Zur Trennung von Klassen ununterscheidbarer Ensembles

Dissertation
zur Erlangung des Grades
des Doktors der Naturwissenschaften
der Technischen Fakultät
der Universität des Saarlandes
von

Bernd Meyer

Saarbrücken
1995

Tag des Kolloquiums:	8. Juni 1995
Dekan:	Prof. Dr. H. Bley
Berichterstatter:	Prof. Dr. J. Buchmann
	Prof. Dr. W. Paul

An dieser Stelle möchte ich Herrn Professor Buchmann herzlich dafür danken, daß ich an seinem Lehrstuhl diese Arbeit erstellen durfte. Er nahm sich stets die Zeit, meine Arbeit durch Ratschläge, Diskussionen und Verbesserungsvorschläge zu unterstützen.

Ferner möchte ich Herrn Professor Paul für die Übernahme des Koreferates danken.

Herr Professor Joel Spencer hat mir mehrmals bei der Lösung der kombinatorischen Probleme geholfen und Referenzen weiterer Quellen angegeben. Ich danke ihm herzlich für seine Hilfe.

Mein besonderer Dank gebührt der Deutschen Forschungsgemeinschaft, die im Rahmen des Projekts „Crypto“ diese Arbeit ermöglichte.

Besonders dankbar bin ich auch meinen Eltern, meinem Bruder und meinem Freundeskreis. Meine Eltern und mein Bruder haben mich während meines Studiums stets tatkräftig unterstützt. Meine lieben Freunde Bärbel Müller, Volker Müller, Petra Naumann-Kipper, Bernhard Kipper, Ralf Roth und Diethelm Schlegel hatten stets Zeit für Diskussionen und motivierende Gespräche.

Schließlich danke ich besonders herzlich meinen lieben Freunden und Kollegen Ingrid Biehl, Christoph Thiel und Christian Thiel, die mir durch die stete Bereitschaft zu Diskussionen, durch viele konstruktive Vorschläge und nicht zuletzt durch ihr sorgfältiges Korrekturlesen sehr geholfen haben.

Inhaltsverzeichnis

1 Einleitung

Die schnell wachsende, weltweite Vernetzung von Computern führt dazu, daß Firmen Dienstleistungen und Informationsquellen auf elektronischem Wege anbieten, um schnell und ständig erreichbar zu sein. Viele dieser Dienste benötigen eine sichere Methode zur Übertragung von Nachrichten zwischen Firma und Kunde. Methoden, die im strengen Sinne der Informationstheorie sicher sind, können in der Praxis nicht verwendet werden, weil bei diesen Verfahren die Schlüssellänge mindestens so groß wie die Länge der zu übertragenden Nachrichten sein muß. Außerdem können informationstheoretisch sichere Verfahren nicht asymmetrisch sein. Das heißt, daß zwischen allen möglichen Kommunikationspartnern vor der Nachrichtenübertragung ein geheimer Schlüsselaustausch stattfinden muß. Praktisch bedeutungsvolle Verfahren benutzen öffentliche Schlüssel. Die Sicherheit solcher Verfahren beruht auf komplexitätstheoretischen Methoden und läßt sich formal auf das Lösen schwieriger algorithmischer Probleme reduzieren. Der Nachteil bei diesem Vorgehen ist, daß es bisher nicht möglich ist, die Existenz algorithmischer Probleme mit effizient zu erzeugenden schweren Probleminstanzen nachzuweisen. Letztlich folgt die Sicherheit aus (bisher) unbewiesenen Annahmen.

Wir betrachten in unserer Arbeit folgendes Modell: Zwei probabilistische Maschinen sind über einen öffentlichen Nachrichtenkanal miteinander verbunden. Beide Maschinen erhalten eine gemeinsame Eingabe, die ebenfalls öffentlich ist, und jeweils eine private, geheime Eingabe. Im Verlauf der Berechnungen tauschen nun die beiden Maschinen Nachrichten über den öffentlichen Kanal aus. Eine weitere Maschine, der Lauscher, kennt die gemeinsame öffentliche Eingabe der beiden Maschinen und hört die ausgetauschten Nachrichten ab, ohne diese Nachrichten zu verändern. Der Lauscher erhält also zu seiner Eingabe mehrere Nachrichten gemäß der von den anderen Maschinen generierten Wahrscheinlichkeitsverteilung. Diese zusätzliche Eingabe kann man als eine mit der gemeinsamen Eingabe indizierte Folge von Wahrscheinlichkeitsverteilungen betrachten. Eine solche Folge heißt Ensemble.

Um formal die Geheimhaltung der ausgetauschten Nachrichten und damit auch die Sicherheit einer kryptographischen Anwendung in obigem Szenario definieren zu können, wurden mehrere, an der Anschauung orientierte Eigenschaften als Grundlage einer Sicherheitsdefinition diskutiert (siehe beispielsweise [21, 27]): *Polynomielle Sicherheit* verlangt, daß es für den Gegner schwierig ist, bei Eingabe einer verschlüsselten Nachricht, die zufällig aus zwei möglichen Nachrichten ausgewählt wurde, zu entscheiden, welche Nachricht verschlüsselt wurde. *Semantische Sicherheit* fordert, daß die Wahrscheinlichkeit, mit Hilfe der verschlüsselten Nachricht eine Funktion der Nachricht zu berechnen, sich nicht wesentlich von der Wahrscheinlichkeit unterscheidet, die man aus der Wahrscheinlichkeitsverteilung der Nachrichten erwarten müßte. *Algorithmische, informationstheoretische Sicherheit* verlangt, daß sich die algorithmische Entropie der Nachrichtenquelle nicht wesentlich ändert, wenn zusätzlich die verschlüsselten Nachrichten bekannt sind.

Alle diese verschiedenen Ansätze haben sich als äquivalent erwiesen [21, 12]. Die verschiedenen Sicherheitsbegriffe führen letztlich eine Ähnlichkeitsrelation auf Ensembles ein. Dabei sind diese Ensembles mit einem Sicherheitsparameter des Modells indiziert. Die in der Literatur gebräuchlichen Sicherheitsbegriffe lassen sich nun auf die Ununterscheidbarkeit von Paaren von Ensemles zurückführen. Wir beschränken uns deshalb auf die Untersuchung der Ununterscheidbarkeit von Ensembles. Ununterscheidbarkeit definiert eine Ähnlichkeitsrelation auf der Menge der Ensembles. Die in obigem Modell erzeugten oder andere daraus errechneten Nachrichten

werden dann als geheim angesehen, wenn es eine Maschine gibt, die eine Folge von Nachrichten erzeugt, so daß das von den Kommunikationspartnern erzeugte Ensemble und das Ensemble des Simulators ununterscheidbar sind. Dabei kennt der Simulator im Gegensatz zu den anderen Maschinen nur öffentliche Eingaben aber keine geheimen Eingaben der Kommunikationspartner. (Siehe auch [9] für die ursprüngliche Definition von interaktiven Beweissystemen und der Zero-Knowledge-Eigenschaft.) Man geht davon aus, daß ein Lauscher von ununterscheidbaren Ensembles die gleiche Information erhält. In der Literatur sind vier verschiedene Abstufungen von Ununterscheidbarkeit gebräuchlich: Perfekte, statistische, Schaltkreis- und algorithmische Ununterscheidbarkeit. Sie unterscheiden sich bezüglich des Berechnungsmodells und der Berechnungskomplexität des Lauschers.

Es wurde in [8] die Existenz von algorithmisch- bzw. schaltkreisununterscheidbaren Ensembles bewiesen, die statistisch ununterscheidbar von der Gleichverteilung bzw. von jedem durch uniforme Maschinen erzeugten Ensemble sind. Ferner wurde in [8] ein interaktives Beweissystem mit Zero-Knowledge-Eigenschaft beschrieben, das unsicher wird, wenn man das Beweissystem iteriert. (Dies führte zu einer Änderung der Definition der Zero-Knowledge-Eigenschaft [23].) Goldreich zeigt in [7], daß genau dann zwei in polynomieller Zeit erzeugbare, algorithmisch ununterscheidbare aber statistisch unterscheidbare Ensembles existieren, wenn es Pseudozufallszahlengeneratoren gibt. In [10, 11] wird untersucht, wie sich die Komplexität von Orakelmaschinen verändert, wenn die Orakelfragen einer Wahrscheinlichkeitsverteilung unterliegen müssen, um einem Lauscher der Orakelfragen keine Information zu verraten. Darüber hinaus deutete die Tatsache, daß Resultate, die für nichtuniforme Lauscher bewiesen wurden, sich meistens nur mit erheblichen Mehraufwand für uniforme Lauscher übertragen ließen, darauf hin, daß algorithmische und Schaltkreisununterscheidbarkeit verschieden sind. (Vergleiche beispielsweise [17] und [13].)

Wir werden zeigen, daß die vier Abstufungen von Ununterscheidbarkeit paarweise verschieden sind. Dazu werden wir ein allgemeines Verfahren zur Konstruktion von Ensemblepaaren angeben, die bezüglich einer gegebenen Komplexitätsklasse oder Menge von Maschinen ununterscheidbar sind. Dies geschieht in zwei Schritten: Zuerst verallgemeinern wir den Begriff des Random-Sets (Siehe zum Beispiel [26, 22, 15] und zeigen dann eine Äquivalenz zwischen der Existenz spezieller ununterscheidbarer Ensembles und der Existenz verallgemeinerter Random-Sets. Durch kombinatorische Argumente zeigen wir dann die Existenz von solchen allgemeineren Sprachen. Schließlich werden wir noch Algorithmen beschreiben, um verallgemeinerte Random-Sets berechnen zu können. Mit diesen Algorithmen lassen sich probabilistische Maschinen beschreiben, die die ununterscheidbaren Ensembles generieren.

Die verallgemeinerten Random-Sets werden durch Diagonalisierung über eine Menge von probabilistischen oder nichtuniformen Maschinen berechnet. Um diese Diagonalisierung durchführen zu können, benötigen wir ein Verfahren, probabilistische Maschinen deterministisch zu simulieren.

Ferner untersuchen wir, wie die Ununterscheidbarkeit von Ensembles von der Anzahl der Nachrichten (Samples) abhängt, die dem Lauscher zur Verfügung stehen. Wir konstruieren ein Ensemblepaar, das algorithmisch ununterscheidbar ist, wenn der Lauscher ein Sample erhält, aber algorithmisch unterscheidbar wird bei mehreren Samples. Die anderen Ununterscheidbarkeitsbegriffe hängen nicht von der Anzahl der Samples ab.

Schließlich definieren wir einen Sicherheitsbegriff, der die Komplexität des Lauschers berück-

sichtigt und nicht auf Vergleich von Ensembles beruht. Wir nennen ein Verfahren sicher, wenn sich die Komplexität eines Lauschers durch zusätzliche Eingabe von Samples nicht erhöht. Wir zeigen, daß Ununterscheidbarkeit im allgemeinen diesen Sicherheitsbegriff impliziert.

Kapitel 2 führt die verwendeten Schreibweisen und Begriffe ein. In Kapitel 3 beschreiben wir ein Verfahren zur deterministischen Simulation probabilistischer Maschinen. In Kapitel 4 wird die Äquivalenz zwischen verallgemeinerten Random-Sets und speziellen ununterscheidbaren Ensembles bewiesen. Kapitel 5 enthält Existenzbeweise und Konstruktionsverfahren für die Random-Sets und deren Anwendung zur Trennung der Ununterscheidbarkeitsbegriffe. In Kapitel 6 untersuchen wir den Einfluß der Sampleanzahl auf die Ununterscheidbarkeit. In Kapitel 7 untersuchen wir die neue Sicherheitsdefinition.

Teile dieser Arbeit sind in [20] veröffentlicht. Resultate zur Trennung von algorithmischer und statistischer Ununterscheidbarkeit sowie zur Trennung von ein und zwei Samples wurden unabhängig von dieser Arbeit und mit anderen Methoden in [5] bewiesen.

2 Einige Definitionen und Notationen

Seien A, B Mengen. Wir schreiben $A \Delta B = (A - B) \cup (B - A)$ für die *symmetrische Differenz von A und B*.

Sei $(a_1, \ldots, a_n) \in \mathbb{R}^n$ ein Vektor. Mit $\|a\| = \max\limits_{1 \le i \le n} |a_i|$ wird die *Maximumsnorm von a* bezeichnet.

Wenn P eine Aussage ist, dann ist $[\![P]\!]$ der *boolesche Wert von P*. Das heißt, daß $[\![P]\!]$ den Wert 1 hat, wenn P wahr ist, und den Wert 0, wenn P falsch ist.

Für ein Wort $w = w_1 \ldots w_n \in \{0,1\}^*$ bezeichne $\mathrm{bin}(w) = \sum\limits_{i=0}^{n-1} w_{n-i} 2^i$ den Wert der Binärzahl w.

Eine Funktion $f : \mathbb{N} \to \mathbb{R}$ hat *superpolynomielles Wachstum*, wenn für alle Konstanten $k \in \mathbb{N}$ gilt:

$$\lim_{n \to \infty} \frac{n^k}{f(n)} = 0.$$

Seien $f, g : \mathbb{N} \to \mathbb{N}$ Funktionen. Wir benutzen folgende Schreibweisen:

$$\begin{aligned} g \in \mathrm{O}(f) &\Leftrightarrow \limsup_{n \to \infty} \frac{g(n)}{f(n)} < \infty, \\ g \in \mathrm{o}(f) &\Leftrightarrow \lim_{n \to \infty} \frac{g(n)}{f(n)} = 0, \\ g \in \omega(f) &\Leftrightarrow \limsup_{n \to \infty} \frac{g(n)}{f(n)} = \infty. \end{aligned}$$

Sei im folgenden $\Sigma = \{0,1\}$ das verwendete *Alphabet*. Für ein Wort $w \in \Sigma^*$ bezeichne $|w|$ die *Länge des Wortes*, für eine Menge M bezeichne $|M|$ die Kardinalität der Menge. Eine Menge $L \subseteq \Sigma^*$ heißt *Sprache über* Σ. Mit $L^{=n}$ wird die Menge aller Wörter $w \in L$ mit $|w| = n$ bezeichnet. Analog ist $L^{<n}$ die Menge aller Wörter in L, deren Länge kleiner als n ist. Die *charakteristische Funktion* $\chi_L : \Sigma^* \to \{0,1\}$ von L ist definiert als $\chi_L(x) = [\![x \in L]\!]$ für alle x. Eine Menge L ist *sparse*, wenn es ein Polynom $p \in \mathbb{N}[X]$ gibt, so daß $|L^{=n}| \le p(n)$ für alle $n \in \mathbb{N}$ gilt. Eine Menge von Sprachen $\mathcal{C} \subseteq 2^{\Sigma^*}$ heißt *Sprachklasse*. Mit SPARSE wird die Klasse aller sparse Sprachen in Σ^* bezeichnet. Eine Sprachklasse $\mathcal{C}$ ist *abgeschlossen unter endlicher Variation*, wenn für jede Sprache $C \in \mathcal{C}$ und jede Sprache $D \subseteq \Sigma^*$ mit $|D| < \infty$ gilt $C \Delta D \in \mathcal{C}$.

Wir verwenden folgende Berechnungsmodelle: Das Berechnungsmodell für uniforme deterministische Berechnungen ist die deterministische *Turingmaschine*. Die formale Definition und eine Beschreibung der Arbeitsweise kann zum Beispiel [18] entnommen werden. Ist M eine Turingmaschine, so bezeichnen wir mit $M(x)$ die Ausgabe, die die Maschine M an Eingabe $x \in \Sigma^*$ macht. Mit $\mathrm{time}_M(n)$ wird die maximale Anzahl von Schritten bezeichnet, die M an Eingaben der Länge n macht. Analog bezeichnet $\mathrm{space}_M(n)$ die maximale Anzahl von Speicherzellen, die M für Berechnungen an Eingaben der Länge n benötigt. Die Turingmaschine M hat *polynomielle Laufzeit*, wenn es ein Polynom $p \in \mathbb{N}[X]$ gibt mit $\mathrm{time}_M(n) \le p(n)$ für alle $n \in \mathbb{N}$. Analog wird *polynomieller Platzbedarf* definiert. Mit $L(M)$ wird die von M *akzeptierte Sprache* bezeichnet.

Seien $f, g : \mathbb{N} \to \mathbb{N}$ Funktionen. Die Funktion f heißt *zeitkonstruierbar*, wenn eine Konstante $\varepsilon > 0$ existiert, so daß $f(n) \geq (1+\varepsilon)n$ gilt für alle bis auf endlich viele $n \in \mathbb{N}$ und wenn es eine Turingmaschine M mit $\text{time}_M \in \mathrm{O}(f)$ gibt, die die Funktion f berechnet. Die Funktion g heißt *bandkonstruierbar*, wenn es eine Turingmaschine M mit $\text{space}_M \in \mathrm{O}(g)$ gibt, die die Funktion g berechnet.

Mit $< \cdot, \cdots, \cdot >: (\Sigma^*)^{\mathbb{N}} \to \Sigma^*$ wird eine in polynomieller Zeit berechenbare und invertierbare *Pairing-Funktion* bezeichnet.

Nichtuniforme deterministische Berechnungen werden durch Familien von Schaltkreisen beschrieben. Eine formale Definition von Schaltkreisen ist zum Beispiel in [2] angegeben. Für einen Schaltkreis C bezeichne $\text{size}(C)$ seine *Größe*, das heißt die Anzahl der Gatter in C. Sei $L \subseteq \Sigma^*$ eine Sprache. Eine Familie $C = \{C_x\}_{x \in L}$ heißt *Schaltkreisfamilie polynomieller Größe*, wenn es ein Polynom $p \in \mathbb{N}[X]$ gibt mit $\text{size}(C_x) \leq p(|x|)$ für alle $x \in L$.

Zur Beschreibung probabilistischer Berechnungen verwenden wir die *probabilistische Turingmaschine*. Eine probabilistische Turingmaschine ist eine nichtdeterministische Turingmaschine (siehe auch [18]), die zu jeder Konfiguration höchstens zwei Nachfolgekonfigurationen besitzt, von denen jede mit Wahrscheinlichkeit $\frac{1}{2}$ benutzt wird. (Siehe auch [2, 6] für eine Beschreibung von probabilistischen Turingmaschinen.) Ist M eine probabilistische Turingmaschine und sind $x, y \in \Sigma^*$ Wörter, dann wird mit $\Pr(M(x) = y)$ die Wahrscheinlichkeit bezeichnet, daß M an Eingabe x mit Ausgabe y hält. Die *Laufzeit von M an Eingaben der Länge $n \in \mathbb{N}$* ist die maximale Länge aller Berechnungspfade, die M an Eingaben der Länge n machen kann. Analog zu den deterministischen Turingmaschinen definiert man *Platzbedarf*, *polynomielle Laufzeit* und *polynomiellen Platzbedarf*.

Im folgenden sei mit Turingtransducer eine beliebige deterministische oder probabilistische, uniforme oder nichtuniforme Turingmaschine mit Ein- und Ausgaben aus Σ^* bezeichnet. Unsere Definition von Turingtransducern erlaubt, daß eine Menge $\mathcal{M}$ von Turingtransducern überabzählbar sein kann. Eine Menge von Turingmaschinen ist dagegen stets abzählbar.

Sei $\mathcal{M}$ eine Menge von Turingtransducern. $\mathcal{M}$ heißt *abgeschlossen unter Konkatenation*, wenn es für Maschinen $M, N \in \mathcal{M}$ eine Maschine $O \in \mathcal{M}$ gibt, die bei Eingabe $x \in \Sigma^*$ sich so verhält, als habe man zuerst M mit Eingabe x gestartet und dann N auf das Ergebnis dieser Berechnung.

Sei $\mathcal{M}$ eine Menge von Turingmaschinen. Die Sprachklasse $\mathrm{BP} \cdot \mathcal{M}$ ist die Menge aller Sprachen $L \subseteq \Sigma^*$ für die gilt: Es existieren eine Maschine $M \in \mathcal{M}$ und eine Konstante $\varepsilon > 0$ und für alle Eingaben $x \in \Sigma^*$ gilt $\Pr(M(x) = \chi_L(x)) \geq \frac{1}{2} + \varepsilon$.

Sei $f : \mathbb{N} \to \mathbb{N}$ eine Funktion. Die Sprachklassen P, BPP, PP, DSPACE(f), EXPSPACE und P/Poly sowie die Begriffe *many-one Reduktion* und *Turing-Reduktion* haben die in der Komplexitätstheorie gebräuchliche Interpretation. (Siehe zum Beispiel [2] für eine Definition der Sprachklassen und many-one und Turing-Reduktionen.)

Seien $\mathcal{C} \subseteq 2^{\Sigma^*}$ eine Sprachklasse und $L \subseteq \Sigma^*$ eine Sprache. Wir schreiben $L \in \mathrm{P}(\mathcal{C})$, wenn es eine Turingmaschine M mit polynomieller Laufzeit und eine Sprache $O \in \mathcal{C}$ gibt, so daß L vermittels M auf O Turing-reduzierbar ist.

1. Definition *Eine Funktion $P : \{0,1\}^* \to [0,1]$ mit der Eigenschaft $\sum_{x \in \{0,1\}^*} P(x) = 1$ heißt*

Wahrscheinlichkeitsverteilung auf Σ^*. *Die Menge* $\mathrm{Supp}(P) = \{x \in \{0,1\}^* : P(x) > 0\}$ *heißt* Träger der Wahrscheinlichkeitsverteilung P.

Sei $L \subseteq \Sigma^*$ eine Sprache. Ein Ensemble *ist eine Familie $U = \{U_x\}_{x \in L}$ von Wahrscheinlichkeitsverteilungen $U_x : \{0,1\}^* \to [0,1]$. Der* Träger eines Ensembles U *ist die Vereinigung der Träger der Wahrscheinlichkeitsverteilungen*

$$\mathrm{Supp}(U) = \bigcup_{x \in L} \mathrm{Supp}(U_x).$$

2. Definition *Seien $L \subseteq \Sigma^*$ eine Sprache, M eine probabilistische Turingmaschine und $t : \mathbb{N} \to \mathbb{N}$ eine Funktion. Das Ensemble $U = \{U_x\}_{x \in L}$, definiert durch $U_x(y) = \Pr(M(x) = y)$ für alle $y \in \Sigma^*$, heißt* das von M induzierte Ensemble U. *Ein Ensemble U heißt* in Zeit t generierbar, *wenn es eine probabilistische Turingmaschine M gibt, so daß U das von M induzierte Ensemble ist und wenn für alle Eingaben $x \in L$ gilt $\mathrm{time}_M(|x|) \leq t(|x|)$. Ein Ensemble U heißt* in polynomieller Zeit generierbar, *wenn es ein Polynom $p \in \mathbb{N}[X]$ gibt und U in Zeit p generierbar ist.*

Häufiger werden wir in Beweisen Ensembles statt mit einer Sprache L mit den natürlichen Zahlen $\mathbb{N}$ indizieren. Damit ist gemeint, daß wir die Sprache $\{1^n : n \in \mathbb{N}\}$ zum Indizieren der Wahrscheinlichkeitsverteilungen benutzen und n statt 1^n schreiben. Einem Ensemble $U = \{U_n\}_{n \in \mathbb{N}}$ können wir stets ein Ensemble $U' = \{U'_x\}_{x \in L}$ durch $U'_x = U_{|x|}$ zuordnen.

3. Definition *Seien $L \subseteq \Sigma^*$ eine Sprache, M ein Turingtransducer, $f : \mathbb{N} \to \mathbb{R}^+$ eine Funktion und seien $U = \{U_x\}_{x \in L}$ und $V = \{V_x\}_{x \in L}$ zwei Ensembles. Die Ensembles U und V heißen* ununterscheidbar *für M und f, wenn folgender Grenzwert existiert und die Gleichung*

$$\lim_{|x| \to \infty} f(|x|) \sum_{y \in \Sigma^*} \Pr(M(< x, y >) = 1)(U_x(y) - V_x(y)) = 0. \tag{1}$$

erfüllt.

Seien $\mathcal{M}$ eine Menge von Turingtransducern und $\mathcal{F}$ eine Menge von Funktionen von $\mathbb{N}$ nach $\mathbb{R}^+$. Die Ensembles U und V heißen für $\mathcal{M}$ und $\mathcal{F}$ ununterscheidbar, wenn sie für alle $M \in \mathcal{M}$ und alle $f \in \mathcal{F}$ ununterscheidbar sind.

Zusätzlich werden wir folgende Sprechweise benutzen: Seien $\mathcal{C} \subseteq 2^{\Sigma^*}$ eine Sprachklasse und $\mathcal{F}$ eine Menge von Funktionen von $\mathbb{N}$ nach $\mathbb{R}^+$. Die Ensembles U und V heißen *$\mathcal{C}$-ununterscheidbar* für $\mathcal{F}$, wenn sie für die Menge aller Turingmaschinen, die Sprachen aus $\mathcal{C}$ entscheiden, und für alle Funktionen $f \in \mathcal{F}$, ununterscheidbar sind.

Wenn im folgenden der Begriff Ununterscheidbarkeit ohne die Angabe einer Funktionenklasse benutzt wird, so ist damit immer Ununterscheidbarkeit für die Menge $\mathbb{N}[X]$ gemeint.

Wenn die Ensembles U und V für den Turingtransducer M ununterscheidbar sind, bedeutet das, daß sich asymptotisch die Akzeptanzwahrscheinlichkeiten von M, wenn die Eingabe gemäß der Wahrscheinlichkeitsverteilungen des Ensembles U verteilt ist, an die Akzeptanzwahrscheinlichkeiten von M, wenn die Eingabe gemäß der Wahrscheinlichkeitsverteilungen des Ensembles V verteilt ist, angleichen. Das heißt, daß die Ausgabe von M unabhängig davon ist, ob die Eingaben gemäß U oder V verteilt sind. Für die Berechnungen, die M ausführt, liefern U und V die gleiche „Information“.

4. Definition *Wir bezeichnen mit* PolyTM *die Menge aller probabilistischen Turingmaschinen mit polynomieller Laufzeit und mit* PolyCirc *die Menge aller Schaltkreisfamilien polynomieller Größe.*

Sei $L \subseteq \Sigma^*$ eine Sprache und seien $U = \{U_x\}_{x \in L}$ und $V = \{U_x\}_{x \in L}$ Ensembles. Die Ensemble U und V heißen

1. perfekt ununterscheidbar, *wenn $U = V$ gilt,*
2. statistisch ununterscheidbar, *wenn für alle $f \in \mathbb{N}[X]$ folgender Grenzwert existiert und die Gleichung*
$$\lim_{|x| \to \infty} f(|x|) \sum_{y \in \Sigma^*} |U_x(y) - V_x(y)| = 0$$
erfüllt,
3. schaltkreisununterscheidbar, *wenn U und V für* PolyCirc *und $\mathbb{N}[X]$ ununterscheidbar sind und*
4. algorithmisch ununterscheidbar, *wenn U und V für* PolyTM *und $\mathbb{N}[X]$ ununterscheidbar sind.*

Wenn wir Ununterscheidbarkeit bezüglich Schaltkreisfamilien polynomieller Größe untersuchen, reicht es aus, wenn wir uns auf Schaltkreisfamilien, die mit $\{1^n : n \in \mathbb{N}\}$ indiziert sind, beschränken. Ist nämlich eine Schaltkreisfamilie $C = \{C_x\}_{x \in L}$ ein Unterscheider für zwei Ensembles, dann existiert auch eine Teilfamilie $\{C_{x_i}\}_{i \in \mathbb{N}}$ von Schaltkreisen mit $|x_i| < |x_{i+1}|$ für alle $i \in \mathbb{N}$, die die Ensembles unterscheidet. Zu einer solchen Familie können wir aber eine Folge $\{C'_i\}_{i \in \mathbb{N}}$ mit identischem Ein- Ausgabeverhalten angeben. Das heißt, daß wir o.B.d.A. davon ausgehen können, daß die Indizes x_i in die Schaltkreisfamilie kodiert werden. Damit vereinfacht sich Gleichung (1) für Ensembles $U = \{U_n\}_{n \in \mathbb{N}}$, $V = \{V_n\}_{n \in \mathbb{N}}$ zu der äquivalenten Form:
$$\lim_{n \to \infty} f(n) \sum_{y \in \Sigma^*} C'_n(y)(U_n(y) - V_n(y)) = 0.$$

Folgende Eigenschaften lassen sich leicht mit Hilfe der Dreiecksungleichung und durch Simulation von Turingmaschinen durch Schaltkreisfamilien zeigen (siehe auch [9]):

5. Proposition *Seien U und V Ensembles. Sind U, V perfekt ununterscheidbar, so sind U, V statistisch ununterscheidbar. Sind U, V statistisch ununterscheidbar, so sind U, V schaltkreisununterscheidbar. Sind U, V schaltkreisununterscheidbar, so sind U, V algorithmisch ununterscheidbar.*

6. Proposition *Sei M eine Menge von Ensembles. Die perfekte, statistische, algorithmische oder Schaltkreisununterscheidbarkeit definiert eine Äquivalenzrelation auf M.*

Eines unserer Hauptziele ist es zu zeigen, daß die Umkehrungen der Folgerungen in Proposition 5 im allgemeinen nicht gültig sind.

3 Simulation von probabilistischen Turingmaschinen

In diesem Kapitel wird eine platzeffiziente Methode zur Simulation von probabilistischen Turingmaschinen durch deterministische Turingmaschinen entwickelt. Das Verfahren erlaubt es nicht nur, die Sprachentscheidungseigenschaft der probabilistischen Maschinen zu simulieren, sondern auch die Akzeptanzwahrscheinlichkeiten exakt auszurechenen. Dazu werden wir die divide-and-conquer Technik des Satzes von Savitch verallgemeinern. Die Rekursionsprozedur des Satzes von Savitch (siehe zum Beispiel [14], Theorem 12.11) liefert immer ein Bit Information: Dieses Bit ist genau dann 1, wenn es eine Berechnung gibt, die eine Konfiguration K_1 in eine Konfiguration K_2 in einer beschränkten Anzahl von Schritten überführt. Wir verwenden Rekursionsprozeduren, die zusätzlich zu K_1, K_2 und der Laufzeitschranke einen Parameter i erhalten. Das Ergebnis der Prozedur ist dann das i-te Bit einer Festkommadarstellung der Wahrscheinlichkeit, daß die Maschine von Konfiguration K_1 in Konfiguration K_2 übergeht.

Bezüglich der verwendeten endlichen binären Festkommazahldarstellung werden die Bits folgendermaßen durchnumeriert:

$$(b_0 b_1 \dots b_{m-1}), \text{ mit } b_i \in \{0,1\}, \forall\, 0 \leq i \leq m-1.$$

Der Wert der Zahl mit Festkommazahldarstellung $(b_0 b_1 \dots b_{m-1})$ beträgt dann $\sum_{i=0}^{m-1} b_i 2^{-i}$.

Ist $b = (b_0 b_1 \dots b_{m-1})$ ein Bitstring, so wird mit $\mathrm{Bit}(b, j)$ das Bit b_j von b bezeichnet.

7. Definition *Eine Funktion* $f : (\Sigma^*)^k \to [0,1]$ *heißt* bitweise berechenbar, *wenn es eine deterministische Turingmaschine gibt, die für alle Eingaben* $(x_1, \dots, x_k) \in (\Sigma^*)^k$, $j \in \mathbb{N}$, *die Funktion* $\mathrm{Bit}(f(x_1, \dots, x_k), j)$ *berechnet. In einer* bitweisen Berechnung *von* $f(x_1, \dots, x_k) = (b_0 b_1 \dots b_{m-1})$ *werden nacheinander für alle* $0 \leq j \leq m-1$ *die Werte von* $\mathrm{Bit}(f(x_1, \dots, x_k), j)$ *berechnet und ausgegeben.*

Eine Funktion von dieser Form ist genau dann berechenbar, wenn sie bitweise berechenbar ist. Bitweise Berechenbarkeit bietet in manchen Fällen jedoch die Möglichkeit, Funktionen platzeffizient auszuwerten, weil sich die Maschinen nur die notwendigen Zwischenergebnisse merken müssen.

8. Lemma *Die bitweise Berechnung der Summe von* n *Binärzahlen der Länge* m *benötigt Platz* $\mathrm{O}(\log(n) + \log(m))$.

Beweis: Seien $s_1, \dots, s_n$ die Bitstrings der Summanden und sei o der Bitstring der Summe. Wir betrachten zunächst den Fall, daß die Binärdarstellung der Summe höchstens die Länge m hat. Zur Berechnung von $\mathrm{Bit}(o, k)$ genügt es, die Überträge der $(m-1)$-ten bis $(k+1)$-ten Stellen zu bestimmen:

Die verwendeten Variablen sind:

i:$\{0, \dots, m-1\}$;
j:$\{1, \dots, n\}$;
zsumme:$\{0, \dots, 2n-1\}$;

```
BEGIN
    zsumme← 0;
    FOR i ← m − 1 DOWNTO k DO
        zsumme←zsumme DIV 2;
        FOR j ← 1 TO n DO
        zsumme←zsumme+Bit(s_j, i);
        OD;
    OD;
    RETURN zsumme MOD 2;
END.
```

Es bleibt noch zu zeigen, daß zsumme $\in \{0, \ldots, 2n-1\}$ ausreicht. Sei $\sigma_i \in \mathbb{N}$ der Wert, den die Variable zsumme im i-ten Durchlauf hat, wenn zsumme $\in \mathbb{N}$ vorausgesetzt wird. Es gilt:

$$\sigma_i \leq \left\lfloor \frac{\sum_{k=0}^{i} n2^k}{2^i} \right\rfloor = \left\lfloor n \sum_{k=0}^{i} 2^{-k} \right\rfloor < 2n.$$

Um $\mathrm{Bit}(s_j, i)$ zu bestimmen, sind noch jeweils ein Zähler der Länge $\lfloor \log(n) \rfloor + 1$ und $\lfloor \log(m) \rfloor + 1$ nötig. Für die Nummer des Ausgabebits wird ein Zähler der Länge $\lfloor \log(m) \rfloor + 1$ benötigt.

Für den allgemeineren Fall, daß die Länge der Summe größer als die Länge der Summanden wird, ist noch eine weitere Fallunterscheidung notwendig. Das gesuchte Bit steht dann ebenfalls am Ende in zsumme nur nicht mehr notwendigerweise an der niederwertigsten Stelle. Die Abschätzung für den benötigten Platz bleibt aber korrekt. □

9. Korollar *Die bitweise Berechnung des Produkts zweier Binärzahlen der Länge m benötigt Platz* $\mathrm{O}(\log(m))$.

Beweis: Man kann die Multiplikation zurückführen auf die Addition von m Partialsummen der Länge $2m$. Der zusätzlich benötigte Platz ist lediglich ein konstantes Vielfaches des Platzes, der zur Addition benötigt wird. Die Abschätzung folgt direkt aus Lemma 8.

Seien $x_0 \ldots x_{m-1}$ und $y_0 \ldots y_{m-1}$ die Faktoren. Die Funktion $\mathrm{Bit}(s_j, i)$ zur Berechnung des i-ten Bits, $0 \leq i \leq 2m-1$, der j-ten Partialsumme s_j, $1 \leq j \leq m$, liefert

$$\mathrm{Bit}(s_j, i) = \begin{cases} x_{i+j-m} \cdot y_{m-j} & \text{falls } m-j \leq i \leq 2m-j-1, \\ 0 & \text{sonst.} \end{cases}$$

Zur Berechnung von $\mathrm{Bit}(s_j, i)$ wird Platz $\mathrm{O}(\log(m))$ benötigt. □

Zusätzlich zur bitweisen Addition und Multiplikation werden wir für unsere Anwendungen in anderen Algorithmen (Satz 14, Satz 23, Satz 30, Satz 39) noch Vergleiche, Maximumsbildung und einfache Differenzen von bitweise berechenbaren Zahlen verwenden:

10. Proposition *Seien $b_1, b_2, \ldots, b_n$ Bitstrings der Länge m. Dann gilt:*

1. *Die Relation $b_1 < b_2$ (analog für $\leq, >, \geq, =, \neq$) läßt sich bitweise in Platz* $\mathrm{O}(\log(m))$ *überprüfen.*

2. *Die Funktion* $\max_{1 \le i \le n} b_i$ *läßt sich bitweise in Platz* $O(\log(n) + \log(m))$ *berechnen.*

3. *Die Funktion* $b_1 \cdot 2^{-k}$ *mit* $k \in \mathbb{N}$ *läßt sich bitweise in Platz* $O(\log(m))$ *berechnen.*

4. *Die Funktion* $1 - b_1$ *läßt sich bitweise in Platz* $O(\log(m))$ *berechnen.*

5. *Die Funktion* $|b_1 - \frac{1}{2}|$ *läßt sich bitweise in Platz* $O(\log(m))$ *berechnen.*

Im folgenden werden wir das Prinzip der bitweisen Berechenbarkeit und die Rekursionsidee des Satzes von Savitch zu Berechnung von Akzeptanzwahrscheinlichkeiten von probabilistischen Turingmaschinen benutzen.

11. Lemma *Seien* $f : \mathbb{N} \to \mathbb{N}$ *zeitkonstruierbar,* $g : \mathbb{N} \to \mathbb{N}$ *bandkonstruierbar mit* $g(n) \geq \log(n)$ *und sei* M *eine probabilistische Turingmaschine mit* $\text{time}_M(|x|) \leq f(|x|)$ *und* $\text{space}_M(|x|) \leq g(|x|)$ *für alle Eingaben* $x \in \Sigma^*$. *Sei* $y \in \{0,1\}^*$ *beliebig aber fest. Dann ist die binäre Festkommazahldarstellung der Funktion* $\Pr(M(x) = y)$ *in Platz* $O(\max\{g(n)\log(f(n)), (\log(f(n)))^2\})$ *bitweise berechenbar.*

Beweis: Zu f existiert eine zeitkonstruierbare Funktion $\hat{f} \in O(f)$, so daß $\hat{f}(n)$ eine Zweierpotenz ist und $\hat{f}(n) \geq f(n)$ gilt für alle $n \in \mathbb{N}$. Eine Maschine zur Berechnung von $\hat{f}$ bestimmt zunächst $m = \lceil \log(f(n)) \rceil$ und zählt dann einen Binärzähler der Länge m ab. Mit Theorem 2.6 aus [2] folgt die Zeitkonstruierbarkeit von $\hat{f}$. Sei M' eine probabilistische Turingmaschine, die wie M arbeitet, aber o.B.d.A. auf allen Berechnungspfaden bei Eingaben der Länge n genau $\hat{f}(n)$ Schritte macht und dabei in jedem Schritt ein Zufallsbit verwendet. Wir können aus M eine solche Maschine M' erhalten, wenn wir parallel eine Maschine mit Laufzeit $\hat{f}$ als Uhr mitlaufen lassen. Sei K die Menge aller Konfigurationen von M' (das heißt die Menge aller Tupel aus Zustand der endlichen Kontrolle, den Positionen aller Köpfe und den Inhalten aller Arbeitsbänder) und seien $\alpha, \beta \in K$. Mit $p(\alpha, \beta, k)$ wird die Wahrscheinlichkeit bezeichnet, daß M' in genau 2^k Schritten von Konfiguration α in Konfiguration β übergeht. Mit $p(\alpha, \beta, k, i)$ wird das i-te Bit der Binärdarstellung von $p(\alpha, \beta, k)$ (entsprechend unserer Festkommazahldarstellung) bezeichnet. Dadurch daß die Laufzeit von M' auf allen Berechnungspfaden genau $\hat{f}(n)$ beträgt, können auch alle auftretenden Wahrscheinlichkeiten als Binärstrings der Länge $\hat{f}(n) + 1$ dargestellt werden.

Das Prädikat $P_{2^k}^{M'}(\alpha, \beta, z)$ mit $z \in \{0,1\}^{2^k}$ ist genau dann 1, wenn die Maschine M' in genau 2^k Schritten von Konfiguration α nach β übergeht und dabei die Zufallsbits z benutzt. Andernfalls hat $P_{2^k}^{M'}(\alpha, \beta, z)$ den Wert 0. M' wird durch Festlegung der Zufallsbits z deterministisch. Es gilt

$$\begin{aligned}
p(\alpha, \beta, k+1) &= \frac{1}{2^{2^{k+1}}} \sum_{z \in \{0,1\}^{2^{k+1}}} P_{2^{k+1}}^{M'}(\alpha, \beta, z) \\
&= \frac{1}{2^{2^{k+1}}} \sum_{z_1, z_2 \in \{1,2\}^{2^k}} \sum_{\gamma \in K} P_{2^k}^{M'}(\alpha, \gamma, z_1) P_{2^k}^{M'}(\gamma, \beta, z_2) \\
&= \sum_{\gamma \in K} \Big(\frac{1}{2^{2^k}} \sum_{z_1 \in \{0,1\}^{2^k}} P_{2^k}^{M'}(\alpha, \gamma, z_1) \Big) \Big(\frac{1}{2^{2^k}} \sum_{z_2 \in \{0,1\}^{2^k}} P_{2^k}^{M'}(\gamma, \beta, z_2) \Big) \\
&= \sum_{\gamma \in K} p(\alpha, \gamma, k) p(\gamma, \beta, k).
\end{aligned}$$

Man kann also mit einer rekursiven Prozedur zur Berechnung von $p(\alpha, \beta, k)$, die (wie im Satz von Savitch) über alle Konfigurationen summiert und sich jeweils selbst für die Wahrscheinlichkeiten der Berechnungspfade halber Länge aufruft, die Akzeptanzwahrscheinlichkeit einer probabilistischen Turingmaschine berechnen. Die Prozedur hat die Parameter α und β für Start- und Zielkonfiguration, k für die Rekursionstiefe und i für die Nummer des zu berechnenden Bits von $p(\alpha, \beta, k)$. Allerdings haben die Binärdarstellungen der $p(\alpha, \beta, k)$ die maximale Länge $\hat{f}(n)+1$. Wird aber die Arithmetik aus Lemma 8 und Korollar 9 benutzt und immer nur das nächste benötigte Bit der Binärdarstellung errechnet, ergibt sich die Aussage des Lemmas. Für den benötigten Platz gilt:

- In der Rekursionsprozedur:
  - $O(g)$ zur Darstellung von α, β und dem Zähler γ,
  - $O(\log\log(f))$ für den Parameter k,
  - $O(\log(f))$ für das zu berechnende Bit i,
  - $O(g + \log(f))$ für die Summe der Wahrscheinlichkeiten über alle Konfigurationen γ und
  - $O(\log(f))$ für das Produkt der Akzeptanzwahrscheinlichkeiten der halbierten Berechnungspfade.
- Im Hauptprogramm:
  - $O(g)$ für einen Zähler, der alle Haltekonfigurationen durchläuft,
  - $O(\log(f))$ für das nächste zu berechnende Bit und
  - $O(g + \log(f))$ für die Summe der Akzeptanzwahrscheinlichkeiten.

Die maximale Rekursionstiefe beträgt $O(\log(f))$.

Falls die simulierte probabilistische Maschine mehrere Arbeitsbänder benutzt, können wir o.B.d.A. davon ausgehen, daß die Inhalte aller Arbeitsbänder auf einem Band der simulierenden Maschine kodiert sind (wie zum Beispiel in [18], Seite 201 ff. angegeben). Dadurch erhöht sich die Länge der Kodierung eines Zustandes der simulierten Maschine nur um eine multiplikative Konstante. Die Laufzeit der simulierten Maschine (und damit die Rekursionstiefe in obigem Algorithmus) bleibt unverändert. Daher ist die Abschätzung für den benötigten Platz auch für Mehrbandmaschinen richtig. □

Insbesondere folgt aus Lemma 11, daß es eine deterministische Turingmaschine M'' gibt, die für alle Eingaben $x \in \Sigma^*$ die Binärdarstellung der Akzeptanzwahrscheinlichkeit $\Pr(M(x) = 1)$ in Platz $O(\max\{g\log(f), (\log(f))^2\})$ bitweise berechnet.

Mit Hilfe des Simulationsresultates aus Lemma 11 können wir analog zu den Beweisen für nichtdeterministische Sprachklassen einen Hierarchiesatz zeigen.

12. Definition *Seien $f, g : \mathbb{N} \to \mathbb{N}$ Funktionen, und sei $0 \leq \varepsilon < \frac{1}{2}$ eine Konstante. Eine Sprache $L \subseteq \Sigma^*$ heißt* in Zeit f, Platz g und mit Fehlerschranke ε entscheidbar, *wenn es eine probabilistische Turingmaschine M gibt mit* $\text{time}_M \in O(f)$, $\text{space}_M \in O(g)$ *und* $\Pr(M(x) = \chi_L(x)) \geq \frac{1}{2} + \varepsilon$ *für alle $x \in \Sigma^*$. Mit* $\text{TSRAN}(f, g, \varepsilon)$ *wird die Menge aller in Zeit f, Platz g und mit Fehlerschranke ε entscheidbaren Sprachen bezeichnet.*

In dieser Notation können wir Lemma 11 folgendermaßen formulieren:

13. Korollar *Seien $f : \mathbb{N} \to \mathbb{N}$ eine zeitkonstruierbare und $g : \mathbb{N} \to \mathbb{N}$ eine bandkonstruierbare Funktion mit $g(n) \geq \log(n)$ mit $n \in \mathbb{N}$. Sei $0 \leq \varepsilon < \frac{1}{2}$ eine Konstante, die bitweise in Platz $O(\max\{g\log(f), (\log(f))^2\})$ berechnet werden kann. Es gilt*

$$\mathrm{TSRAN}(f, g, \varepsilon) \subseteq \mathrm{DSPACE}(\max\{g\log(f), (\log(f))^2\}).$$

Die bitweise Berechenbarkeit von ε in Platz $O(\max\{g\log(f), (\log(f))^2\})$ ist notwendig, um zu einer gegebenen Eingabe entscheiden zu können, ob die Akzeptanzwahrscheinlichkeit der probabilistischen Turingmaschine größer als $\frac{1}{2} + \varepsilon$ ist. Ohne diese Bedingung kann die Platzkomplexität eines Algorithmus, der ε bitweise berechnet, größer werden, als der bei der Simulation benötigte Platz.

14. Satz *Seien $f, g : \mathbb{N} \to \mathbb{N}$ Funktionen, sei g bandkonstruierbar, gelte $g(n) \geq \log(n)$ für $n \in \mathbb{N}$ und $g^2 \in o(f)$. Sei $0 \leq \varepsilon < \frac{1}{2}$ eine in Platz $O(g^2)$ bitweise berechenbare Konstante. Es gilt:*

$$\mathrm{TSRAN}(2^{g(n)}, g(n), \varepsilon) \subsetneq \mathrm{TSRAN}(2^{f(n)}, f(n), \varepsilon).$$

Beweis: Wir konstruieren eine Sprache $L \in \mathrm{TSRAN}(2^{f(n)}, f(n), \varepsilon) - \mathrm{TSRAN}(2^{g(n)}, g(n), \varepsilon)$. Sei ρ eine injektive Abbildung, die jeder probabilistischen Turingmaschine M eine Kodierung $\rho(M)$ zuordnet und sei N eine deterministische Turingmaschine, die die Konstante ε in Platz $g(n)^2$ bitweise berechnet. Wir konstruieren eine deterministische Maschine M', die L akzeptiert und dabei höchstens Platz $O(f)$ benötigt. Sei $x \in \Sigma^*$ die Eingabe für M':

- Zunächst prüft M', ob x die Form $\rho(M)01^t$ hat, wobei M eine probabilistische Turingmaschine und $t \in \mathbb{N}$ ist. Ist x nicht die Kodierung einer Turingmaschine, so gibt M' den Wert 1 aus und hält.
- Die Maschine M' markiert $g(|x|)^2$ Bandzellen.
- Nun versucht M' die Haltewahrscheinlichkeit von M bei Eingabe x zu bestimmen. Wird dabei mehr als der markierte Platz benötigt oder ist die Haltewahrscheinlichkeit kleiner als 1, so gibt M' den Wert 1 aus und hält.
- Die Maschine M' prüft nun mit Hilfe der Maschine N, ob $\Pr(M(x) = 1) \geq \frac{1}{2} + \varepsilon$ gilt. Falls ja, so gibt M' den Wert 0 aus sonst den Wert 1. M' hält.

Die Maschine M' arbeitet deterministisch und ist somit auch eine probabilistische Maschine. Insbesondere gilt damit $L(M') \in \mathrm{TSRAN}(2^{f(n)}, f(n), \varepsilon)$. Die Halte- und Akzeptanzwahrscheinlichkeiten im dritten und vierten Schritt können mit der Konstruktion aus Korollar 13 bestimmt werden. Es bleibt noch zu zeigen, daß $L(M') \notin \mathrm{TSRAN}(2^{g(n)}, g(n), \varepsilon)$ gilt. Angenommen es gibt eine probabilistische Maschine M'', die in Platz $g(n)$ und Zeit $2^{g(n)}$ arbeitet und für die $L(M'') = L(M')$ gilt. Betrachten wir nun M'' bei Eingabe $x = \rho(M'')01^s$. Für eine hinreichend

große Zahl $s \in \mathbb{N}$ kann M' bei Eingabe x die Halte- und die Akzeptanzwahrscheinlichkeit ausrechnen.

$$x \notin L(M'') \Rightarrow \Pr(M''(x) = 1) \leq \frac{1}{2} - \varepsilon \Rightarrow x \in L(M') = L(M''), \tag{2}$$
$$x \in L(M'') \Rightarrow \Pr(M''(x) = 1) \geq \frac{1}{2} + \varepsilon \Rightarrow x \notin L(M') = L(M'').$$

In (2) gilt $\Pr(M''(x) = 1) \leq \frac{1}{2} - \varepsilon$, weil M'' nach Voraussetzung eine TSRAN$(2^{g(n)}, g(n), \varepsilon)$-Maschine ist. □

In [4] werden ähnliche Simulationsresultate gezeigt, die keine Annahmen über die Laufzeit der probabilistischen Maschinen enthalten. Die dort verwendeten Techniken sind verschieden von unserer Vorgehensweise und nicht so elementar: Für eine Eingabe fester Länge wird in [4] die probabilistische Turingmaschine als ein stochastischer endlicher Automat aufgefaßt. Sei A die Matrix der Übergangswahrscheinlichkeiten dieses Automaten. Die simulierende Turingmaschine berechnet die Matrix

$$A^* = \sum_{i=0}^{\infty} A^i.$$

Diese Matrix enthält die gesuchten Wahrscheinlichkeiten.

4 Probabilistisch schwer berechenbare Funktionen und Ununterscheidbarkeit

In diesem Kapitel beschreiben wir ein allgemeines Verfahren zur Berechnung ununterscheidbarer Ensembles mit der zusätzlichen Eigenschaft, daß die Elemente der Träger der Wahrscheinlichkeitsverteilungen gleichverteilt sind. Satz 17 ist dabei von zentraler Bedeutung: Er zeigt eine äquivalente Charakterisierung von ununterscheidbaren Ensembles mit dieser Gleichverteilungseigenschaft der Träger der Wahrscheinlichkeitsverteilungen durch sogenannte starke Random-Sets auf. Damit ist die Existenz dieser speziellen ununterscheidbaren Ensembles auf die Existenz starker Random-Sets zurückgeführt.

Der Begriff des Random-Set formalisiert die Eigenschaft einer Sprache, probabilistisch schwer berechenbar zu sein. Die Definition ist ähnlich zu der Definition von Complexity Cores (siehe zum Beispiel [19]): Sei $L \subseteq \Sigma^*$ eine rekursive Sprache. Eine Menge $S \subseteq \Sigma^*$ heißt Complexity Core für L, wenn für jede Turingmaschine M, die L entscheidet und für alle Polynome $p \in \mathbb{N}[X]$ die Menge $\{x \in S \ : \ \mathrm{time}_M(|x|) \leq p(|x|)\}$ endlich ist. Anstatt einer superpolynomiellen Laufzeit fordern wir, daß bei gleichverteilter Eingabe die Erfolgswahrscheinlichkeit minimiert wird:

15. Definition *Seien M ein Turingtransducer, $f : \mathbb{N} \to \mathbb{R}^+$ eine Funktion und seien $A, B \subseteq \Sigma^*$ Sprachen mit $A \subseteq B$ und $|B| = \infty$. Die Sprache A heißt auf B für M bezüglich f* probabilistisch schwer berechenbar, *wenn folgender Grenzwert existiert und die Gleichung*

$$\lim_{n\to\infty} \frac{f(n)}{|B^{=n}|} \sum_{x \in B^{=n}} \Big(\Pr(M(x) = \chi_A(x)) - \Pr(M(x) = 1 - \chi_A(x)) \Big) = 0. \tag{3}$$

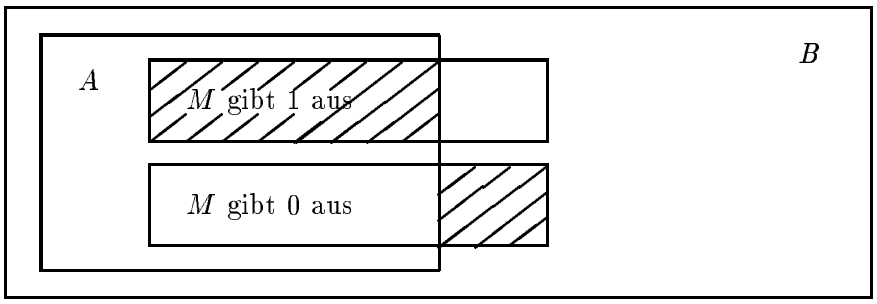
erfüllt.

Seien $\mathcal{M}$ eine Menge von Turingtransducern und $\mathcal{F}$ eine Menge von Funktionen von $\mathbb{N}$ nach $\mathbb{R}^+$. Analog heißt die Sprache A auf B für $\mathcal{M}$ bezüglich $\mathcal{F}$ probabilistisch schwer berechenbar, wenn sie für alle $M \in \mathcal{M}$ und alle $f \in \mathcal{F}$ schwer berechenbar ist. Die Sprache A heißt stark $\mathcal{M}$-random bezüglich B, *wenn sie für $\mathcal{M}$ bezüglich $\mathbb{N}[X]$ schwer berechenbar ist.*

Definition 15 hat folgende intuitive Bedeutung: Die Turingtransducer $M \in \mathcal{M}$ werden als verallgemeinerte (probabilistische) Sprachentscheider betrachtet. Wird ein beliebiger Turingtransducer $M \in \mathcal{M}$ auf die Elemente $x \in B^{=n}$ angewendet, wobei die Elemente gleichverteilt und unabhängig voneinander generiert werden, so konvergiert die Wahrscheinlichkeit, daß M richtig entscheidet, ob $x \in A$ gilt, für wachsende n gegen die Wahrscheinlichkeit, daß M sich irrt, ob $x \in A$ gilt. Das heißt, daß kein Turingtransducer $M \in \mathcal{M}$ besser als durch Raten entscheiden kann, ob für eine zufällige Eingabe $x \in B^{=n}$ auch $x \in A^{=n}$ gilt.

Die durch Gleichung (3) gegebene Definition der Random-Eigenschaft unterscheidet sich von der üblichen Definition von Random-Sets in der Komplexitätstheorie (in [26, 22, 15] wird die Existenz von Random-Sets für verschiedene Komplexitätsklassen und bezüglich verschiedenen Berechnungsmodellen gezeigt) in folgenden Punkten: Random-Sets sind bezüglich Sprachklassen definiert, es wird normalerweise nur der Fall $B = \Sigma^*$ betrachtet und die zusätzliche Bedingung von Gleichung (3) an die Konvergenzgeschwindigkeit des Grenzwertes fehlt.

Wir werden nun einen Zusammenhang zwischen Ununterscheidbarkeit und der starken Random-Set-Eigenschaft herstellen. Das folgende Beispiel zeigt, daß dafür die Menge $\mathcal{M}$ der Unterscheider eine zusätzliche Abschlußeigenschaft haben muß. Sei $A^{=n}$ Träger einer Wahrscheinlichkeitsverteilung U mit Gleichverteilung über dem Träger und $B^{=n} - A^{=n}$ Träger einer anderen Verteilung V und sei $M \in \mathcal{M}$ ein deterministischer Turingtransducer. Dann ist folgende Situation denkbar:



Der Anteil der Strings $x \in B^{=n}$, für die M korrekt antwortet (schraffierter Ausgabebereich), ist so groß wie der Anteil der Strings aus $B^{=n}$, für die M sich irrt (nichtschraffierter Ausgabebereich). Damit erfüllen die Mengen A und B die starke Random-Set-Eigenschaft bezüglich des Turingtransducers M. Andererseits ist aber der Anteil Strings aus $A^{=n}$, für den M den Wert 1 ausgibt, deutlich größer als der entsprechende Anteil Strings von $B^{=n} - A^{=n}$. Das heißt, daß M ein Unterscheider für U und V ist. Betrachten wir nun einen Turingtransducer M', der 1 ausgibt, wenn M den Wert 1 oder 0 ausgibt, und 0 sonst. Für M' ist nun der Anteil der Strings $x \in B^{=n}$, für die M' korrekt antwortet, größer als der Anteil der Strings aus $B^{=n}$, für die M' sich irrt. Dies widerspricht der Random-Set-Eigenschaft. Wenn die Menge der Turingtransducer $\mathcal{M}$ unter solchen „Ausgabetransformationen“ abgeschlossen ist, kann die beschriebene Situation nicht auftreten. Dann sind Ununterscheidbarkeit und starke Random-Set-Eigenschaft äquivalent. Diesen beobachteten Zusammenhang werden wir im folgenden Satz 17 nun formal beweisen.

16. Definition *Sei $\mathcal{M}$ eine Menge von Turingtransducern. Die Menge $\mathcal{M}$ heißt* abgeschlossen unter many-one Reduktionen mit linearer Laufzeit, *wenn für alle Turingtransducer $M \in \mathcal{M}$ und alle Turingmaschinen M' mit linearer Laufzeit ein Turingtransducer $M'' \in \mathcal{M}$ mit folgender Eigenschaft existiert: M'' arbeitet zuerst wie M und wendet dann M' auf das Ergebnis der Berechnung von M an.*

Eine Menge von Turingtransducern $\mathcal{M}$ ist unter obigen „Ausgabetransformationen“ abgeschlossen, wenn $\mathcal{M}$ unter many-one Reduktionen mit linearer Laufzeit abgeschlossen ist. Das Ergebnis 0 oder 1 kann in konstanter Zeit durch ein anderes festes Wort ersetzt werden. Schlimmstenfalls wird eine beliebige Ausgabe gelöscht, dazu wird lineare Laufzeit benötigt, und duch ein anderes festes Wort ersetzt.

Für das ganze Kapitel gilt folgende, vereinfachte Notation: Alle benutzten Ensembles haben die Eigenschaft, daß der Index der Wahrscheinlichkeitsverteilungsfamilien genau die unäre Kodierung der Länge der Wörter des Trägers der Wahrscheinlichkeitsverteilungen ist. Das heißt für ein Ensemble $U = \{U_n\}_{n \in \mathbb{N}}$ gilt immer $\mathrm{Supp}(U_n) \subseteq \Sigma^n$ für alle $n \in \mathbb{N}$. Deshalb können wir

o.B.d.A. davon ausgehen, daß ein Unterscheider M von einem beliebigen Sample $x \in \mathrm{Supp}(U)$ den Index $n \in \mathbb{N}$ mit $x \in \mathrm{Supp}(U_n)$ ausrechnen kann. Wir werden daher den Index der Wahrscheinlichkeitsverteilung nicht mehr dem Unterscheider explizit als Eingabe geben. Zum Beispiel vereinfacht sich Gleichung (1) zu der (in diesem Kapitel) äquivalenten Form:

$$\lim_{n\to\infty} f(n) \sum_{x\in\Sigma^*} \Pr(M(x)=1)(U_n(x)-V_n(x)) = 0.$$

17. Satz *Sei $\mathcal{M}$ eine Menge von Turingtransducern und sei $\mathcal{M}$ unter many-one Reduktionen mit linearer Laufzeit abgeschlossen. Seien $A, B \subseteq \Sigma^*$ Sprachen mit $A \subseteq B$ und $|B| = \infty$. Folgende Aussagen sind äquivalent:*

- *Die Sprache A ist stark $\mathcal{M}$-random bezüglich B,*
- *die Ensembles $U = \{U_n\}_{n\in\mathbb{N}}$ und $V = \{V_n\}_{n\in\mathbb{N}}$ mit*

$$\begin{aligned} U_n(x) &= \frac{1}{|A^{=n}|}[\![x \in A^{=n}]\!] \text{ und} \\ V_n(x) &= \frac{1}{|B^{=n} - A^{=n}|}[\![x \in B^{=n} - A^{=n}]\!] \end{aligned}$$

sind $\mathcal{M}$-ununterscheidbar, und es gilt für alle Konstanten $k \in \mathbb{N}$

$$\lim_{n\to\infty} n^k \Big(\frac{|A^{=n}|}{|B^{=n}|} - \frac{1}{2}\Big) = 0. \tag{4}$$

In Behauptung 18 und Behauptung 19 beweisen wir die in Satz 17 formulierte Äquivalenz. Die Beweise beider Behauptungen haben einen ähnlichen Aufbau. Seien die Voraussetzungen für die Behauptungen wie in Satz 17.

18. Behauptung *Ist die Sprache A stark $\mathcal{M}$-random bezüglich B, dann sind die Ensembles $U = \{U_n\}_{n\in\mathbb{N}}$ und $V = \{V_n\}_{n\in\mathbb{N}}$ (definiert wie in Satz 17) $\mathcal{M}$-ununterscheidbar, und es gilt für alle Konstanten $k \in \mathbb{N}$*

$$\lim_{n\to\infty} n^k \Big(\frac{|A^{=n}|}{|B^{=n}|} - \frac{1}{2}\Big) = 0.$$

Beweis: Wir führen einen Widerspruchsbeweis. Angenommen die Sprache A ist stark $\mathcal{M}$-random bezüglich B, und es gibt einen Turingtransducer $M \in \mathcal{M}$, der die Ensembles U und V unterscheidet. Das heißt formal, daß der Grenzwert von

$$\Big|\sum_{x\in\Sigma^*} \Pr(M(x)=1)(U_n(x)-V_n(x))\Big| \tag{5}$$

für n gegen ∞ entweder nicht existiert oder von 0 verschieden ist.

Weil $\mathcal{M}$ unter many-one Reduktionen mit linearer Laufzeit abgeschlossen ist, können wir o.B.d.A. voraussetzen, daß die Maschine M nur Ausgaben aus $\{0,1,?\}$ macht. (Dabei steht „?“ im folgenden für einen beliebigen aber festen, von 0 und 1 verschiedenen String aus Σ^*.) Betrachten wir zunächst den Fall, daß in (5) die Summe unendlich oft positiv wird. Wir zeigen,

daß ein Turingtransducer M' existiert, bezüglich dem die starke Random-Eigenschaft nicht gilt. M' arbeitet folgendermaßen:

$$M'(x) = \begin{cases} 1 & \text{falls } M(x) = 1 \text{ oder } M(x) = 0, \\ 0 & \text{sonst.} \end{cases}$$

Da nach Voraussetzung $\mathcal{M}$ unter many-one Reduktionen mit linearer Laufzeit abgeschlossen ist, gilt $M' \in \mathcal{M}$. Es gelten folgende Umformungen:

$$\begin{aligned}
&\frac{1}{|B^{=n}|} \sum_{x \in B^{=n}} \Big(\Pr(M'(x) = \chi_A(x)) - \Pr(M'(x) = \chi_{\overline{A}}(x)) \Big) \\
&= \frac{1}{|B^{=n}|} \Big(\sum_{x \in A^{=n}} \Big(\Pr(M(x) = 1) + \Pr(M(x) = 0) - \Pr(M(x) = ?) \Big) \\
&\quad + \sum_{x \in B^{=n} - A^{=n}} \Big(\Pr(M(x) = ?) - (\Pr(M(x) = 1) + \Pr(M(x) = 0)) \Big) \Big) \\
&= \frac{1}{|B^{=n}|} \Big(\sum_{x \in A^{=n}} \Big(\Pr(M(x) = 1) + \Pr(M(x) = 0) - (1 - \Pr(M(x) = 1) - \Pr(M(x) = 0)) \Big) \\
&\quad + \sum_{x \in B^{=n} - A^{=n}} \Big(1 - \Pr(M(x) = 1) - \Pr(M(x) = 0) - (\Pr(M(x) = 1) + \Pr(M(x) = 0)) \Big) \Big) \\
&= -\frac{2}{|B^{=n}|} \sum_{x \in B^{=n}} \Big(\Pr(M(x) = \chi_A(x)) - \Pr(M(x) = \chi_{\overline{A}}(x)) \Big) + 2\Big(\frac{1}{2} - \frac{|A^{=n}|}{|B^{=n}|} \Big) \\
&\quad + \Big(\frac{4}{|B^{=n}|} - \frac{2}{|A^{=n}|} \Big) \sum_{x \in A^{=n}} \Pr(M(x) = 1) \\
&\quad - \Big(\frac{4}{|B^{=n}|} - \frac{2}{|B^{=n} - A^{=n}|} \Big) \sum_{x \in B^{=n} - A^{=n}} \Pr(M(x) = 1) \\
&\quad + 2 \sum_{x \in \Sigma^*} \Pr(M(x) = 1)(U_n(x) - V_n(x)). \qquad (6)
\end{aligned}$$

Weil $\mathcal{M}$ unter many-one Reduktionen mit linearer Laufzeit abgeschlossen ist, gibt es einen Turingtransducer $M'' \in \mathcal{M}$ mit $\Pr(M''(x) = 1) = 1$ für alle $x \in \Sigma^*$. Da ferner A stark $\mathcal{M}$-random bezüglich B ist, gilt für alle $k \in \mathbb{N}$

$$\begin{aligned}
&\lim_{n \to \infty} \frac{n^k}{|B^{=n}|} \Big(\sum_{x \in B^{=n}} \Pr(M''(x) = \chi_A(x)) - \Pr(M''(x) = \chi_{\overline{A}}(x)) \Big) = 0 \\
&\Leftrightarrow \quad \lim_{n \to \infty} \frac{n^k}{|B^{=n}|} \Big(|A^{=n}| - (|B^{=n}| - |A^{=n}|) \Big) = 0 \\
&\Leftrightarrow \quad \lim_{n \to \infty} n^k \Big| \frac{|A^{=n}|}{|B^{=n}|} - \frac{1}{2} \Big| = 0. \qquad (7)
\end{aligned}$$

Wir untersuchen nun den Grenzwert von

$$\lim_{n \to \infty} \frac{n^c}{|B^{=n}|} \sum_{x \in B^{=n}} \Big(\Pr(M'(x) = \chi_A(x)) - \Pr(M'(x) = \chi_{\overline{A}}(x)) \Big). \qquad (8)$$

Dazu genügt es, die Grenzwerte der Summanden in (6) zu betrachten. Der erste Summand hat den Grenzwert 0, da nach Voraussetzung A stark $\mathcal{M}$-random bezüglich B ist. Der zweite

Summand hat wegen (7) den Grenzwert 0. Den dritten Grenzwert können wir folgendermaßen abschätzen:

$$\begin{aligned}
&\lim_{n\to\infty} n^c \Big| \frac{4}{|B^{=n}|} - \frac{2}{|A^{=n}|} \Big| \sum_{x\in A^{=n}} \Pr(M(x)=1) \\
&\leq \lim_{n\to\infty} n^c \frac{|4|A^{=n}| - 2|B^{=n}||}{|B^{=n}|\cdot|A^{=n}|} \cdot |A^{=n}| \\
&= 4 \lim_{n\to\infty} n^c \Big| \frac{|A^{=n}|}{|B^{=n}|} - \frac{1}{2} \Big| \\
&= 0.
\end{aligned}$$

Dabei haben wir (7) benutzt. Analog folgt, daß der vierte Summand in (6) den Grenzwert 0 hat. Da nach (5) der letzte Summand in (6) nicht den Grenzwert 0 hat, kann auch (8) nicht den Grenzwert 0 haben. Das widerspricht der Voraussetzung, daß A stark $\mathcal{M}$-random bezüglich B ist.

Der Fall, daß in (5) die Summe unendlich oft negativ wird, läßt sich analog beweisen. □

19. Behauptung *Seien die Ensembles $U = \{U_n\}_{n\in\mathbb{N}}$ und $V = \{V_n\}_{n\in\mathbb{N}}$ (definiert wie in Satz 17) $\mathcal{M}$-ununterscheidbar, und es gelte für alle Konstanten $k \in \mathbb{N}$*

$$\lim_{n\to\infty} n^k \Big(\frac{|A^{=n}|}{|B^{=n}|} - \frac{1}{2} \Big) = 0, \tag{9}$$

dann ist die Sprache A stark $\mathcal{M}$-random bezüglich B.

Beweis: Wir führen wieder einen Widerspruchsbeweis. Angenommen die Ensembles U und V sind $\mathcal{M}$-ununterscheidbar, und es gibt einen Turingtransducer $M \in \mathcal{M}$, für den A nicht stark $\mathcal{M}$-random bezüglich B ist. Das heißt formal, daß der Grenzwert von

$$\frac{1}{|B^{=n}|} \Big| \sum_{x\in B^{=n}} \Pr(M(x) = \chi_A(x)) - \Pr(M(x) = \chi_{\overline{A}}(x)) \Big| \tag{10}$$

für n gegen ∞ entweder nicht existiert oder von 0 verschieden ist.

Weil $\mathcal{M}$ unter many-one Reduktionen mit linearer Laufzeit abgeschlossen ist, können wir o.B.d.A. voraussetzen, daß die Maschine M nur Ausgaben aus $\{0, 1, ?\}$ macht. Betrachten wir zunächst den Fall, daß in (10) die Summe unendlich oft positiv wird. Wir zeigen, daß ein Turingtransducer M' existiert, der U und V unterscheidet. M' arbeitet folgendermaßen:

$$M'(x) = \begin{cases} 1 & \text{falls } M(x) = 1 \text{ oder } M(x) = 0, \\ 0 & \text{sonst.} \end{cases}$$

Da nach Voraussetzung $\mathcal{M}$ unter many-one Reduktionen mit linearer Laufzeit abgeschlossen ist, gilt $M' \in \mathcal{M}$. Es gelten folgende Umformungen:

$$\sum_{x\in\Sigma^*} \Pr(M'(x) = 1)(U_n(x) - V_n(x))$$

$$
\begin{aligned}
&= \sum_{x\in\Sigma^*} (\Pr(M(x)=1)+\Pr(M(x)=0))\Big(\frac{[\![x\in A^{=n}]\!]}{|A^{=n}|} - \frac{[\![x\in B^{=n}-A^{=n}]\!]}{|B^{=n}|-|A^{=n}|}\Big)\\
&= 2\sum_{x\in\Sigma^*}\Pr(M(x)=1)(U_n(x)-V_n(x)) + \frac{1}{|A^{=n}|}\sum_{x\in A^{=n}}(\Pr(M(x)=0)-\Pr(M(x)=1))\\
&\quad + \frac{1}{|B^{=n}|-|A^{=n}|}\sum_{x\in B^{=n}-A^{=n}}(\Pr(M(x)=1)-\Pr(M(x)=0))\\
&= 2\sum_{x\in\Sigma^*}\Pr(M(x)=1)(U_n(x)-V_n(x))\\
&\quad + \Big(\frac{1}{|A^{=n}|}-\frac{2}{|B^{=n}|}\Big)\sum_{x\in A^{=n}}(\Pr(M(x)=0)-\Pr(M(x)=1))\\
&\quad + \Big(\frac{1}{|B^{=n}|-|A^{=n}|}-\frac{2}{|B^{=n}|}\Big)\sum_{x\in B^{=n}-A^{=n}}(\Pr(M(x)=1)-\Pr(M(x)=0))\\
&\quad - \frac{2}{|B^{=n}|}\sum_{x\in B^{=n}}\Big(\Pr(M(x)=\chi_A(x))-\Pr(M(x)=\chi_{\overline{A}}(x))\Big) \qquad (11)
\end{aligned}
$$

Wir untersuchen nun den Grenzwert von

$$\lim_{n\to\infty} n^c \sum_{x\in\Sigma^*}\Pr(M'(x)=1)(U_n(x)-V_n(x)). \qquad (12)$$

Dazu genügt es wieder, die Grenzwerte der Summanden in (11) zu betrachten. Der erste Summand hat den Grenzwert 0, da nach Voraussetzung die Ensembles U und V $\mathcal{M}$-ununterscheidbar sind. Den zweiten Grenzwert können wir folgendermaßen abschätzen:

$$
\begin{aligned}
&\lim_{n\to\infty} n^c\Big|\frac{1}{|A^{=n}|}-\frac{2}{|B^{=n}|}\Big|\cdot\Big|\sum_{x\in A^{=n}}(\Pr(M(x)=0)-\Pr(M(x)=1))\Big|\\
&\le \lim_{n\to\infty} n^c \frac{||B^{=n}|-2|A^{=n}||}{|A^{=n}|\cdot|B^{=n}|}\cdot|A^{=n}|\\
&= 2\lim_{n\to\infty} n^c\Big|\frac{1}{2}-\frac{|A^{=n}|}{|B^{=n}|}\Big|\\
&= 0.
\end{aligned}
$$

Dabei haben wir (9) benutzt. Analog folgt, daß der dritte Summand den Grenzwert 0 hat. Da nach (10) der letzte Summand in (11) nicht den Grenzwert 0 hat, kann auch (12) nicht den Grenzwert 0 haben. Das widerspricht der Voraussetzung, daß U und V $\mathcal{M}$-ununterscheidbar sind.

Der Fall, daß in (10) die Summe unendlich oft negativ wird, läßt sich analog beweisen. □

Wenn wir die klassische Definition von Random-Sets um die Forderung der superpolynomiellen Konvergenzgeschwindigkeit verallgemeinern, erhalten wir einen Spezialfall von Satz 17 für Sprachklassen.

20. Definition *Sei $\mathcal{C}\subseteq 2^{\Sigma^*}$ eine Sprachklasse und seien $A,B\subseteq\Sigma^*$ Sprachen mit $A\subseteq B$ und $|B|=\infty$. Die Sprache A heißt* stark $\mathcal{C}$-random bezüglich B, *wenn für alle Konstanten $k\in\mathbb{N}$ und für alle Sprachen $C\in\mathcal{C}$ gilt:*

$$\lim_{n\to\infty} n^k\Big(\frac{|(C^{=n}\Delta A^{=n})\cap B^{=n}|}{|B^{=n}|}-\frac{1}{2}\Big)=0.$$

Wie in Satz 17 zeigt man:

21. Satz *Sei $\mathcal{C} \subseteq 2^{\Sigma^*}$ eine Sprachklasse mit $\Sigma^* \in \mathcal{C}$. Seien $A, B \subseteq \Sigma^*$ Sprachen mit $A \subseteq B$ und $|B| = \infty$. Folgende Aussagen sind äquivalent:*

- *Die Sprache A ist stark $\mathcal{C}$-random bezüglich B,*
- *die Ensembles $U = \{U_n\}_{n\in\mathbb{N}}$ und $V = \{V_n\}_{n\in\mathbb{N}}$ mit*

$$\begin{aligned} U_n(x) &= \frac{1}{|A^{=n}|}[\![x \in A^{=n}]\!] \text{ und} \\ V_n(x) &= \frac{1}{|B^{=n} - A^{=n}|}[\![x \in B^{=n} - A^{=n}]\!] \end{aligned}$$

sind $\mathcal{C}$-ununterscheidbar, und es gilt für alle Konstanten $k \in \mathbb{N}$

$$\lim_{n\to\infty} n^k \Big(\frac{|A^{=n}|}{|B^{=n}|} - \frac{1}{2}\Big) = 0. \tag{13}$$

Die Bedingung $\Sigma^* \in \mathcal{C}$ wird nur in der Beweisrichtung von der Random-Eigenschaft zur Ununterscheidbarkeit benötigt, um zu zeigen, daß die Trägermengen von U und V die gleiche Mächtigkeit haben bis auf einen Anteil, der asymptotisch kleiner ist als jedes Polynom.

5 Trennung der Ununterscheidbarkeitsbegriffe

Wir wollen in diesem Kapitel die Existenz von berechenbaren Ensembles zeigen, die die in Definition 4 beschriebenen Ununterscheidbarkeitsbegriffe trennen. Zur Konstruktion dieser Ensembles benutzen wir die im letzten Kapitel bewiesene Äquivalenz zwischen starken Random-Sets und Ensembles mit der zusätzlichen Eigenschaft, daß die Elemente der Träger der Wahrscheinlichkeitsverteilungen gleichverteilt sind. Mit Hilfe von Satz 17 haben wir bereits die Existenz dieser speziellen ununterscheidbaren Ensembles auf die Existenz starker Random-Sets zurückgeführt. Wir werden nun mittels geeigneter Diagonalisierungen starke Random-Sets berechnen. Auf diese Weise erhalten wir Ensembles, die algorithmisch ununterscheidbar bzw. schaltkreisununterscheidbar sind. Gleichzeitig werden die starken Random-Sets noch zusätzliche strukturelle Eigenschaften erhalten, die die Unterscheidbarkeit durch polynomielle Schaltkreisfamilien bzw. die statistische Unterscheidbarkeit der zugehörigen Ensembles garantieren.

Die Konstruktionen in diesem Kapitel lassen sich alle in zwei Teile gliedern: Im ersten Teil wird durch ein kombinatorisches Argument bewiesen, daß Random-Sets mit den gewünschten Eigenschaften existieren. Danach geben wir Algorithmen zur Berechnung dieser Sprachen an und untersuchen die Komplexität dieser Algorithmen.

Wir benutzen nun Korollar 13 zur Simulation probabilistischer Turingmaschinen, um mit einer Verallgemeinerung des Diagonalisierungsverfahrens aus [26] probabilistisch schwer berechenbare Sprachen zu konstruieren.

Wir benutzen folgendes kombinatorisches Resultat:

22. Lemma *Seien $\ell, n \in \mathbb{N}$, $n > 0$. Dann existieren für alle Familien $(a_i)_{1 \le i \le n}$ von Vektoren $a_i \in \mathbb{R}^\ell$ mit Maximumsnorm $\|a_i\| \le 1$ für $1 \le i \le n$ Gewichte $\varepsilon_1, \ldots, \varepsilon_n \in \{+1, -1\}$, so daß gilt*

$$\left\| \sum_{i=1}^{n} \varepsilon_i a_i \right\| < \sqrt{2n \ln(2\ell)}.$$

Beweis: In [1] und [25] wird mit probabilistischen Methoden eine schärfere Version des Lemmas bewiesen. Ein direkter Beweis dieses Lemmas ohne probabilistische Methoden steht in [24]. □

23. Satz *Seien $\varepsilon > 0$ eine Konstante, $s, S : \mathbb{N} \to \mathbb{N}$ bandkonstruierbare Funktionen mit $s(n) > \log(n)$ für alle $n \in \mathbb{N}$, sei $t : \mathbb{N} \to \mathbb{N}$ zeitkonstruierbar und gelte $\max\{s \log(t), (\log(t))^2\} \in o(S)$. Dann gibt es eine Sprache $L \in \mathrm{DSPACE}(S)$, so daß L für alle s-platz- und t-zeitbeschränkten probabilistischen Turingmaschinen auf Σ^* bezüglich $S^{\frac{1}{2}-\varepsilon}$ probabilistisch schwer berechenbar ist.*

Beweis: Wir beschreiben die Arbeitsweise einer deterministischen Turingmaschine M', die in Platz $S(n)$ die Sprache L entscheidet. Sei $\sigma_1, \sigma_2, \ldots$ die lexikographisch angeordnete Aufzählung von Σ^*. Sei $x \in \Sigma^*$ beliebig und gelte $x = \sigma_q$. Die Maschine M' markiert bei Eingabe x zunächst $S(|x|)$ Bandzellen und berechnet dann die Konstanten $\ell = \lceil \log(S(|x|)) \rceil$, $c = 2^\ell$ und $d = c\lfloor \frac{q}{c} \rfloor$.

Die Maschine M' prüft das Akzeptanzverhalten aller probabilistischen Maschinen, die eine Kodierung der Länge ℓ haben, auf allen Eingaben im Bereich $\{\sigma_d, \sigma_{d+1}, \ldots, \sigma_{d+c-1}\}$. Dabei werden alle 2^c Möglichkeiten für $\chi_L|_{\{\sigma_d, \ldots, \sigma_{d+c-1}\}}$ durchprobiert. Insbesondere gilt $x \in$

$\{\sigma_d, \sigma_{d+1}, \ldots, \sigma_{d+c-1}\}$. Dabei können wir o.B.d.A. davon ausgehen, daß die überprüften Maschinen nur die Werte $\{0, 1, ?\}$ ausgeben.

Ferner nehmen wir o.B.d.A. an, daß jede Turingmaschine beliebig lange Kodierungen haben kann, zum Beispiel Kodierungen der Form $\rho(M)01^*$. Dafür gibt es zwei Gründe: Erstens müssen beliebige Bandalphabete von kodierten Maschinen mit dem festen Alphabet von M' simuliert werden, wodurch sich der Platzbedarf um eine multiplikative Konstante r erhöht und die Bedingung $r \max\{s(n)\log(t(n)), (\log(t(n)))^2\} \leq S(n)$ wird erst ab einer bestimmten Eingabelänge gültig. Zweitens wird über alle Maschinen, deren Kodierungen eine feste Länge haben, gleichzeitig diagonalisiert, und eine Maschine, deren Kodierung in die Diagonalmenge aufgenommen wird, bleibt daher auch für alle längeren Eingaben in der Diagonalmenge.

Sei $b = b_0 \ldots b_{c-1} \in \{0,1\}^c$ ein Bitstring. Wir interpretieren die einzelnen Bits in b als die Funktionswerte einer charakteristischen Funktion über den Strings $\{\sigma_d, \ldots, \sigma_{d+c-1}\}$. Die Funktion nimmt für den String σ_{d+i} mit $0 \leq i \leq c-1$ den Wert $\mathrm{Bit}(b,i)$ an. Sei $j \in \{0,1\}^\ell$ die Kodierung einer Turingmaschine und werde mit M_j die zugehörige Maschine bezeichnet. Die Maschine M' muß zur Berechnung von L die Werte von

$$\begin{aligned} F(b,j) &= \Big|\frac{1}{c}\sum_{i=0}^{c-1} \Pr(M_j(\sigma_{d+i}) = \mathrm{Bit}(b,i)) - \Pr(M_j(\sigma_{d+i}) = 1 - \mathrm{Bit}(b,i))\Big| \\ &= \Big|\frac{1}{c}\sum_{i=0}^{c-1} \mathrm{Bit}(b,i)\Pr(M_j(\sigma_{d+i}) = 1) + (1 - \mathrm{Bit}(b,i))(\Pr(M_j(\sigma_{d+i}) = 0)) \\ &\quad -\mathrm{Bit}(b,i)\Pr(M_j(\sigma_{d+i}) = 0) - (1 - \mathrm{Bit}(b,i))(\Pr(M_j(\sigma_{d+i}) = 1))\Big| \end{aligned}$$

für verschiedene b und j miteinander vergleichen. Es folgt aus Proposition 10 und Korollar 13, daß die Funktion F bitweise in Platz $\mathrm{O}(\max\{s\log(t), (\log(t))^2\})$ berechnet werden kann und daher zwei Funktionswerte in Platz $\mathrm{O}(\max\{s\log(t), (\log(t))^2\})$ miteinander verglichen werden können. Der Funktionswert von $F(b,j)$ ist ein Maß dafür, wie gut die Maschine M_j eine Sprache L entscheiden kann, wenn die charakteristische Funktion $\chi_L|_{\{\sigma_d,\ldots,\sigma_{d+c-1}\}}$ durch den Bitstring b gegeben ist. Die Funktion F kann nur dann sinnvolle Werte liefern, wenn die Simulation von M_j für alle Eingaben aus $\{\sigma_d, \ldots, \sigma_{d+c-1}\}$ durchgeführt werden kann. Deshalb prüft M', ob die zu simulierende Maschine M_j auf allen Eingaben σ_{d+i} mit Wahrscheinlichkeit 1 hält. Eine äquivalente Formulierung ist

$$\Pr(M_j(\sigma_{d+i}) = 0) + \Pr(M_j(\sigma_{d+i}) = 1) + \Pr(M_j(\sigma_{d+i}) = ?) = 1. \tag{14}$$

Die Bedingung (14) läßt sich ebenfalls in Platz $\mathrm{O}(\max\{s\log(t), (\log(t))^2\})$ überprüfen.

Die Maschine M' sucht für jeden möglichen Bitstring b den jeweils ungünstigsten Gegner—das heißt die Maschine aus der Diagonalmenge, die die größte Abweichung der Funktion F bewirkt—und minimiert dann über alle Bitstrings. Die von M' verwendeten Variablen lassen sich problemlos in Platz $\mathrm{O}(S)$ unterbringen. M' arbeitet nach folgendem Algorithmus:

```
BEGIN
    min← 0;
    minimum_undefiniert←true;
    /* Schleife über alle Möglichkeiten für χ_L von {σ_d, ..., σ_{d+c-1}} */
    FORALL b ∈ {0,1}^c DO
```

```
        maximum_undefiniert←true;
        /* Schleife über alle Maschinenkodierungen der Länge ℓ */
        FORALL j ∈ {0,1}^ℓ DO
            IF j ist Kodierung einer Maschine M THEN
                /* Prüfe, ob die Simulation von M auf allen Eingaben hält */
                maschine_hält_immer←true;
                FOR k ← 0 TO c − 1 DO
                    IF Pr(M(σ_{d+k}) hält) < 1
                    THEN maschine_hält_immer←false;
                    FI;
                OD;
                IF maschine_hält_immer THEN
                    IF maximum_undefiniert THEN
                        max← j;
                        maximum_undefiniert←false;
                    ELSIF F(b, j) > F(b,max) THEN
                        max← j;
                    FI;
                FI;
            FI;
        OD;
        IF ¬ maximum_undefiniert THEN
            /* Das heißt, die Maschine mit der Nummer max bewirkt von
             allen simulierten Maschinen die größte Abweichung */
            IF minimum_undefiniert THEN
                min← b;
                min_machine←max;
                minimum_undefiniert←false;
            ELSIF F(b,max) < F(min,min_machine) THEN
                min← b;
                min_machine←max;
            FI;
        FI;
    OD;
    RETURN Bit(min, q − d);
END.
```

Es bleibt noch zu zeigen, daß die von M' entschiedene Sprache L die Aussage des Satzes erfüllt. Sei eine beliebige s-platz- und t-zeitbeschränkte probabilistische Turingmaschine M gegeben und sei die Eingabe hinreichend lang, so daß es eine Maschine M_j in der Diagonalmenge gibt mit $\Pr(M_j(x) = y) = \Pr(M(x) = y)$ für alle $x, y \in \Sigma^*$ und daß M' die Simulation für alle Eingaben durchführen kann. Mit Lemma 22 gilt für $\chi_L|_{\{\sigma_d,\ldots,\sigma_{d+c-1}\}}$ die Abschätzung

$$\max_{j\in\{0,1\}^\ell} \left| \frac{1}{c} \sum_{k=0}^{c-1} \Pr(M_j(\sigma_{d+k}) = \chi_L(\sigma_{d+k})) - \Pr(M_j(\sigma_{d+k}) = 1 - \chi_L(\sigma_{d+k})) \right|$$

$$
\begin{aligned}
&= \max_{j\in\{0,1\}^\ell} \Big| \frac{1}{c}\sum_{k=0}^{c-1} \chi_L(\sigma_{d+k})\Pr(M_j(\sigma_{d+k})=1) + (1-\chi_L(\sigma_{d+k}))(\Pr(M_j(\sigma_{d+k})=0)) \\
&\quad -\chi_L(\sigma_{d+k})\Pr(M_j(\sigma_{d+k})=0) - (1-\chi_L(\sigma_{d+k}))(\Pr(M_j(\sigma_{d+k})=1))\Big| \\
&= \max_{j\in\{0,1\}^\ell} \Big| \frac{1}{c}\sum_{k=0}^{c-1} (2\chi_L(\sigma_{d+k})-1)(2\Pr(M_j(\sigma_{d+k})=1) + \Pr(M_j(\sigma_{d+k})=?) - 1)\Big| \\
&= \max_{j\in\{0,1\}^\ell} \frac{2}{c}\Big| \sum_{k=0}^{c-1} \underbrace{(2\chi_L(\sigma_{d+k})-1)}_{\in\{+1,-1\}} \underbrace{\Big(\Pr(M_j(\sigma_{d+k})=1) + \frac{1}{2}\Pr(M_j(\sigma_{d+k})=?) - \frac{1}{2}\Big)}_{\in[-\frac{1}{2},\frac{1}{2}]} \Big| \\
&< \frac{2\sqrt{2c\ln(2c)}}{c}.
\end{aligned}
$$

Das heißt, die Differenz der Summe der Wahrscheinlichkeiten von richtigen und falschen Antworten für jeweils c Eingaben beträgt höchstens $2\sqrt{2c\ln(2c)}$. Somit beträgt die mittlere Abweichung für alle Eingaben der Länge $|x|$ höchstens

$$
\frac{\lfloor \frac{\Sigma^{|x|}}{c} \rfloor 2\sqrt{2c\ln(2c)}}{\Sigma^{|x|}} \leq \frac{2\sqrt{2c\ln(2c)}}{c} \leq \frac{1}{S^{\frac{1}{2}-\varepsilon}(|x|)}.
$$

□

24. Korollar *Für die Sprache L aus Satz 23 gilt: Es gibt eine Konstante $c > 0$, so daß für alle $n \in \mathbb{N}$, $n > c$ gilt:*

$$
|\Sigma^n|\Big(\frac{1}{2} - \frac{1}{2S^{\frac{1}{2}-\varepsilon}(n)}\Big) \leq |L^{=n}| \leq |\Sigma^n|\Big(\frac{1}{2} + \frac{1}{2S^{\frac{1}{2}-\varepsilon}(n)}\Big).
$$

Beweis: Sei M die Turingmaschine, die immer 1 ausgibt und hält, sei die Eingabe hinreichend lang, so daß M in der Diagonalmenge vorkommt und simuliert werden kann, und sei $c > 0$ wie in der Voraussetzung. Für alle $n > c$ gilt nun mit Satz 23:

$$
\begin{aligned}
&\frac{1}{|\Sigma^{=n}|}\Big| \sum_{x\in\Sigma^{=n}} \Pr(M(x)=\chi_L(x)) - \Pr(M(x)=1-\chi_L(x))\Big| \\
&= \frac{1}{|\Sigma^{=n}|}\Big| |L^{=n}| - |\Sigma^{=n} - L^{=n}|\Big| \\
&= \frac{1}{|\Sigma^{=n}|}\Big| 2|L^{=n}| - |\Sigma^{=n}|\Big| \leq \frac{1}{S^{\frac{1}{2}-\varepsilon}(n)}. \qquad (15)
\end{aligned}
$$

Aus (15) folgt die Behauptung. □

Wenn wir in Satz 23 für die Funktion S eine superpolynomiell monoton wachsende bandkonstruierbare Funktion verwenden, können wir damit algorithmisch ununterscheidbare Ensemblepaare generieren. Sei PolyTM die in Definition 4 definierte Menge: Die in Satz 23 konstruierte Sprache ist dann stark PolyTM-random bezüglich Σ^*. Seien U und V definiert wie in Satz 17. Das heißt nach Satz 17, daß die Ensembles U und V algorithmisch ununterscheidbar sind.

Es kann allerdings sein, daß die Wahrscheinlichkeiten, mit denen Strings aus $\mathrm{Supp}(U)$ oder $\mathrm{Supp}(V)$ auftreten, von einer probabilistischen Turingmaschine nicht berechnet werden können.

In einem solchen Fall kann eine probabilistische Turingmaschine die Wahrscheinlichkeitsverteilungen der Ensembles nur approximieren. Das folgende Lemma zeigt, daß durch die Approximation die Ununterscheidbarkeit von U und V nicht verloren geht.

25. Lemma *Sei $s : \mathbb{N} \to \mathbb{N}$ bandkonstruierbar und sei $L \in \mathrm{DSPACE}(s)$ eine Sprache. Dann existiert eine probabilistische Turingmaschine Z mit Platzbedarf $\mathrm{O}(|\Sigma|^n + s(n))$, so daß das Ensemble $U = \{U_n\}_{n\in\mathbb{N}}$ mit $U_n(x) = \frac{1}{|L^{=n}|}\chi_{L^{=n}}(x)$, $x \in \Sigma^*$, und das von Z induzierte Ensemble $V = \{V_n\}_{n\in\mathbb{N}}$ mit $V_n(x) = \Pr(Z(1^n) = x)$, $x \in \Sigma^*$, statistisch ununterscheidbar sind.*

Beweis: Die Maschine Z bestimmt bei Eingabe 1^n zunächst für alle Wörter der Länge n, ob das Wort zu L gehört. Sei $\ell = |L^{=n}|$. Nun erzeugt Z einen zufälligen $\{0,1\}$-String w der Länge $\lfloor \log(\ell) \rfloor + 1$. Sei $k = \mathrm{bin}(w)$. Gilt $0 \le k < \ell$, so gibt Z den $(k+1)$-ten String aus $L^{=n}$ aus und hält. Falls $k \ge \ell$ gilt, so bestimmt Z einen neuen zufälligen $\{0,1\}$-String und testet wieder. Nach n^2 erfolglosen Versuchen gibt Z den Wert 0^n aus und hält.

Man kann leicht überprüfen, daß Z die gewünschten Eigenschaften hat. □

26. Satz *Seien* PolyTM *wie in Definition 4 definiert, $S(n) = \lfloor n^{\log(n)} \rfloor$ und sei $L \subseteq \Sigma^*$ die in Satz 23 berechnete, für* PolyTM *schwer berechenbare Sprache. Seien Z_1 und Z_2 die nach Lemma 25 zu den Sprachen L und $\Sigma^* - L$ konstruierten, probabilistischen Turingmaschinen. Dann haben Z_1 und Z_2 Platzverbrauch $\mathrm{O}(2^n)$ und die von Z_1 und Z_2 induzierten Ensembles sind algorithmisch ununterscheidbar.*

Beweis: Die Funktion S ist superpolynomiell und bandkonstruierbar. Die Menge PolyTM ist unter many-one Reduktionen mit linearer Laufzeit abgeschlossen. Nach Satz 17 sind die Ensembles $U = \{U_n\}_{n\in\mathbb{N}}$ und $V = \{V_n\}_{n\in\mathbb{N}}$ mit

$$\begin{aligned} U_n(x) &= \frac{1}{|L^{=n}|}[\![x \in L^{=n}]\!], \\ V_n(x) &= \frac{1}{|\Sigma^n| - |L^{=n}|}[\![x \in \Sigma^n - L^{=n}]\!] \text{ für alle } x \in \Sigma^*, \end{aligned}$$

PolyTM-random. Lemma 25 sichert die Existenz von probabilistischen Turingmaschinen Z_1 und Z_2, deren induzierte Ensembles statistisch ununterscheidbar von U und V sind. Weil aus statistischer die algorithmische Ununterscheidbarkeit folgt (Proposition 5) und weil Ununterscheidbarkeit transitiv ist (Proposition 6), sind die von Z_1 und Z_2 induzierten Ensembles algorithmisch ununterscheidbar. Es ergibt sich direkt aus Lemma 25 und der Definition von S, daß Z_1 und Z_2 Platzverbrauch $\mathrm{O}(2^n)$ haben. □

Im folgenden werden wir Ensembles konstruieren, die die verschiedenen Ununterscheidbarkeitsbegriffe aus Definition 4 trennen. Dazu berechnen wir zum Beispiel eine Sprache, die stark random ist für die in Definition 4 eingeführte Menge PolyTM aller probabilistischen Turingmaschinen mit polynomieller Laufzeit. Damit ist das analog zu Satz 17 definierte Ensemblepaar U und V algorithmisch ununterscheidbar. Um die Unterscheidbarkeit für polynomielle Schaltkreisfamilien zu erhalten, sollen die Ensembles nur einen sparse Träger haben. Sparse Sprachen können von Schaltkreisfamilien polynomieller Größe entschieden werden. Die Vorgehensweise

zur Trennung von Ununterscheidbarkeit durch polynomielle Schaltkreisfamilien und statistische Unterscheidbarkeit ist sehr ähnlich. In diesem Fall können die Träger der Ensembles natürlich nicht sparse sein, sonst wären die Ensembles schaltkreisunterscheidbar. Statt dessen werden die Träger der Ensembles disjunkte Mengen und daher statistisch unterscheidbar sein.

Das Diagonalisierungsverfahren aus Satz 23 kann nicht direkt zur Konstruktion von Ensembles mit sparse Träger verwendet werden, weil aus Korollar 24 folgt, daß die Träger von U und V nicht sparse sein können.

27. Satz

$$\mathrm{P/Poly} = \bigcup_{\substack{S \subseteq \Sigma^* \\ S \text{ sparse}}} \mathrm{P}(S).$$

Beweis: Siehe Theorem 5.5 in [2], Seite 112. □

Aus Satz 27 folgt, daß es zu einer beliebigen sparse Sprache immer eine Schaltkreisfamilie polynomieller Größe gibt, die genau diese sparse Sprache entscheidet.

28. Lemma *Seien $k, \ell \in \mathbb{N}$ mit $\ell \geq k \geq 1$. Dann gilt*

$$\binom{\ell}{k} \geq \left(\frac{\ell}{k}\right)^k.$$

Beweis: Die Behauptung des Lemmas folgt direkt aus Gleichung

$$\binom{\ell}{k} = \prod_{i=0}^{k-1} \frac{\ell - i}{k - i}$$

und der Äquivalenz

$$\frac{\ell - 1}{k - 1} \geq \frac{\ell}{k} \Leftrightarrow \ell \geq k$$

durch Induktion über k. □

29. Lemma *Sei $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$ eine Konstante. Dann gibt es eine Konstante $c > 0$, so daß für alle $n \in \mathbb{N}$ mit $n > c$ und für alle Familien $(a_i)_{1 \leq i \leq 2^n}$ von Vektoren $a_i \in \mathbb{R}^n$ mit Maximumsnorm $\|a_i\| \leq 1$, $1 \leq i \leq 2^n$, eine nichtleere Menge $M \subseteq \{1, \ldots, 2^n\}$ der Mächtigkeit $|M| \leq 2n^2$ und Gewichte $e_i \in \{+1, -1\}$, $i \in M$, existieren mit*

$$\left\| \sum_{i \in M} e_i a_i \right\| \leq 2^{-(1-\varepsilon)n^2}.$$

Ferner gilt, daß genau die Hälfte der e_i, $i \in M$, den Wert $+1$ hat.

Beweis: Der Beweis beruht auf einem Abzählargument nach dem Schubfachprinzip. Sei

$$S = \left\{(b_1, \ldots, b_{2^n}) \ : \ b_i \in \{0,1\}, \sum_{i=1}^{2^n} b_i = n^2\right\}.$$

Die Menge S zählt alle Möglichkeiten auf, Summen aus genau n^2 Einträgen a_i der Familie zu bilden. Daher gilt $|S| = \binom{2^n}{n^2}$. Betrachten wir nun die Abbildung $f : \{0,1\}^{2^n} \to \mathbb{R}^n$ mit

$$f(b_1, \ldots, b_{2^n}) = \sum_{i=1}^{2^n} b_i a_i.$$

Die Bildmenge $f(S)$ ist eine Menge von Punkten in einem n-dimensionalen Würfel der Kantenlänge n^2. Dieser Würfel wird nun gleichmäßig entlang jeder Kante unterteilt und in n-dimensionale Würfel der Kantenlänge δ zerlegt. Dabei wird $\delta > 0$ minimal gewählt, so daß folgende Bedingung noch erfüllt bleibt

$$\binom{2^n}{n^2} > \left(\frac{n^2}{\delta}\right)^n.$$

Ordnet man nun den Elementen $s \in S$ den Würfel zu, in dem der Summenvektor $f(s)$ liegt, dann gibt es mindestens einen Würfel der Kantenlänge δ der zwei Urbilder hat. Seien $s, s' \in S$ zwei solche Urbilder mit $s = (b_1, \ldots, b_{2^n})$ und $s' = (b'_1, \ldots, b'_{2^n})$. Nach Konstruktion gilt $\|f(s - s')\| \leq \delta$. Die Menge M definieren wir durch

$$M = \{a_i \ : \ b_i \neq b'_i, \ 1 \leq i \leq 2^n\}.$$

Der Vektor $s - s'$ enthält die Werte der Gewichte $e_i \in \{+1, -1\}$ der gesuchten Menge M. Es folgt sofort, daß genau die Hälfte der von 0 verschiedenen Komponenten in $s - s'$ den Wert $+1$ haben.

Es bleibt noch zu zeigen, daß man δ hinreichend klein wählen kann. Nach Lemma 28 gilt

$$\binom{2^n}{n^2} \geq \left(\frac{2^n}{n^2}\right)^{n^2}.$$

Wir wählen daher δ so, daß

$$\left(\frac{2^n}{n^2}\right)^{n^2} > \left(\frac{n^2}{\delta}\right)^n \Leftrightarrow \left(\frac{2^n}{n^2}\right)^n > \frac{n^2}{\delta} \Leftrightarrow \delta > 2^{-n^2+2(n+1)\log(n)}.$$

Da für beliebiges $\varepsilon > 0$ und hinreichend große $n \in \mathbb{N}$ stets $\varepsilon n^2 > 2(n+1)\log(n)$ gilt, folgt die Behauptung. □

30. Satz *Seien $s, S : \mathbb{N} \to \mathbb{N}$ bandkonstruierbare Funktionen mit $s(n) \geq n$ für alle $n \in \mathbb{N}$, sei $t : \mathbb{N} \to \mathbb{N}$ zeitkonstruierbar und gelte $\max\{s(n)\log(t(n)), (\log(t(n)))^2, s^3(n)\} \in o(S(n))$. Sei $\mathcal{M} = \{M_i\}_{i \in \mathbb{N}}$ eine in $\mathrm{DSPACE}(S)$ konstruierbare Aufzählung von s-platz- und t-zeitbeschränkten probabilistischen oder deterministischen Turingmaschinen. Dann gibt es unendliche Sprachen $A, B \in \mathrm{DSPACE}(S)$, $A \subseteq B$, mit folgenden Eigenschaften:*

- *A ist stark $\mathcal{M}$-random bezüglich B,*
- *es gibt eine S-platzbeschränkte, deterministische Turingmaschine M', so daß A nicht M'-random bezüglich B ist und*
- *B ist sparse.*

Beweis: O.B.d.A. können wir davon ausgehen, daß die Maschinen M_i nur die Werte $\{0,1,?\}$ ausgeben. Wir beschreiben nun die Arbeitsweise einer deterministischen Turingmaschine M', die die Sprachen A und B in Platz S entscheidet. Dazu führen wir folgende Schreibweisen ein: Ein Paar (T,f) bestehend aus einer Menge $T \subseteq \Sigma^n$ und einer Funktion $f : T \to \{0,1\}$ heißt *geeignet für* Σ^n, wenn gilt:

- $|T| = 2k$ mit $k \in \{1,\ldots,n^2\}$ und
- $\sum_{i \in T} f(i) = \frac{1}{2}|T|$.

Sei mit S_n die Menge aller geeigneten Paare der Menge Σ^n bezeichnet. Ein geeignetes Paar $P \in S_n$ kann in Platz $O(n^3)$ dargestellt werden, (Platz $n|T| \leq 2n^3$ zur Darstellung von T und Platz $|T| \leq 2n^2$ zur Darstellung von f) und es kann in polynomieller Zeit überprüft werden, ob ein Paar geeignet ist. O.B.d.A. können wir davon ausgehen, daß die Menge S_n geordnet ist (zum Beispiel lexikographisch geordnet bezüglich einer Kodierung von geeigneten Paaren).

Jedes geeignete Paar $(T,f) \in S_n$ beschreibt eine Möglichkeit, die bisher konstruierten Teilmengen $A^{<n}$ und $B^{<n}$ um Strings der Länge n zu erweitern. Dabei wird die Menge T die Strings aus $B^{=n}$ enthalten, und die Funktion $f : T \to \{0,1\}$ ist die charakteristische Funktion von A eingeschränkt auf die Menge $B^{=n}$. Wir werden nun durch vollständige Suche über die Menge aller geeigneten Paare S_n, das bezüglich der Ordnung auf S_n kleinste geeignete Paar bestimmen, das die starke Random-Eigenschaft erhält.

Sei $x \in \Sigma^*$ eine Eingabe für M' der Länge $|x| = n$. Sei $I_n \subseteq \{M_1,\ldots,M_n\}$ die Menge der Maschinen M_j, die auf allen Eingaben aus T in höchstens $t(n)$ Schritten mit Wahrscheinlichkeit 1 halten. Die Turingmaschine M durchläuft ganz S_n und bestimmt das bezüglich der Ordnung auf S_n kleinste geeignete Paar $P = (T,f)$, das den Ausdruck

$$A(n,P) = \max_{M \in I_n} \left| \frac{1}{|T|} \sum_{x \in T} \Pr(M(x) = f(x)) - \Pr(M(x) = 1 - f(x)) \right| \tag{16}$$

minimiert. Dabei werden die Maschinen $M \in I_n$ genau $t(n)$ Schritte lang simuliert. Wir definieren $x \in A$ genau dann, wenn $x \in T$ und $f(x) = 1$ bzw. $x \in B$ genau dann, wenn $x \in T$. Nach Konstruktion erhalten wir $|A \cap \Sigma^n| = \frac{1}{2}|B \cap \Sigma^n|$ für alle $n \in \mathbb{N}$.

Der Wert von (16) und die Haltewahrscheinlichkeit von M_j

$$\Pr(M_j(x) = 1) + \Pr(M_j(x) = 0) + \Pr(M_j(x) = ?),$$

sind nach Proposition 10 und Korollar 13 bitweise in Platz $O(\max\{s\log(t), (\log(t))^2\})$ berechenbar. Daher kann das geeignete Paar P in Platz $O(\max\{s\log(t), (\log(t))^2\})$ berechnet werden.

Es gilt:

$$\begin{aligned}
&\Big|\frac{1}{|T|}\sum_{x\in T}\Pr(M_j(x)=f(x))-\Pr(M_j(x)=1-f(x))\Big| \\
&= \Big|\frac{1}{|T|}\sum_{x\in T} f(x)\Pr(M_j(x)=1)+(1-f(x))\Pr(M_j(x)=0) \\
&\quad -f(x)\Pr(M_j(x)=0)-(1-f(x))\Pr(M_j(x)=1)\Big| \\
&= \Big|\frac{1}{|T|}\sum_{x\in T}(2f(x)-1)(\Pr(M_j(x)=1)-(1-\Pr(M_j(x)=1)-\Pr(M_j(x)=?)))\Big| \\
&= \frac{2}{|T|}\Big|\sum_{x\in T}\underbrace{(2f(x)-1)}_{\in\{+1,-1\}}\underbrace{\Big(\Pr(M_j(x)=1)+\frac{1}{2}\Pr(M_j(x)=?)-\frac{1}{2}\Big)}_{\in[-\frac{1}{2},\frac{1}{2}]}\Big|.
\end{aligned}$$

Sei $\varphi : \Sigma^n \to \{1,\ldots,2^n\}$ eine Bijektion der Strings der Länge n auf die natürlichen Zahlen. Lemma 29 kann folgendermaßen angewendet werden:

$$a_{\varphi(x)} = \Big(\Pr(M_j(x)=1)\Big)_{j=1,\ldots,n}^{T},\ e_{\varphi(x)} = 2f(x)-1 \text{ für } x\in\Sigma^n.$$

Aus Lemma 29 folgt deshalb für alle $\varepsilon > 0$ und für alle hinreichend großen $n \in \mathbb{N}$ die Existenz eines geeigneten Paares $\overline{P} \in S_n$ mit

$$A(n,\overline{P}) \le 2^{-(1-\varepsilon)n^2}.$$

Die Folge der geeigneten Paare definiert die Sprachen A und B.

Die beschriebene, deterministische Turingmaschine M' kann leicht zu einer Maschine verändert werden, die die charakteristische Funktion $\chi_A(x)$ berechnet. Für diese neue Maschine kann aber A nicht random bezüglich B sein, denn $A(n,P)$ in (16) nimmt für diese Maschine nur noch den Wert 1 an. □

Mit Satz 30 und Satz 17 folgt nun die Existenz von Ensembles U und V, die bezüglich einer Klasse $\mathcal{M}$ von Turingtransducern ununterscheidbar sind. Lemma 25 sichert die Existenz von probabilistischen Turingmaschinen, die die Wahrscheinlichkeitsverteilungen der Ensembles U und V approximieren. Es gilt:

31. Satz *Es gibt Ensembles U und V, die algorithmisch ununterscheidbar aber schaltkreisunterscheidbar sind. Ferner gibt es probabilistische Turingmaschinen Z_1 und Z_2 mit Platzverbrauch $\mathrm{O}(2^n)$, so daß die von Z_1 und Z_2 induzierten Ensembles algorithmisch ununterscheidbar aber schaltkreisunterscheidbar sind.*

Beweis: Sei $S : \mathbb{N} \to \mathbb{N}$ mit $S(n) = \lfloor n^{\log n} \rfloor$. Die Funktion S ist superpolynomiell und bandkonstruierbar. Sei PolyTM die in Definition 4 definierte Menge von Turingmaschinen. Die Menge PolyTM ist unter many-one Reduktionen mit linearer Laufzeit abgeschlossen. Mit Hilfe von Satz 30 können wir nun die Existenz von Sprachen $A, B \in \mathrm{DSPACE}(S)$ folgern, so daß A stark PolyTM-random bezüglich B ist. Die Laufzeit jeder einzelnen Maschine aus PolyTM ist

ein Polynom. Weil die Funktion S superpolynomiell wächst, ist die Random-Eigenschaft für jede Maschine aus PolyTM nur endlich oft nicht erfüllt. Es folgt aus Satz 17, daß die Ensembles $U = \{U_n\}_{n\in\mathbb{N}}$ und $V = \{V_n\}_{n\in\mathbb{N}}$ mit

$$\begin{aligned} U_n(x) &= \frac{1}{|A^{=n}|}[\![x \in A^{=n}]\!] \text{ und} \\ V_n(x) &= \frac{1}{|B^{=n} - A^{=n}|}[\![x \in B^{=n} - A^{=n}]\!] \end{aligned}$$

PolyTM-ununterscheidbar sind. Lemma 25 sichert die Existenz von probabilistischen Turingmaschinen Z_1 und Z_2, deren induzierte Ensembles statistisch ununterscheidbar von U bzw. V sind. Weil aus statistischer die algorithmische Ununterscheidbarkeit folgt (Proposition 5) und weil Ununterscheidbarkeit transitiv ist (Proposition 6), sind die von Z_1 und Z_2 induzierten Ensembles algorithmisch ununterscheidbar.

Ferner gilt, daß die Sprachen A und B sparse sind. Daher existiert nach Satz 27 eine Schaltkreisfamilie polynomieller Größe $C = \{C_n\}_{n\in\mathbb{N}}$, die A entscheidet. Diese Schaltkreisfamilie ist ein Unterscheider für die von Z_1 und Z_2 induzierten Ensembles. (Man kann zum Beispiel für jedes $n \in \mathbb{N}$ eine Kodierung des in Satz 30 berechneten, geeigneten Paares (T, f) als Advicestring verwenden. Der zugehörige Polynomzeitalgorithmus gibt bei Eingabe x den Wert von $f(x)$ aus.) Weil die Schaltkreisfamilie C für alle $x \in \mathrm{Supp}(U) \cup \mathrm{Supp}(V)$ den Wert von $\chi_A(x)$ berechnet, folgt mit Lemma 25 für alle Konstanten $k \in \mathbb{N}$:

$$\lim_{n\to\infty} n^k \Big(\Big| \sum_{x\in\Sigma^n} C_n(x)(\Pr(Z_1(1^n) = x) - \Pr(Z_2(1^n) = x)) \Big| - 1 \Big) = 0.$$

□

Im folgenden zeigen wir die Existenz von Ensembles, die schaltkreisununterscheidbar aber statistisch unterscheidbar sind. Die Konstruktion aus Satz 30 läßt sich nicht direkt für Schaltkreisfamilien polynomieller Größe übertragen: Die Menge der Turingmaschinen ist abzählbar, die Menge aller Schaltkreisfamilien polynomieller Größe ist überabzählbar. Anstatt Kodierungen von Maschinen in eine mit der Länge der Eingabe wachsende Diagonalmenge aufzunehmen, werden wir jetzt über eine Menge von „Präfixen“ von Schaltkreisfamilien diagonalisieren. Dabei wird die Größe der Schaltkreise in den Präfixen superpolynomiell in der Länge der Eingabe wachsen. Für jede beliebige Schaltkreisfamilie polynomieller Größe $C = \{C_n\}_{n\in\mathbb{N}}$ gibt es eine Konstante $n_C \in \mathbb{N}$, so daß für alle $n > n_C$ der Präfix $(C_1, \ldots, C_n)$ Element der Diagonalmenge ist.

32. Lemma *Es gibt eine Sprache $L \in$ EXPSPACE mit folgender Eigenschaft: Für alle Konstanten $\varepsilon > 0$ und für alle Schaltkreisfamilien polynomieller Größe $C = \{C_n\}_{n\in\mathbb{N}}$ gilt:*

$$\lim_{n\to\infty} \frac{2^{(\frac{1}{2}-\varepsilon)n}}{|\Sigma^n|} \sum_{x\in\Sigma^n} \Big([\![C_n(x) = \chi_L(x)]\!] - \frac{1}{2} \Big) = 0.$$

Beweis: Die Sprache L wird wie in Satz 30 in Runden konstruiert. In Runde $n \in \mathbb{N}$ wird durch vollständige Suche über alle Erweiterungsmöglichkeiten der bisher konstruierten Sprache $L^{<n}$ mit Strings der Länge n die Erweiterung gesucht, die die Random-Eigenschaft am besten

erhält. Ein kombinatorisches Argument zeigt, daß die auf diese Weise konstruierte Sprache die Aussage des Lemmas erfüllt.

Sei $n \in \mathbb{N}$ eine beliebige Zahl und sei $S = \{A \ : \ A \subseteq \Sigma^n\}$ das Mengensystem aller Mengen aus Strings der Länge n. Wir wählen für jede Zahl $n \in \mathbb{N}$ eine Aufzählung $\{S_i\}_{0 \le i \le 2^{2^n}-1}$ des Mengensystems S und identifizieren die Elemente von S mit den natürlichen Zahlen im Bereich $\{0, \ldots, 2^{2^n} - 1\}$. Wir definieren charakteristische Funktionen

$$\chi_i(x) = [\![x \in S_i]\!].$$

Das heißt, $\chi_i(x)$ ist genau dann 1, wenn x ein Element der i-ten Menge des Systems S ist. Sei ρ eine injektive Kodierungsfunktion, die Schaltkreise als Strings über Σ kodiert. Die Funktion ρ habe folgende Eigenschaften: Es existiert ein Polynom $p \in \mathbb{N}[X]$ mit $|\rho(C)| \le p(\mathrm{size}(C))$ für alle Schaltkreise C, und es existiert ein Algorithmus, der überprüft, ob ein String $x \in \Sigma^*$ einen Schaltkreis kodiert $\rho(C) = x$. (Siehe zum Beispiel [2], Seite 100 ff. für die Existenz einer Kodierung mit diesen Eigenschaften.)

Wir beschreiben nun die Arbeitsweise einer deterministischen Turingmaschine M, die die Sprache L entscheidet. Sei $x \in \Sigma^*$ eine Eingabe für M der Länge $|x| = n$. Die Maschine M minimiert den Wert

$$A(n,i) = \max_{\substack{C \text{ Schaltkreis,} \\ |\rho(C)| < n^{\log(n)}}} \Big| \frac{1}{|\Sigma^n|} \sum_{x \in \Sigma^n} [\![C(x) = \chi_i(x)]\!] - \frac{1}{2} \Big| \tag{17}$$

über alle $i \in \{0, \ldots, 2^{2^n} - 1\}$. Sei i' der kleinste Wert i, der (17) minimiert. Dann gibt die Maschine M den Wert $\chi_{i'}(x)$ aus. Weil der Platz, der zum Speichern von i und i' und zur Simulation der Schaltkreise benötigt wird, exponentiell in n ist, gilt $L \in \mathrm{EXPSPACE}$. Darüber hinaus gilt

$$A(n,i) = \max_{\substack{C \text{ Schaltkreis,} \\ |\rho(C)| < n^{\log(n)}}} \frac{1}{|\Sigma^n|} \Big| \sum_{x \in \Sigma^n} \underbrace{\big(C(x) - \frac{1}{2}\big)}_{\in\{-\frac{1}{2},\frac{1}{2}\}} \underbrace{\big(2\chi_i(x) - 1\big)}_{\in\{+1,-1\}} \Big|.$$

Sei $\varphi : \Sigma^n \to \{1, \ldots, 2^n\}$ eine Bijektion der Strings der Länge n auf die natürlichen Zahlen, und sei $(C_1, \ldots, C_\ell)$ die Folge aller Schaltkreise C_j mit $|\rho(C_j)| < n^{\log(n)}$ für $1 \le j \le \ell$. Lemma 22 kann folgendermaßen angewendet werden:

$$a_{\varphi(x)} = \Big(C_j(x) - \frac{1}{2}\Big)_{j=1,\ldots,\ell}^{T}, \ \varepsilon_{\varphi(x)} = 2\chi_i(x) - 1 \text{ für } x \in \Sigma^n.$$

Es folgt mit Lemma 22, daß

$$A(n,i') < \frac{\sqrt{2 \cdot 2^n \ln\big(2 \cdot 2^{n^{\log(n)}}\big)}}{2^n} \le 2^{-\frac{1}{2}n + \frac{1}{2}\log^2(n) + \log(2\sqrt{\ln(2)}) + 1}.$$

Weil es für jede Konstante $\varepsilon > 0$ eine Konstante $c > 0$ gibt, so daß $-\varepsilon n + \frac{1}{2}\log^2(n) + \log(2\sqrt{\ln(2)}) + 1 < 0$ für alle $n \in \mathbb{N}$, $n > c$, gilt, folgt die Behauptung des Lemmas. □

Insbesondere ist die Sprache L stark PolyCirc-random bezüglich Σ^*. Anknüpfend an die Betrachtungen zu Definition 20 erhalten wir mit Lemma 32:

33. Korollar *Die Sprache* $L \in \mathrm{EXPSPACE}$ *aus Lemma 32 ist ein* P/Poly *Random-Set.*

Mit Lemma 32 und Satz 21 folgt nun die Existenz von Ensembles U und V, die bezüglich der Sprachklasse P/Poly und damit bezüglich der in Definition 4 definierten Menge PolyCirc ununterscheidbar sind. Lemma 25 sichert die Existenz von probabilistischen Turingmaschinen, die die Wahrscheinlichkeitsverteilungen der Ensembles U und V approximieren. Es gilt:

34. Satz *Es gibt Ensembles U und V, die schaltkreisununterscheidbar aber statistisch unterscheidbar sind. Ferner gibt es probabilistische Turingmaschinen Z_1 und Z_2 mit Platzverbrauch $O(2^n)$, so daß die von Z_1 und Z_2 induzierten Ensembles schaltkreisununterscheidbar aber statistisch unterscheidbar sind.*

Beweis: Sei $L \subseteq \Sigma^*$ die Sprache aus Lemma 32. Wir setzen $A = L$ und $B = \Sigma^*$. Die Schaltkreisfamilie aus PolyCirc, die immer den Wert 1 ausgibt, akzeptiert die Sprache Σ^*. Es folgt aus Satz 21, daß die Ensembles $U = \{U_n\}_{n\in\mathbb{N}}$ und $V = \{V_n\}_{n\in\mathbb{N}}$ mit

$$\begin{aligned} U_n(x) &= \frac{1}{|A^{=n}|}[\![x \in A^{=n}]\!] \text{ und} \\ V_n(x) &= \frac{1}{|B^{=n} - A^{=n}|}[\![x \in B^{=n} - A^{=n}]\!] \end{aligned}$$

PolyCirc-ununterscheidbar sind. Lemma 25 sichert die Existenz von probabilistischen Turingmaschinen Z_1 und Z_2, deren induzierte Ensembles statistisch ununterscheidbar von U bzw. V sind. Weil aus statistischer die Schaltkreisununterscheidbarkeit folgt (Proposition 5) und weil Ununterscheidbarkeit transitiv ist (Proposition 6), sind die von Z_1 und Z_2 induzierten Ensembles schaltkreisununterscheidbar.

Ferner gilt, daß die Trägermengen $\mathrm{Supp}(U)$ und $\mathrm{Supp}(V)$ disjunkt sind. Damit können aber die Ensembles U und V (und auch die von Z_1 und Z_2 induzierten Ensembles) nicht statistisch ununterscheidbar sein. Mit Lemma 25 gilt für alle Konstanten $k \in \mathbb{N}$:

$$\lim_{n\to\infty} n^k \Big(\sum_{x\in\Sigma^*} |\Pr(Z_1(1^n) = x) - \Pr(Z_2(1^n) = x)| - 2 \Big) = 0.$$

□

Es ist einfach, Beispiele zu finden, die die Trennung von statistischer und perfekter Ununterscheidbarkeit zeigen.

35. Satz *Folgendes Paar von Ensembles U und V ist statistisch ununterscheidbar aber nicht identisch. Wir definieren*

$$\begin{aligned} U_n(x) &= \frac{1}{|\Sigma^n| - 1}[\![x \in \Sigma^n - \{0^n\}]\!] \text{ und} \\ V_n(x) &= \frac{1}{|\Sigma^n|}[\![x \in \Sigma^n]\!]. \end{aligned}$$

Beweis: Für alle Konstanten $k \in \mathbb{N}$ gilt

$$\lim_{n\to\infty} n^k \sum_{x\in\Sigma^*} |U_n(x) - V_n(x)| = \lim_{n\to\infty} \frac{n^k}{2^{n-1}} = 0.$$

□

Wir haben in diesem Kapitel für verschiedene Komplexitätsklassen die Existenz von probabilistisch schwer berechenbaren Sprachen gezeigt und Verfahren zur Konstruktion dieser Sprachen angegeben. Mit Hilfe der im vorherigen Kapitel bewiesenen Äquivalenz konnten wir diese Sprachen zur Konstruktion von ununterscheidbaren Ensembles benutzen. Durch Ensemblepaare mit speziellen strukturellen Eigenschaften konnten wir zeigen, daß alle vier in Definition 4 eingeführten Ununterscheidbarkeitsbegriffe paarweise verschieden sind.

6 Unterscheidbarkeit und Anzahl der Samples

Im letzten Kapitel haben wir Ununterscheidbarkeitsbegriffe bezüglich der Komplexität des Unterscheiders getrennt. Es folgte, daß die vier in der Literatur gebräuchlichen Begriffe (siehe Definition 4) paarweise verschieden sind. Darüber hinaus zeigt Satz 30 (für eine Familie von hinreichend schnell wachsenden bandkonstruierbaren Funktionen) die Existenz einer unendlichen Hierachie von ununterscheidbaren Ensembles. Alle diese Unterscheidungsalgorithmen hatten immer ein einzelnes, gemäß den Wahrscheinlichkeitsverteilungen der Ensembles generiertes Sample als Eingabe. Nun soll untersucht werden, wie sich die Anzahl der Samples, die der Unterscheidungsalgorithmus als Eingabe erhält, auf die Fähigkeit auswirkt, die Ensembles zu unterscheiden. Dabei wird ein struktureller Unterschied zwischen algorithmischer Ununterscheidbarkeit und Schaltkreisununterscheidbarkeit sichtbar.

Mit $< \cdot, \cdots, \cdot >: (\Sigma^*)^{\mathbb{N}} \to \Sigma^*$ bezeichnen wir eine in polynomieller Zeit berechenbare und invertierbare Pairing-Funktion. Wir benötigen folgende Schreibweise:

36. Definition *Seien $L \subseteq \Sigma^*$ eine Sprache, $U = \{U_x\}_{x \in L}$ ein Ensemble und $t : L \to \mathbb{N}$ eine Funktion. Das Ensemble $U^t = \{U_x^t\}_{x \in L}$ ist wie folgt definiert:*

$$U_x^t(y) = \begin{cases} \prod_{i=1}^{t(x)} U_x(y_i) & \text{falls } y =< y_1, \ldots, y_{t(x)} >, \\ 0 & \text{sonst.} \end{cases}$$

Das heißt, die Wahrscheinlichkeitsverteilung U_x^t entsteht aus der Wahrscheinlichkeitsverteilung U_x, indem $t(x)$ Samples unabhängig voneinander gemäß der Verteilung U_x erzeugt werden.

37. Satz *Seien U, V zwei Ensembles, und sei $p \in \mathbb{N}[X]$ ein Polynom. Die Ensembles $U^{p(|\cdot|)}$ und $V^{p(|\cdot|)}$ sind genau dann perfekt ununterscheidbar (oder statistisch oder schaltkreisununterscheidbar), wenn U und V perfekt ununterscheidbar (oder statistisch oder schaltkreisununterscheidbar) sind. Sind U und V in polynomieller Zeit generierbar, so sind $U^{p(|\cdot|)}$ und $V^{p(|\cdot|)}$ genau dann algorithmisch ununterscheidbar, wenn U und V algorithmisch ununterscheidbar sind.*

Beweis: Der Beweis für die algorithmische Ununterscheidbarkeit funktioniert folgendermaßen: Seien $L \subseteq \Sigma^*$ eine Sprache, $p \in \mathbb{N}[X]$ ein Polynom und seien $U = \{U_x\}_{x \in L}$ und $V = \{V_x\}_{x \in L}$ in polynomieller Zeit generierbare Ensembles. Es ist leicht zu zeigen, daß aus der Ununterscheidbarkeit von $U^{p(|\cdot|)}$ und $V^{p(|\cdot|)}$ die Ununterscheidbarkeit von U und V folgt. Der Unterscheider liest nur das erste Sample einer Folge und ignoriert die restliche Eingabe. Weil alle Samples unabhängig voneinander erzeugt werden, folgt daraus die Ununterscheidbarkeit von U und V.

Wir zeigen nun die Rückrichtung des Satzes durch einen indirekten Beweis. Wir betrachten Samplefolgen $s_i = (\alpha_1, \ldots, \alpha_i, \beta, \gamma_1, \ldots, \gamma_{p(|x|)-i-1})$ bei denen die α_j, $1 \leq j \leq i$, gemäß der Wahrscheinlichkeitsverteilung U_x gebildet werden, β die Eingabe des neuen Unterscheiders ist und die γ_j, $1 \leq j \leq p(|x|) - i - 1$, gemäß V_x gebildet werden. Sind nun die Ensembles $U^{p(|\cdot|)}$ und $V^{p(|\cdot|)}$ unterscheidbar, so gibt es auch für unendlich viele Wahrscheinlichkeitsverteilungen U_x und V_x ein $1 \leq i \leq p(|x|)$, so daß der Unterscheider zwischen s_i und s_{i-1} einen polynomiellen Unterschied messen kann. Der neue Unterscheider rät zufällig die Stelle i und generiert Samples

α_j gemäß U und γ_j gemäß V. Damit haben wir eine Polynomzeitreduktion konstruiert, die die Unterscheidbarkeit von Samplefolgen auf die Unterscheidbarkeit einzelner Samples reduziert.

Wir beschreiben nun formal diese Polynomzeitreduktion und zeigen ihre Korrektheit: Seien L, p, U und V gegeben und seien $U^{p(|\cdot|)}$ und $V^{p(|\cdot|)}$ algorithmisch unterscheidbar. Sei die probabilistische Turingmaschine M ein Unterscheider mit polynomieller Laufzeit und seien M_1 und M_2 probabilistische Turingmaschinen mit polynomieller Laufzeit, die die Ensembles U und V erzeugen. Wir beschreiben nun die Arbeitsweise einer Maschine M', die ein Unterscheider für U und V ist. Sei $< x, z >$ die Eingabe für M':

- Die Maschine M' bestimmt zufällig und gleichverteilt eine Zahl $i \in \{0, \ldots, p(|x|) - 1\}$.
- Durch i-maliges Anwenden der Maschine M_1 und $(p(|x|) - i - 1)$-maliges Anwenden der Maschine M_2 werden Strings $u =< u_1, \ldots, u_i >$ und $v =< v_1, \ldots, v_{p(|x|)-i-1} >$ gemäß den Wahrscheinlichkeitsverteilungen U_x^i und $V_x^{p(|x|)-i-1}$ erzeugt.
- Zuletzt simuliert M' die Berechnung der Maschine M an Eingabe $< x, < u, z, v >>$ und gibt das Ergebnis der Berechnung von M aus.

Weil $i < p(|x|)$ gilt und die Laufzeiten von M_1 und M_2 polynomiell in $|x|$ sind, ist auch die Laufzeit von M' polynomiell in $| < x, z > |$. Wenn die Wahrscheinlichkeit $\frac{1}{p(|x|)}$, mit der die Zahl i bestimmt wird, nicht exakt in polynomieller Zeit berechnet werden kann, reicht es aus, den Wert mit ähnlichen Methoden wie im Beweis zu Lemma 25 zu approximieren.

Wir müssen noch zeigen, daß die Turingmaschine M' ein Unterscheider für U und V ist:

$$
\begin{aligned}
&\Big| \sum_{y \in \Sigma^*} \Pr(M'(< x, y >) = 1)(U_x(y) - V_x(y)) \Big| \\
&= \frac{1}{p(|x|)} \Big| \sum_{i=0}^{p(|x|)-1} \sum_{u,z,v \in \Sigma^*} \Pr(M(< x, < u, z, v >>) = 1) U_x^i(u)(U_x(z) - V_x(z)) V_x^{p(|x|)-i-1}(v) \Big| \\
&= \frac{1}{p(|x|)} \Big| \sum_{y \in \Sigma^*} \Pr(M(< x, y >) = 1)(U_x^{p(|x|)}(y) - V_x^{p(|x|)}(y)) \Big|.
\end{aligned}
$$

Damit folgt aus der Unterscheidbarkeit von $U^{p(|\cdot|)}$ und $V^{p(|\cdot|)}$ auch die Unterscheidbarkeit von U und V.

Für perfekte Ununterscheidbarkeit ist nichts zu zeigen. Die Beweise für statistische und Schaltkreisununterscheidbarkeit haben eine ähnliche Struktur. Sie benutzen auch die Teleskopsumme aus obigem Beweis. Beweise für statistische und Schaltkreisununterscheidbarkeit stehen zum Beispiel in [3]. □

In Satz 37 kann für den Fall der statistischen und der Schaltkreisununterscheidbarkeit p durch eine polynomiell beschränkte Folge ersetzt werden. Für den Fall der algorithmischen Ununterscheidbarkeit genügt eine in polynomieller Zeit berechenbare, polynomiell beschränkte Folge.

Für den Fall der algorithmischen Ununterscheidbarkeit in Satz 37 ist es wichtig, daß die Ensembles in polynomieller Zeit generierbar sind. Auf diese Voraussetzung kann nicht verzichtet werden, wie wir im folgenden beweisen. Falls die Ensembles nicht in polynomieller Zeit generierbar sind, kann in der Samplefolge Information enthalten sein, die dem Unterscheider hilft,

die Ensembles zu trennen. Wir konstruieren nun zwei Ensembles mit der Eigenschaft, daß die Ensembles algorithmisch ununterscheidbar sind, wenn der Unterscheider ein Sample erhält, die aber unterscheidbar werden, wenn der Unterscheider zwei Samples oder mehr erhält. Dabei werden die Samples wie in Definition 36 unabhängig voneinander erzeugt.

38. Lemma *Sei $\mathcal{M}$ eine abzählbare Menge von Turingtransducern. Es existiert eine Folge $\{S_n\}_{n\in\mathbb{N}}$ von Mengen $S_n \subseteq \Sigma^n$ mit folgenden Eigenschaften:*

1. *Für alle Konstanten $0 \leq \varepsilon < 1$ gibt es eine Konstante $c > 0$, so daß für alle $n \in \mathbb{N}$, $n > c$ gilt*
$$|S_n| \geq 2^{\varepsilon n}.$$
2. *Für alle Folgen von Mengen $\{A_n\}_{n\in\mathbb{N}}$, $\{B_n\}_{n\in\mathbb{N}}$ mit $A_n, B_n \subseteq S_n$ und $|A_n| = |B_n| = k_n \in \mathbb{N}$ sind die Ensembles $U = \{U_n\}_{n\in\mathbb{N}}$, $V = \{V_n\}_{n\in\mathbb{N}}$ mit*
$$U_n(x) = \frac{1}{k_n}[\![x \in A_n]\!] \text{ und } V_n(x) = \frac{1}{k_n}[\![x \in B_n]\!]$$
$\mathcal{M}$-ununterscheidbar.

Beweis: Wir benutzen ein Diagonalisierungsargument: Sei $\{M_i\}_{i\in\mathbb{N}}$ eine Aufzählung aller Maschinen in $\mathcal{M}$. Betrachte die Funktion $a_n : \Sigma^n \to \mathbb{R}^{\lfloor \log n \rfloor}$, die einer Eingabe $x \in \Sigma^n$ das Tupel
$$a_n(x) = \Big(\Pr(M_1(<1^n, x>) = 1), \ldots, \Pr(M_{\lfloor \log n \rfloor}(<1^n, x>) = 1)\Big) \tag{18}$$
zuordnet. Die Bilder $a_n(\Sigma^n)$ liegen alle im $\lfloor \log n \rfloor$ dimensionalen Einheitswürfel. Nun unterteilen wir jede Komponente des Vektors und damit den Würfel gleichmäßig in r Intervalle $[0, \frac{1}{r}[, [\frac{1}{r}, \frac{2}{r}[, \ldots, [1 - \frac{1}{r}, 1]$. Dabei wählen wir $r = n^{\log n} = 2^{\log^2 n}$. Die Abbildung a_n ordnet jeder Eingabe eindeutig einen der $r^{\log n} = 2^{\log^3 n}$ verschiedenen Würfel zu. Nach dem Schubfachprinzip gibt es damit einen Würfel der Kantenlänge r, dem mindestens $2^{n-\log^3 n}$ Strings der Länge n zugeordnet werden. Sei S_n die Menge dieser Strings. Für alle Konstanten $0 \leq \varepsilon < 1$ gibt es eine Konstante $c > 0$, so daß für alle $n \in \mathbb{N}$, $n > c$ gilt $-(1-\varepsilon)n \leq -\log^3 n$, und es folgt die erste Behauptung des Lemmas.

Seien $\{U_n\}_{n\in\mathbb{N}}$ und $\{V_n\}_{n\in\mathbb{N}}$ wie in der Voraussetzung und sei der Turingtransducer M ein Unterscheider. Nach Konstruktion gibt es $i \in \mathbb{N}$, so daß $\Pr(M(x) = 1) = \Pr(M_i(x) = 1)$ für alle Eingaben $x \in \Sigma^*$ gilt. O.B.d.A. sei im folgenden $n > 2^i$. Seien $u_1, u_2, \ldots, u_{k_n}$ und $v_1, v_2, \ldots, v_{k_n}$ die Elemente von $\mathrm{Supp}(U_n)$ und $\mathrm{Supp}(V_n)$ in einer beliebigen Reihenfolge. Wir benutzen die Schreibweise $T_n = \{(u_i, v_i) : 1 \leq i \leq k_n\}$.

$$\begin{aligned}
&\Big|\sum_{x\in\Sigma^*} \Pr(M(<1^n, x>) = 1)(U_n(x) - V_n(x))\Big| \\
&= \frac{1}{k_n}\Big|\sum_{x\in A_n} \Pr(M_i(<1^n, x>) = 1) - \sum_{x\in B_n} \Pr(M_i(<1^n, x>) = 1)\Big| \\
&\leq \frac{1}{k_n}\sum_{(x,y)\in T_n} \Big|\Pr(M_i(<1^n, x>) = 1) - \Pr(M_i(<1^n, y>) = 1)\Big| \\
&\leq \frac{1}{rk_n}|T_n| = \frac{1}{r}.
\end{aligned}$$

Da r superpolynomiell wächst, sind die Ensembles U und V $\mathcal{M}$-ununterscheidbar. □

Weil die Mengen der Folge $\{S_n\}_{n\in\mathbb{N}}$ aus Lemma 38 exponentielle Größe haben, kann man durch ein Abzählargument die Existenz von Paaren von Strings mit zusätzlichen Eigenschaften in den Mengen S_n folgern. Die Wahrscheinlichkeitsverteilung, die eine probabilistische Turingmaschine bei Eingabe eines solchen Strings generiert, liefert keine Information. Sind aber beide Strings zugänglich, so können sie verglichen werden und machen auf diese Weise die Paare unterscheidbar.

39. Satz *Es gibt (in* EXPSPACE*) berechenbare algorithmisch ununterscheidbare Ensembles U und V, die algorithmisch unterscheidbar werden, wenn der Unterscheider zwei unabhängig voneinander gezogene Samples erhält. Das heißt, U^2 und V^2 sind algorithmisch unterscheidbar.*

Beweis: Sei $S \subseteq \Sigma^n$ eine Menge mit $|S| \geq 2^k + 1$, für $k \in \mathbb{N}$. Dann gibt es Strings $x, x' \in S$, $x \neq x'$, so daß x und x' auf den ersten (oder letzten) k Bits übereinstimmen. Der Beweis folgt aus dem Schubfachprinzip: Wir haben 2^k Schubfächer entsprechend den Bitfolgen der ersten (oder letzten) k Stellen aber $2^k + 1$ Strings.

Sei PolyTM die Menge aus Definition 4. Betrachten wir nun die Folge der Mengen $\{S_n\}_{n\in\mathbb{N}}$, die in Lemma 38 für PolyTM konstruiert wird: Nach Lemma 38 gilt (mit $\varepsilon = \frac{3}{4}$) für hinreichend große $n \in \mathbb{N}$, daß $|S_n| \geq 2^{\frac{3}{4}n}$. Wir teilen nun S_n in zwei gleichmächtige, disjunkte Mengen X_n und Y_n auf. Für $n > 8$ gilt nun $\frac{1}{2}n + 1 < \varepsilon n - 1$. Das heißt, daß $|X_n| > 2^{\frac{1}{2}n+1}$ und $|Y_n| > 2^{\frac{1}{2}n+1}$ gilt. Dann gibt es aber Strings $x, x' \in X_n$, $x \neq x'$, die auf mehr als der linken Hälfte und $y, y' \in Y_n$, $y \neq y'$, die auf mehr als der rechten Hälfte übereinstimmen. Nimmt man nun die Mengen $A_n = \{x, x'\}$ und $B_n = \{y, y'\}$ als Träger der Wahrscheinlichkeitsverteilungen für Eingaben der Länge n, so sind die Ensembles nach Lemma 38 algorithmisch ununterscheidbar. Hat man aber 2 Samples zur Verfügung, so sind diese mit Wahrscheinlichkeit $\frac{1}{2}$ verschieden und man kann eindeutig bestimmen von welcher Verteilung sie stammen, je nachdem ob sie auf der linken oder der rechten Hälfte identisch sind. Der Unterscheider M gibt 1 aus, wenn beide Eingaben verschieden und auf mehr als der linken Hälfte identisch sind und sonst 0. M unterscheidet die Ensembles unendlich oft mit algorithmischem Unterschied $\frac{1}{4}$.

Mit den Methoden aus Proposition 10 und Korollar 13 lassen sich die Akzeptanzwahrscheinlichkeiten der Maschinen in PolyTM mit superpolynomiellem Platzbedarf berechnen. Damit gibt es aber auch eine in EXPSPACE berechenbare Folge von Mengen mit den Eigenschaften aus Lemma 38, und es folgt die Behauptung des Satzes. □

40. Korollar *Die in Satz 39 konstruierten Ensembles U und V haben folgende Eigenschaften:*

- *U und V können nicht in Polynomzeit erzeugt werden,*
- *U und V sind algorithmisch ununterscheidbar aber schaltkreisunterscheidbar.*

Beweis: In Satz 39 wird die Existenz von zwei Ensembles gezeigt, die bei zweimaliger Anwendung algorithmisch unterscheidbar werden. Mit Satz 37 folgt die erste Behauptung.

Weil die Träger der Wahrscheinlichkeitsverteilungen der Ensembles U und V sparse sind, gibt es nach Satz 27 Schaltkreisfamilien polynomieller Größe, die die Sprachen $\mathrm{Supp}(U)$ und $\mathrm{Supp}(V)$ entscheiden. Diese Schaltkreisfamilien sind aber auch Unterscheider für U und V. □

Es stellt sich nun die Frage, ob die Konstruktion aus Satz 39 auch zur Trennung von k und ℓ Samples mit $2 \leq k < \ell$ verallgemeinert werden kann. Es wird sich zeigen, daß eine solche Verallgemeinerung der Beweistechnik von Satz 39 nicht möglich ist. Dazu machen wir zunächst folgende Beobachtung:

41. Korollar *Sei $\mathcal{M}$ die Menge aller probabilistischen Turingmaschinen (ohne Laufzeitbeschränkungen und ohne Annahmen über die Haltewahrscheinlichkeiten). Es gibt Ensembles U und V, so daß U und V $\mathcal{M}$-ununterscheidbar aber U^2 und V^2 $\mathcal{M}$-unterscheidbar sind.*

Die Behauptung folgt sofort, weil man die Konstruktion aus Satz 39 auch für die Menge $\mathcal{M}$ durchführen kann. Es werden Ensembles U und V konstruiert, die bezüglich jeder probabilistischen Turingmaschine ununterscheidbar aber trotzdem schaltkreisunterscheidbar sind. Allerdings sind dann die Sprachen $\mathrm{Supp}(U)$ und $\mathrm{Supp}(V)$ nicht mehr Turing-entscheidbar: Gäbe es nämlich eine Turingmaschine M, die zum Beispiel die Sprache $\mathrm{Supp}(U)$ entscheidet, dann wäre auch $M \in \mathcal{M}$. Aber die Maschine M könnte U und V unterscheiden.

Die Konstruktion aus Satz 39 kann allgemein für jede beliebige probabilistische Turingmaschine durchgeführt werden, weil lediglich die Information über die Kardinalität der Mengen der Folge $\{S_n\}_{n\in\mathbb{N}}$ in Lemma 38 ausgenutzt wird. Es wird keine strukturelle Information über die Menge $\mathcal{M}$ der Maschinen verwendet. Andererseits werden wir im folgenden zeigen, daß bei einer Trennung von k und ℓ Samples die Kardinalität der Mengen der Folge $\{S_n\}_{n\in\mathbb{N}}$ beschränkt bleibt, wenn wir keine strukturelle Informationen über die Menge $\mathcal{M}$ der Maschinen verwenden. Damit ist dann das in Satz 39 benutzte kombinatorische Argument nicht mehr anwendbar:

Wir benötigen folgende Schreibweise: Sei S eine Menge und $n \in \mathbb{N}$. Dann ist $K_n(S) = \{T \subseteq S \ : \ |T| = n\}$ das System aller n-elementigen Teilmengen von S. Analog zu (18) definieren wir eine Abbildung $a'_n : K_k(\Sigma^n) \to \mathbb{R}^{g(n)}$ mit $g \in \omega(1)$. Wenn wir den Wertebereich wie in Lemma 38 gleichmäßig unterteilen, können wir die Abbildung a'_n als eine Färbung $\alpha_n : K_k(\Sigma^n) \to \{1, \ldots, c_n\}$ auffassen, wobei die Folge $\{c_i\}_{i\in\mathbb{N}}$ superpolynomiell wächst. Um die Konstruktion aus Satz 39 anzuwenden, benötigen wir eine Folge $\{S_n\}_{n\in\mathbb{N}}$ von Mengen $S_n \subseteq \Sigma^n$ wachsender Kardinalität mit $|\alpha_n(K_k(S_n))| = 1$ für alle $n \in \mathbb{N}$.

42. Lemma *Seien $k, n \in \mathbb{N}$ mit $k \geq 2$. Dann gibt es eine Färbung $\beta : K_k(\Sigma^n) \to \{1, \ldots, n\}$ mit der Eigenschaft, daß die Kardinalität aller Mengen $S \subseteq \Sigma^n$ mit $|\beta(K_k(S))| = 1$ höchstens k beträgt.*

Beweis: Sei $\sigma \in \Sigma^n$. Dann bezeichnet $\sigma(i), 1 \leq i \leq n$, das i-te Bit des Strings σ. Die Färbung β wird folgendermaßen definiert: Sei $Y = \{y_1, \ldots, y_k\} \subseteq \Sigma^n$ und gelte $y_1 < y_2 < \ldots < y_k$ bezüglich der lexikographischen Ordnung auf Σ^n. Dann gilt

$$\beta(Y) = \min\{1 \leq i \leq n \ : \ y_1(i) \neq y_2(i)\}.$$

Seien $S \subseteq \Sigma^n$ mit $\beta(K_k(S)) = \{i\} \subseteq \{1, \ldots, n\}$ und $|S| = k + 1$, und seien $s_1, s_2, s_3 \in S$ die drei lexikographisch kleinsten Strings in S mit $s_1 < s_2 < s_3$. Es gilt nun

$$\left.\begin{array}{ll} \beta(S - \{s_3\}) = i & \Rightarrow s_1(i) \neq s_2(i) \\ \beta(S - \{s_2\}) = i & \Rightarrow s_1(i) \neq s_3(i) \end{array}\right\} \Rightarrow s_2(i) = s_3(i)$$

Damit gilt aber auch $\beta(S - \{s_1\}) \neq i$ im Widerspruch zur Voraussetzung. □

Eine Beweismethode, die Satz 39 verallgemeinert, muß daher weitere, strukturelle Eigenschaften der Folge von Mengen $\{S_n\}_{n\in\mathbb{N}}$ berücksichtigen. Es ist nicht bekannt, ob es Konstanten $k, \ell \in \mathbb{N}$ mit $2 \leq k < \ell$ gibt, so daß k und ℓ Samples getrennt werden können.

7 Ein neuer Sicherheitsbegriff

Ein interaktives Beweissystem, das die Zero-Knowledge-Eigenschaft hat, wird als ein kryptographisch (theoretisch) sicheres Verfahren angesehen. Um die Zero-Knowledge-Eigenschaft nachzuweisen, muß man zeigen, daß die Familie von Wahrscheinlichkeitsverteilungen über den ausgetauschten Nachrichten der beiden Kommunikationspartner ununterscheidbar ist von der Familie von Wahrscheinlichkeitsverteilungen, die ein Simulator generiert. Dabei muß der Simulator die gleiche Berechnungskomplexität wie der gegnerische Algorithmus haben, der die ausgetauschten Nachrichten der beiden Kommunikationspartner abhört, und der Simulator muß sein Ensemble ohne die Geheiminformationen der Kommunikationspartner generieren.

Durch den Test auf Ununterscheidbarkeit wird aber lediglich eine „Ähnlichkeitsrelation“ auf den Ensembles definiert. Die Ununterscheidbarkeit von Ensembles ist nicht sehr intuitiv, wenn man testen will, wie sehr die abgehörten Nachrichten, die Berechnungskomplexität des gegnerischen Algorithmus erhöhen. Man kann ein kryptographisches Verfahren auch dann als (theoretisch) sicher betrachten, wenn die abgehörten Nachrichten dem Gegner nur Berechnungen ermöglichen, die er auch ohne die abgehörten Nachrichten durchführen kann.

Dieser Sicherheitsbegriff hängt von dem Berechnungsmodell ab, das dem gegnerischen Algorithmus zu grunde liegt. Wir werden im folgenden Sprachentscheidungsprobleme für probabilistische Maschinen mit beschränkter Fehlerwahrscheinlichkeit als Berechnungsmodell des Gegners betrachten. Wir modellieren folgende Situation: Ein Gegner hört die unsichere Leitung zweier Kommunikationspartner ab, ohne die Kommunikation zu beeinflussen. Die Eingabe des Gegners besteht aus der gemeinsamen Eingabe der Kommunikationpartner und den abgehörten Nachrichten. Dabei unterliegen die Nachrichten den Wahrscheinlichkeitsverteilungen des Ensembles, das von den Kommunikationspartnern generiert wird und das mit der gemeinsamen Eingabe indiziert ist.

Abstrakt gesehen betrachten wir, wie sich das Akzeptanzverhalten von probabilistischen Maschinen ändert, wenn sie zusätzlich zu ihrer Eingabe ein (oder mehrere) gemäß einer Wahrscheinlichkeitsverteilung generierte Samples als Eingabe erhalten.

43. Definition *Seien $S \subseteq \Sigma^*$ eine Sprache, $\mathcal{M}$ eine Menge von probabilistischen Turingmaschinen und sei $E = \{E_x\}_{x \in S}$ ein Ensemble. Wir schreiben* $\mathrm{BP}(\mathcal{M}, E)$ *für die Menge aller Sprachen $L \subseteq S$, für die es eine Konstante $\varepsilon > 0$, eine Turingmaschine $M \in \mathcal{M}$ und ein Polynom $p \in \mathbb{N}[X]$ gibt, so daß für alle $x \in S$ gilt:*

$$\sum_{y \in \Sigma^*} E_x^{p(|x|)}(y) \Pr(M(< x, y >) = \chi_L(x)) \geq \frac{1}{2} + \varepsilon.$$

Dabei darf die Laufzeit der Maschine M nur von der Länge der Eingabe x abhängen. Die Konstante ε heißt die Fehlerschranke von M.

Definition 43 beschreibt die Menge der Sprachen, die von den Maschinen aus $\mathcal{M}$ mit beschränkter Fehlerwahrscheinlichkeit entschieden werden können, wenn die Maschinen zusätzlich zur Eingabe $x \in S$ polynomiell viele Samples gemäß E erhalten. Weil in unserem Modell der Gegner passiv ist, keinen Einfluß auf das Ensemble hat, sind die Samples unabhängig voneinander.

Zusätzlich werden wir folgende Schreibweise benutzen: Sei $\mathcal{M}$ eine Menge von probabilistischen Turingmaschinen und sei $\mathcal{C} = \mathrm{BP} \cdot \mathcal{M}$ die Menge der Sprachen, die von den Maschinen aus $\mathcal{M}$ mit beschränkter Fehlerwahrscheinlichkeit entschieden werden können. Dann identifizieren wir die Maschinen mit den von ihnen entschiedenen Sprachen und schreiben auch $\mathrm{BP}(\mathcal{C}, E)$ für die Klasse $\mathrm{BP}(\mathcal{M}, E)$.

44. Definition *Seien U, V Ensembles und $\mathcal{M}$ eine Menge von probabilistischen Turingmaschinen. U* verrät keine $\mathcal{M}$-Information, *wenn $\mathrm{BP}(\mathcal{M}, U) \subseteq \mathrm{BP} \cdot \mathcal{M}$ gilt. U* verrät weniger $\mathcal{M}$-Information als V, *wenn $\mathrm{BP}(\mathcal{M}, U) \subseteq \mathrm{BP}(\mathcal{M}, V)$ gilt. U und V haben* die gleiche $\mathcal{M}$-Information, *wenn $\mathrm{BP}(\mathcal{M}, U) = \mathrm{BP}(\mathcal{M}, V)$.*

Analoge Relationen lassen sich für Sprachklassen definieren.

Aus den Definitionen 43 und 44 erhalten wir:

45. Proposition *Sei $\mathcal{M}$ eine Menge von probabilistischen Turingmaschinen, und seien U und V Ensembles:*

- *Sind U und V perfekt ununterscheidbar, so verraten U und V die gleiche $\mathcal{M}$-Information.*
- *Gibt es eine probabilistische Turingmaschine $M \in \mathcal{M}$, so daß M das Ensemble U induziert und ist $\mathcal{M}$ unter Konkatenation abgeschlossen, so verrät U keine $\mathcal{M}$-Information.*
- *Gibt es eine probabilistische Turingmaschine $M \in \mathcal{M}$ mit $U = M(V)$ und ist $\mathcal{M}$ unter Konkatenation abgeschlossen, so verrät U weniger $\mathcal{M}$-Information als V.*

Wenn eine Sprachklasse $\mathcal{C}$ mächtig genug ist, um Tests polynomiell oft zu wiederholen, kann man die Fehlerschranke exponentiell klein wählen.

46. Definition *Seien $A, B \subseteq \Sigma^*$ Sprachen. A heißt* mehrheitsreduzierbar auf B, *wenn es eine in polynomieller Zeit berechenbare Funktion $f : \Sigma^* \to \Sigma^*$ mit $f(x) =< y_1, \ldots, y_k >$, $k \in \mathbb{N}$, gibt, für die gilt:*

$$x \in A \Leftrightarrow \left|\{i \ : \ y_i \in B, 1 \leq i \leq k\}\right| > \frac{k}{2}.$$

47. Lemma *Sei $\mathcal{M}$ eine Menge von probabilistischen Turingmaschinen und sei $\mathrm{BP} \cdot \mathcal{M}$ unter Mehrheitsreduktionen abgeschlossen. Seien $S \subseteq \Sigma^*$ eine Sprache und $E = \{E_x\}_{x \in S}$ ein Ensemble. Für jede Sprache $L \in \mathrm{BP}(\mathcal{M}, E)$ und für jedes Polynom $p \in \mathbb{N}[X]$ gibt es eine Maschine $M \in \mathcal{M}$ und ein Polynom $q \in \mathbb{N}[X]$, so daß für alle $x \in S$ gilt:*

$$\sum_{y \in \Sigma^*} E_x^{q(|x|)}(y) \Pr(M(< x, y >) = \chi_L(x)) \geq 1 - 2^{-p(|x|)}.$$

Beweis: Der Beweis geht analog zu [16] Proposition 2.23, Seite 75 f. Die Anzahl der Samples und Zufallsbits wächst polynomiell in der Länge der Eingabe. □

Als Anwendung von Lemma 47 erhalten wir:

48. Korollar

1. *Es gibt ein Ensemble U mit* $\mathrm{Supp}(U) \notin \mathrm{BP}(\mathrm{PolyTM}, U)$.
2. *Sei $L \in$ SPARSE eine Sprache. Dann gibt es ein Ensemble $V = \{V_n\}_{n\in\mathbb{N}}$, so daß $L \in \mathrm{BP}(\mathrm{PolyTM}, V)$ gilt.*
3. *Für jedes Ensemble $U = \{U_n\}_{n\in\mathbb{N}}$ gibt es eine unendliche Folge $U_1, U_2, \ldots$ von Ensembles $U_i = \{U_{i,n}\}_{n\in\mathbb{N}}$ mit $U = U_1$, so daß für alle $j \in \mathbb{N}$ gilt:*

$$\mathrm{BP}(\mathrm{PolyTM}, U_j) \subsetneq \mathrm{BP}(\mathrm{PolyTM}, U_{j+1}) \subsetneq \mathrm{P/Poly}.$$

Beweis: Der Beweis ist in fünf aufeinander aufbauende Teile gegliedert:
Hilfsbehauptung 1: *Sei $U = \{U_n\}_{n\in\mathbb{N}}$ ein beliebiges Ensemble. Dann gilt*

$$\mathrm{BP}(\mathrm{PolyTM}, U) \subsetneq \mathrm{P/Poly}.$$

Beweis: Wir zeigen zunächst, daß $\mathrm{BP}(\mathrm{PolyTM}, U) \subseteq \mathrm{P/Poly}$ gilt und konstruieren dann mit Diagonalisierung eine Sprache $L \in \mathrm{P/Poly} - \mathrm{BP}(\mathrm{PolyTM}, U)$. Analog zum Beweis BPP $\subseteq$ P/Poly (siehe zum Beispiel [2], Corollary 6.3), kann man mit Lemma 47 die Fehlerwahrscheinlichkeit so sehr verringern, daß es Folgen von Zufallsbits und Folgen von Samples gibt, die für alle Eingaben einer festen Länge geeignet sind. Das heißt, daß die Maschine alle Eingaben einer festen Länge mit diesen Folgen korrekt entscheidet. Diese Folgen von Zufallsbits und Samples sind dann der Advicestring.

Die Sprache $L \in \mathrm{P/Poly} - \mathrm{BP}(\mathrm{PolyTM}, U)$ wird nun folgendermaßen konstruiert: Sei $C_1, C_2, \ldots$ eine beliebige Aufzählung aller Sprachen in $\mathrm{BP}(\mathrm{PolyTM}, U)$. Eine solche Aufzählung existiert, weil die Menge aller probabilistischen Turingmaschinen mit polynomieller Laufzeit aufzählbar ist. Für Wörter der Länge $n \in \mathbb{N}$ und für alle $1 \leq i \leq n$ wird das bezüglich der lexikographischen Ordnung i-te Wort x genau dann in die Sprache L aufgenommen, wenn $x \notin C_i$ gilt. Nach Konstruktion ist die Sprache L sparse und von allen Sprachen in $\mathrm{BP}(\mathrm{PolyTM}, U)$ verschieden. Mit Satz 27 folgt $L \in \mathrm{P/Poly}$.

Beweis von 1. : Die Existenz eines Ensembles U mit $\mathrm{Supp}(U) \notin \mathrm{BP}(\mathrm{PolyTM}, U)$ folgt direkt aus Hilfsbehauptung 1: Wir benötigen dazu das P/Poly Random-Set L aus Korollar 33 und definieren das Ensemble $U = \{U_n\}_{n\in\mathbb{N}}$ durch $U_n(x) = \frac{1}{|L^{=n}|}[\![x \in L^{=n}]\!]$ für alle $n \in \mathbb{N}$ und $x \in \Sigma^*$. Wegen Hilfsbehauptung 1 sind dann alle Sprachen, die probabilistische Turingmaschinen in polynomieller Zeit mit Hilfe dieses Ensembles entscheiden können, in P/Poly. Weil L aber ein P/Poly Random-Set ist, gibt es für jede Fehlerschranke $\varepsilon > 0$, für jedes Polynom $p \in \mathbb{N}[X]$ und für jede probabilistische Turingmaschine M mit polynomieller Laufzeit unendlich viele Eingaben $x \in \Sigma^*$, für die gilt

$$\Big| \sum_{y\in\Sigma^*} U_x^{p(|x|)}(y)\mathrm{Pr}(M(<x,y>) = \chi_L(x)) - \frac{1}{2} \Big| \leq \varepsilon.$$

Das heißt aber nach Definition 43, daß $L \notin \mathrm{BP}(\mathrm{PolyTM}, U)$.

Beweis von 2. : Sei $L \in$ SPARSE beliebig und sei $p \in \mathbb{N}[X]$ ein Polynom mit $|L^{=n}| \leq p(n)$ für alle $n \in \mathbb{N}$. Wir wählen $V = \{V_n\}_{n\in\mathbb{N}}$ mit $V_n(x) = \frac{1}{|L^{=n}|}[\![x \in L^{=n}]\!]$. Erhält eine probabilistische

Turingmaschine zum Beispiel $p^2(|x|)$ Samples gemäß V zur Eingabe x, so konvergiert die Wahrscheinlichkeit, daß die Menge der Samples ganz $L^{=n}$ umfaßt, schneller als n^{-k} gegen 1 für alle $k \in \mathbb{N}$. Damit erhält die Maschine mit großer Wahrscheinlichkeit $L^{=|x|}$ als Eingabe und kann daher die Sprache $L = \mathrm{Supp}(V)$ korrekt entscheiden.

Hilfsbehauptung 2: *Für jedes Ensemble* $U = \{U_n\}_{n\in\mathbb{N}}$ *gibt es ein Ensemble* $V = \{V_n\}_{n\in\mathbb{N}}$, *so daß* $\mathrm{BP}(\mathrm{PolyTM}, V) \not\subseteq \mathrm{BP}(\mathrm{PolyTM}, U)$.
Beweis: Aus der Hilfsbehauptung 1 wissen wir, daß eine sparse Sprache

$$L \in \mathrm{P/Poly} - \mathrm{BP}(\mathrm{PolyTM}, U)$$

existiert. Im Beweis zur zweiten Behauptung des Korollars wird ein Ensemble V konstruiert mit $L \in \mathrm{BP}(\mathrm{PolyTM}, V)$. Damit folgt die Behauptung.

Beweis von 3. : Sei $U = \{U_n\}_{n\in\mathbb{N}}$ ein beliebiges Ensemble. Wir konstruieren jetzt durch vollständige Induktion die Folge $U_1, U_2, \ldots$ von Ensembles. Sei U_j, $j \in \mathbb{N}$, bereits konstruiert. Aus der Hilfsbehauptung 2 folgt die Existenz eines Ensembles $V = \{V_n\}_{n\in\mathbb{N}}$, so daß $\mathrm{BP}(\mathrm{PolyTM}, V) \not\subseteq \mathrm{BP}(\mathrm{PolyTM}, U_j)$. Betrachte nun das Ensemble $U_{j+1} = \{U_{j+1,n}\}_{n\in\mathbb{N}}$ mit

$$U_{j+1,n}(x) = \begin{cases} \frac{1}{2}U_{j,n}(y) & \text{falls } x = 0y, \\ \frac{1}{2}V_n(y) & \text{falls } x = 1y, \\ 0 & \text{sonst.} \end{cases}$$

Werden Samples gemäß des Ensembles U_{j+1} generiert, so ist ein Sample jeweils mit Wahrscheinlichkeit $\frac{1}{2}$ von V oder von U_j. Außerdem kann eine Maschine am ersten Bit des Samples erkennen, gemäß welchem Ensemble das Sample generiert wurde. Im Mittel genügen $k \in \mathbb{N}$ Samples gemäß U_{j+1}, um mit Wahrscheinlichkeit $1 - \frac{1}{2^k}$ ein Sample von V oder von U_j zu erhalten. Damit gilt $\mathrm{BP}(\mathrm{PolyTM}, V) \cup \mathrm{BP}(\mathrm{PolyTM}, U_j) \subseteq \mathrm{BP}(\mathrm{PolyTM}, U_{j+1})$, und es folgt die Behauptung. □

Mit Definition 43 lassen sich also beliebig viele nichttriviale Sprachklassen definieren. Die Existenz des Ensembles aus der ersten Behauptung des Korollars liegt darin begründet, daß der Zugriff der Maschinen auf das Ensemble im Gegensatz zur normalen Orakelturingmaschine stark eingeschränkt ist. Wenn der Träger des Ensembles eine hohe Komplexität aufweist, können die Maschinen diese Information nicht nutzen.

Wir wollen nun den Zusammenhang zwischen dem neuen Sicherheitsbegriff und Ununterscheidbarkeit untersuchen. Es wird sich zeigen, daß statistisch ununterscheidbare oder schaltkreisununterscheidbare Ensembles bezüglich probabilistischen Turingmaschinen mit polynomieller Laufzeit die gleiche Information verraten.

49. Satz *Sei* $\mathcal{C} \subseteq 2^{\Sigma^*}$ *eine unter endlicher Variation abgeschlossene Sprachklasse. Sind* U *und* V *statistisch ununterscheidbare Ensembles, dann gilt* $\mathrm{BP}(\mathcal{C}, U) = \mathrm{BP}(\mathcal{C}, V)$.

Beweis: Seien $L \in \mathrm{BP}(\mathcal{C}, U)$ eine beliebige Sprache, $p \in \mathbb{N}[X]$ ein Polynom und sei M ein Turingtransducer, der L mit Hilfe von p Samples des Ensembles U entscheidet. Dann gilt:

$$\Big| \sum_{y\in\Sigma^*} (U_x^{p(|x|)}(y) - V_x^{p(|x|)}(y))\mathrm{Pr}(M(< x, y >) = \chi_L(x)) \Big|$$

$$\leq \sum_{y\in\Sigma^*} \left|U_x^{p(|x|)}(y) - V_x^{p(|x|)}(y)\right| \Pr(M(<x,y>) = \chi_L(x))$$

$$\leq \underbrace{\max_{z\in\Sigma^*} \Pr(M(<x,z>) = \chi_L(x))}_{\leq 1} \sum_{y\in\Sigma^*} \left|U_x^{p(|x|)}(y) - V_x^{p(|x|)}(y)\right|. \qquad (19)$$

Weil nach Voraussetzung U und V statistisch ununterscheidbar sind, folgt mit Satz 37, daß nur für endlich viele Eingaben x der Betrag von (19) größer als die Fehlerschranke der Maschine M sein kann. Das heißt, es gibt eine Fehlerschranke $\varepsilon' \in \mathbb{R}$ mit $\varepsilon' < \varepsilon$, so daß M die Sprache L bis auf endlich viele Wörter mit Hilfe von p Samples des Ensembles V entscheidet. Da $\mathcal{C}$ unter endlicher Variation abgeschlossen ist, folgt die Behauptung. Der Beweis für $\mathrm{BP}(\mathcal{C}, V) \subseteq \mathrm{BP}(\mathcal{C}, U)$ geht analog. □

Für die Fälle der algorithmischen und schaltkreisununterscheidbaren Ensembles beweisen wir zunächst folgende Hilfsbehauptung:

50. Lemma *Seien U und V Ensembles. Gilt $\mathrm{BP}(\mathrm{PolyTM}, U) \neq \mathrm{BP}(\mathrm{PolyTM}, V)$, dann gibt es eine Konstante $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$, eine probabilistische Turingmaschine M mit polynomieller Laufzeit, ein Polynom $p \in \mathbb{N}[X]$ und unendlich viele Eingaben $x \in \Sigma^*$, so daß sich die Akzeptanzwahrscheinlichkeit von M bei Eingabe x und $p(|x|)$ Samples gemäß U von der Akzeptanzwahrscheinlichkeit bei Eingabe x und $p(|x|)$ Samples gemäß V um mindestens den Betrag ε unterscheiden.*

Beweis: O.B.d.A. seien $\mathrm{BP}(\mathrm{PolyTM}, U) \not\subseteq \mathrm{BP}(\mathrm{PolyTM}, V)$ und $L \in \mathrm{BP}(\mathrm{PolyTM}, U) - \mathrm{BP}(\mathrm{PolyTM}, V)$ eine Sprache. Sei M eine probabilistische Maschine mit polynomieller Laufzeit, die L mit Hilfe des Ensembles U entscheidet und sei $\varepsilon > 0$ eine Fehlerschranke für M. Wir haben das Problem, daß M zusammen mit dem Ensemble V nicht mehr Definition 43 erfüllen muß. Das heißt, es muß nicht mehr eine von 0 verschiedene Fehlerschranke existieren. Angenommen für jede Konstante $\varepsilon' \in \mathbb{R}$, $\varepsilon' > 0$, gibt es nur endlich viele Eingaben, für die sich die Akzeptanzwahrscheinlichkeiten von M mit U und V um den Betrag ε' unterscheiden. Dann ist jede Konstante kleiner als $\varepsilon - \varepsilon'$ bis auf endlich viele Eingaben eine neue gültige Fehlerschranke. Weil BPP unter endlicher Variation abgeschlossen ist, folgt dann $L \in \mathrm{BP}(\mathrm{PolyTM}, V)$ im Widerspruch zur Voraussetzung. Daher existieren also $p \in \mathbb{N}[X]$ und $\varepsilon > 0$, für die es unendlich viele Eingaben x gibt, so daß sich die Akzeptanzwahrscheinlichkeit von M bei Eingabe x und $p(|x|)$ Samples gemäß U von der Akzeptanzwahrscheinlichkeit bei Eingabe x und $p(|x|)$ Samples gemäß V um mindestens ε unterscheiden. □

51. Satz *Seien U und V Ensembles. Sind U und V schaltkreisununterscheidbar, so gilt auch* $\mathrm{BP}(\mathrm{PolyTM}, U) = \mathrm{BP}(\mathrm{PolyTM}, V)$.

Beweis: Wir führen einen indirekten Beweis: Sei $\mathrm{BP}(\mathrm{PolyTM}, U) \neq \mathrm{BP}(\mathrm{PolyTM}, V)$. Der Beweis gliedert sich in folgende Schritte: Wir wissen bereits aus Lemma 50, daß es eine probabilistische Turingmaschine M und unendlich viele Eingaben aus Σ^* gibt, an denen M bezüglich U und V einen konstanten Unterschied ε in der Akzeptanzwahrscheinlichkeit aufweist. Für diese Eingaben existieren Zufallsstrings für die Maschine M, so daß M für jeden dieser Strings einen Unterschied von mindestens ε in den Wahrscheinlichkeiten für U und V aufweist. Es wird

eine Schaltkreisfamilie konstruiert, die die Maschine M mit diesen Zufallsstrings simuliert. Die Schaltkreisfamilie ist ein Unterscheider für U und V.

Seien L, ε, M, p und die Eingaben x wie im Beweis zu Lemma 50. Wir betrachten nun den Fall, daß unendlich viele dieser Eingaben zur Sprache L gehören. Der Fall, daß unendlich viele Eingaben zu $\overline{L}$ gehören, geht analog.

Sei $q \in \mathbb{N}[X]$ ein Polynom, das die Laufzeit von M beschränkt, und sei M' eine Maschine, die bei Eingabe $< x, r > \in \Sigma^*$ wie $M(x)$ arbeitet, wobei die Zufallsbits gemäß r gewählt werden. Dann gilt:

$$\begin{aligned} &\Big| \sum_{y \in \Sigma^*} (U_x^{p(|x|)}(y) - V_x^{p(|x|)}(y)) \Pr(M(< x, y >) = 1) \Big| \\ &\quad \leq \quad 2^{-q(|x|)} \sum_{r \in \Sigma^{q(|x|)}} \sum_{y \in \Sigma^*} \Pr(M'(<< x, y >, r >) = 1) |U_x^{p(|x|)}(y) - V_x^{p(|x|)}(y)|. \end{aligned}$$

Das heißt, daß zu jeder Eingabe $x \in \Sigma^*$, für die ein Unterschied von mindestens ε in der Akzeptanzwahrscheinlichkeit besteht, ein Zufallsstring $r(x) \in \Sigma^{q(|x|)}$ existiert mit

$$\Big| \sum_{y \in \Sigma^*} (U_x^{p(|x|)}(y) - V_x^{p(|x|)}(y)) \Pr(M'(<< x, y >, r(x) >) = 1) \Big| \geq \varepsilon.$$

Wenn wir eine Schaltkreisfamilie polynomieller Größe konstruieren, die die Maschine M' simuliert (Siehe zum Beispiel [2]), so können wir für unendlich viele Eingabelängen einen Repräsentanten x' mit einem Unterschied von mindestens ε finden und den Zufallsstring $r(x')$ in den Advicestring kodieren. Eine solche Schaltkreisfamilie ist ein Unterscheider für $U^{p(|\cdot|)}$ und $V^{p(|\cdot|)}$. Mit Satz 37 folgt die Behauptung. □

Für den Fall der algorithmischen Ununterscheidbarkeit ist die Situation komplizierter, weil nach Satz 39 die Anzahl der Samples, die ein Unterscheider als Eingabe erhält, eine Rolle spielt. Wenn man die Maschine M aus Lemma 50 als Unterscheider nimmt, erhält man folgendes Resultat:

52. Korollar *Seien U und V Ensembles. Gilt* $\mathrm{BP}(\mathrm{PolyTM}, U) \neq \mathrm{BP}(\mathrm{PolyTM}, V)$*, dann gibt es ein Polynom $p \in \mathbb{N}[X]$, so daß $U^{p(|\cdot|)}$ und $V^{p(|\cdot|)}$ algorithmisch unterscheidbar sind.*

Damit haben wir gezeigt, daß für perfekt, statistisch oder schaltkreisununterscheidbare Ensembles U und V die Sprachklassen $\mathrm{BP}(\mathrm{PolyTM}, U)$ und $\mathrm{BP}(\mathrm{PolyTM}, V)$ gleich sind. Für algorithmisch ununterscheidbare Ensembles sind die Sprachklassen gleich, wenn $U^{p(|\cdot|)}$ und $V^{p(|\cdot|)}$ für alle Polynome $p \in \mathbb{N}[X]$ algorithmisch ununterscheidbar sind.

Es ist eine ungelöste Frage, ob auch die „Umkehrung“ dieser Resultate gilt. Man kann leicht Beispiele von Ensembles konstruieren, die unterscheidbar sind aber die gleiche Information verraten: $U = \{U_x\}_{x \in \Sigma^*}$ mit $U_x(y) = [\![y = 1^{|x|}]\!]$ und $V = \{V_x\}_{x \in \Sigma^*}$ mit $V_x(y) = [\![y = 0^{|x|}]\!]$ für alle $y \in \Sigma^*$. Für diese Beispiele gilt, daß es probabilistische Turingmaschinen M, M' mit polynomieller Laufzeit gibt, so daß U von $M(V)$ und V von $M'(U)$ ununterscheidbar sind. Es ist aber nicht bekannt, ob alle Beispiele von dieser einfachen Form sind.

Literatur

[1] Noga Alon, Joel H. Spencer und Paul Erdös. *The Probabilistic Method.* Wiley-Interscience Series in Discrete Mathematics and Optimization, 1992.

[2] José Luis Balcázar, Josep Díaz und Joaquim Gabarró. *Structural Complexity 1.* Springer-Verlag, 1988.

[3] Ingrid Biehl, Johannes Buchmann, Bernd Meyer, Christian Thiel und Christoph Thiel. Tools for proving zero knowledge. In *Advances in Cryptology EUROCRYPT*, S. 356–365, 1992.

[4] A. Borodin, S. Cook und N. Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. *Information and Control*, 58:113–136, 1983.

[5] Michael J. Fischer und Sophia A. Paleologou. On the indistinguishability of probabilistic ensembles. Preliminary Version, March 1994.

[6] John Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.

[7] Oded Goldreich. A note on computational indistinguishability. *Information Processing Letters*, 34:277–281, 1990.

[8] Oded Goldreich und Hugo Krawczyk. Sparse pseudorandom distributions. In *Crypto*, S. 113–127, 1989.

[9] Shafi Goldwasser, Silvio Micali und Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, February 1989.

[10] Yenjo Han, Lane A. Hemaspaandra und Thomas Thierauf. Threshold computation and cryptographic security. Technical Report UR-DCS-TR-93-461, University of Rochester, Department of Computer Science, 1993.

[11] Yenjo Han, Lane A. Hemaspaandra und Thomas Thierauf. Threshold computation and cryptographic security. In *International Symposium on Symbolic and Algebraic Computation*, S. 230–239, 1993.

[12] Ralf Handl. *Sicherheit kryptographischer Protokolle.* Dissertation, Universität des Saarlandes, 1993.

[13] Johan Håstad, Russell Impagliazzo, Leonid A. Levin und Michael Luby. Construction of a pseudo-random generator from any one-way function. Technical Report TR-91-068, International Computer Science Institute Berkeley, December 1991.

[14] John E. Hopcroft und Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[15] Dung T. Huynh. On solving hard problems by polynomial-size circuits. *Information Processing Letters*, 24:171–176, February 1987.

[16] Johannes Köbler, Uwe Schöning und Jacobo Torán. *The Graph Isomorphism Problem.* Progress in Theoretical Computer Science. Birkhäuser, 1993.

[17] Evangelos Kranakis. *Primality and Cryptography.* Wiley-Teubner Series in Computer Science, 1986.

[18] Harry R. Lewis und Christos H. Papadimitriou. *Elements of the Theory of Computation.* Prentice-Hall, 1981.

[19] N. Lynch. On reducibility to complex or sparse sets. *Journal of the Association for Computing Machinery*, 22(3):341–345, July 1975.

[20] Bernd Meyer. Constructive separation of classes of indistinguishable ensembles. In *Structure in Complexity Theory*, S. 198–204, 1994.

[21] Silvio Micali, Charles Rackoff und Bob Sloan. The notion of security for probabilistic cryptosystems. *SIAM Journal on Computing*, 17(2):412–426, April 1988.

[22] Martin Mundhenk und Rainer Schuler. Random languages for nonuniform complexity classes. *Journal of Complexity*, 7:296–310, 1991.

[23] Yair Oren. On the cunning power of cheating verifiers: Some observations about zero knowledge proofs. In *IEEE Symposium on Foundations of Computer Science*, S. 462–471, 1987.

[24] Joel Spencer. Sequences with small discrepancy relative to n events. *Compositio Mathematica*, 47(3):365–392, 1982.

[25] Joel Spencer. Six standard deviations suffice. *Transactions of the American Mathematical Society*, 289(2):679–706, June 1985.

[26] Robert E. Wilber. Randomness and the density of hard problems. In *IEEE Symposium on Foundations of Computer Science*, S. 335–342, 1983.

[27] Andrew C. Yao. Theory and applications of trapdoor functions. In *IEEE Symposium on Foundations of Computer Science*, S. 80–91, 1982.

Stichwortverzeichnis