der Preis ist jedoch die Vereinigungseigenschaft: Wenn mehrere Martingalfamilien zu einer « universellen Martingalfamilie » vereinigt werden sollen, so hat man das Problem, dass die Zuständigkeitsbereiche der Familie i. Allg. nur dann klein bleiben werden, wenn die Zuständigkeitsbereiche der zu vereinigenden Familien alle gleich sind. Dies ist eine Schwäche die direkt aus der Idee folgt, die dem Ansatz zugrundeliegt, und sich deshalb kaum beheben lässt.

Der letzte Ansatz ist der von uns eingeführte Σ-Ansatz mit Strategien, deren Abhängigkeitsmengen beschränkt sind. Dieser Ansatz ist recht einfach und elegant, hat aber den Nachteil, dass anders als sonst üblich die Beschränkungen nicht auf die Martingale sondern auf deren zugrundeliegende Strategien verhängt werden. Wir haben gesehen, dass dieser Ansatz allgemeiner ist, als der Ansatz der Martingalfamilien. Wir können also **SPARSE** überdecken, da wir darauf viele aber einfache Wetten tätigen können. Auch hier haben wir aber das Problem, dass es uns nicht gelungen ist, zu zeigen, wie sich dieses Maß für Vereinigungen von Nullmengen verhält.

	Γ-Maß	$\mathscr{Q}$-Maß	$\mathscr{F}$-Maß	Σ-Maß
Idee	Nur Abhängigkeitsmengenbeschränkung.	Supermartingale mit Abhängigkeitsmengenbeschränkung; ein Martingal muss jeweils eine unter Quotienten abgeschlossene Menge überdecken um zulässig zu sein; jede Teilmenge einer auf diese Weise überdeckten Menge ist auch Null.	Viele Martingale mit gemeinsamen Konto, die aber immer nur ihre eigene Kontoentwicklung kennen und nicht den Gesamtkontostand.	Alle Schranken werden auf eine Strategie (die Wett*anteile* ausgibt) verhängt; dadurch viele, aber « schlecht informierte » Wetten möglich.
Fairness	Gleichheitsbedingung.	Kleiner-Gleich-Bedingung, d.h. Geld kann « weggeworfen » werden.	Gleichheitsbedingung.	Gleichheitsbedingung.
Schranken	Auf das Martingal.	Auf das Martingal.	Auf jedes einzelne Martingal in der Auflistung, aber *nicht* auf das « Gesamtmartingal » (d.h. das Produkt der einzelnen Martingale).	Auf die zu Grunde liegende Strategie.
Erfolgskriterium	Kapital des Martingals.	Kapital des Martingals.	Produkt aus den Kapitalwerten der einzelnen Martingale.	Kapital des zur Strategie gehörenden Martingals.
Univ. Überdeck.	Unbekannt.	Nummerierte Aufzählungen, siehe Abschnitt 4.4.	Im Allgemeinen vermutlich nicht möglich.	Im Allgemeinen vermutlich nicht möglich.

Tabelle 7.1: Zusammenfassung der Definitionen der Maße

	Γ-Maß	$\mathscr{Q}$-Maß	$\mathscr{F}$-Maß	Σ-Maß
Vereinigungen	Wenn es eine universelle Maschine für die Martingale gibt, die die zu vereinigenden Mengen überdecken, so hat auch die Vereinigung Maß 0. Siehe [AlS], Theorem 3.	Wenn es eine universelle Maschine für die Martingale gibt, die die zu vereinigenden Mengen überdecken, so hat auch die Vereinigung Maß 0. *Beweis:* Jede Überdeckung ist selbst eine nummerierte Aufzählung. Konstruiere aus der Aufzählung der Aufzählungen dann unter Verwendung einer geeigneten Paarfunktion eine neue Überdeckung.	Vereinigungen sind möglich, wenn die Zuständigkeitsbereiche der Familien, die die zu vereinigengen Megen überdecken, gleich eingeteilt sind. Siehe [Mo], Theorem 3.1. Für allgemeine Vereinigungen fehlen Erkenntnisse.	Unbekannt.
SPARSE	Wird nicht überdeckt. Siehe Satz 9.	Wird überdeckt. Folgt aus [S], Theorem 22.	Wird überdeckt. Siehe Satz 23.	Lässt sich ohne Probleme durch die Strategie $s \equiv 0.75$ überdecken.
Bi-Immunität	Unbekannt.	Umatürliches Bi-Immunitätsverhalten. Siehe Beispiel 19.	Natürliches Verhalten. Siehe Beispiel 24.	Wenn eine Sprache eine andere Sprache $L \in \mathbf{P}$ enthält, bedeckt sie folgende Strategie: $s(X \upharpoonright x) = \begin{cases} 0 & \text{falls } L(x) = 1 \\ \frac{1}{2} & \text{sonst} \end{cases}$ Der Maßansatz verhält sich also bezüglich Bi-Immunität gut.
Zufällige Folgen	Es enstehen zufällige Folgen, denen Eigenschaften fehlen, die man intuitiv erwarten würde. Es lassen sich z.B. zufällige Folgen mit starkem Überschuss an 0en konstruieren, siehe Satz 11.	Weitgehend, aber nicht vollkommen intuitiv zu erwartendes Verhalten bezüglich zufälliger Folgen, vergleiche [S], Abschnitt 5.3.	Jede Folge, die einen ausreichenden Überschuss an Nullen oder Einsen aufweist kann ohne Probleme überdeckt werden (unter Ausnutzung der Tatsache, dass Martingalfamilien an jeder Stelle wetten können).	Jede Folge, die einen ausreichenden Überschuss an Nullen oder Einsen aufweist, kann ohne Probleme überdeckt werden (da an jeder Stelle gewettet werden kann).

Tabelle 7.2: Zusammenfassung der Eigenschaften der Maße

Literaturverzeichnis

[AlS] E. Allender und M. Strauss. Measure on small complexity classes, with applications to BPP. In: Proc. 35'th Foundations of Computer Science Conference, Seiten 807–818. IEEE, 1994.

[AlS2] E. Allender and M. Strauss. Measure on P: Robustness of the notion. In: Proc. of the 20'th Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, Band 969, Seiten 129–138. Springer-Verlag, 1995.

[ASMa] Klaus Ambos-Spies und Elvira Mayordomo. Resource-bounded measure and randomness. In: A. Sorbi (Herausgeber), Complexity, Logic, and Recursion Theory, Lecture Notes in Pure and Applied Mathematics, Band 187, Seiten 1–47. Marcel Dekker, 1997.

[P] C.H. Papadimitriou. Computational Complexity. Addison Wesley, 1994.

[HMU] J.E. Hopcroft, R. Motwani, J.D. Ullman. Introduction to Automata Theory, Languages, and Computation, Second edition. Addison Wesley, 2003.

[L] J.H. Lutz. The quantitative structure of exponential time. In: L. A. Hemaspaandra and A. L. Selman (Herausgeber), Complexity Theory Retrospective II, Seiten 225–254. Springer-Verlag, 1997.

[Mo] P. Moser. Martingale families and dimension in P. In: Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, Lecture Notes in Computer Science, Band 3988, Seiten 388–397. Springer-Verlag, 2006.

[S] M. Strauss, Measure on P: Strength of the notion. Information and Computation, 136(1), Seiten 1–23. 1997.