Zufallsextraktoren für Quellen variierender Qualität

Diplomarbeit von Dominique Unruh

Institut für Algorithmen und Kognitive Systeme (IAKS)
Universität Karlsruhe

Betreuer: Prof. Dr. Thomas Beth
Dr. Jörn Müller-Quade

Erklärung

Ich versichere, diese Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben.

Karlsruhe, den 11. Juli 2003

Dominique Unruh

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Zusammenfassung

Bei der Konstruktion von Zufallsgeneratoren, die auf physikalischen Prozessen basieren, ist es zumeist nicht möglich, gleichverteilte und unabhängige Ausgabebits zu erhalten. Daher ist es nötig, die erzeugten Daten nachzubearbeiten, um guten Zufall zu erhalten (wobei der Begriff Zufall im Sinne von Gleichverteilung verwendet wird). In dieser Arbeit stellen wir ein Verfahren zur Nachbearbeitung vor, die adaptive Extraktion. Dessen zentrale Eigenschaften sind, daß das Verhalten der Zufallsquelle nicht exakt bekannt sein muß, und daß die Menge an erzeugtem guten Zufall sich automatisch der Qualität der Eingabe anpaßt.

Weiterhin stellen wir ein an HMM (*hidden Markov models*) orientiertes Verfahren zur Modellierung von Quellen vor, welches besonders gut mit dem adaptiven Extraktionsverfahren zusammenarbeitet.

Zuletzt behandeln wir noch praktische Aspekte der erarbeiteten Theorie: Wir stellen einen statistischen Test vor, um die über eine Quelle getroffenen Annahmen zu überprüfen, und wir untersuchen eine spezielle physikalische Quelle in Hinblick auf die Nachbearbeitbarkeit mittels unseres Verfahrens.

1.2 Abstract

When constructing random number generators based on physical processes, it is usually not possible to get uniformly and independently distributed output bits. We therefore present a method for postprocessing the output of the random source and generating good randomness, we call that method the adaptive extraction. Its two main features are: First, we do not need to know the source's behaviour in every detail (i.e. it suffices to put some constraints on the distribution of the output). Secondly, the amount of generated good randomness (i.e. nearly uniformly distributed data) is automatically adapted to the quality of the input.

We then present a technique for modelling sources, similar to hidden Markov models, which is especially suited for use with the adaptive extraction.

Finally we investigate some practical aspects of our theory: We present a statistical test to verify assumptions made on the source, and we examine a given physical source with respect to the feasibility of adaptive extraction.

1.3 Überblick

In Kapitel 1 führen wir in das Thema dieser Arbeit ein und stellen die darin erarbeiteten Verfahren vor.

In Kapitel 2 behandeln wir Aspekte der Notation und grundlegende Begriffe wie Entropie und Zufälligkeit. Ein besonders wichtiger Begriff ist hier unter anderem der der min-Entropie, es handelt sich dabei gewissermaßen um den garantierten Informationsgehalt der von einer Quelle ausgegebenen Nachrichten (siehe Definition 2.4).

In Kapitel 3 wird das Leftover Hash Lemma [HILL93] vorgestellt. Dies ist ein Verfahren zur blockweisen Extraktion aus Quellen mit bekannter und nicht verschwindender min-Entropie. Da dieses Verfahren sehr viel initialen Zufall benötigt (die Menge an initialem Zufall ist größer als die an zu bearbeitendem), wird es üblicherweise nicht direkt zur Extraktion eingesetzt, sondern als Komponente komplexerer Verfahren (siehe z. B. [NTS95]).

Wir werden das Leftover Hash Lemma etwas generalisieren und dann im nächsten Kapitel zur Konstruktion unseres Extraktionsverfahrens verwenden.

In Kapitel 4 stellen wir eine Methode vor, wichtige Eigenschaften der vorliegenden Zufallsquelle in einer Funktion wiederzugeben, der Symbolgewichtung.

Mit Hilfe dieser Kennfunktion können wir dann ein Extraktionsverfahren formulieren, welches sehr wenig Voraussetzungen an die Quelle stellt (insbesondere wird keine nicht verschwindende min-Entropie vorausgesetzt). Dieses Verfahren analysiert jeden zu bearbeitenden Block und extrahiert aus diesem je nach Eignung mehr oder weniger Zufall. Daher nennen wir dieses Verfahren adaptive Extraktion.

Wir schließen das Kapitel mit einigen beispielhaften Symbolgewichtungen ab.

In Kapitel 5 untersuchen wir eine Modellierungsmethode für Zufallsquellen, welche eine Erweiterung des Konzepts der HMM (*hidden Markov models*) darstellt. Für so modellierte Quellen können wir dann ein Berechnungsverfahren für die Symbolgewichtung angeben und somit aus ihnen adaptiv Zufall extrahieren.

Auch für CHMM geben wir einige Beispiele zusammen mit den zugehörigen Symbolgewichtungen an.

In Kapitel 6 prüfen wir, inwiefern die für das Resultat der adaptiven Extraktion gezeigten Eigenschaften in formale Sicherheitsbeweise für kryptographische Protokolle eingearbeitet werden können. Hierbei legen wir unserer Augenmerk auf vergleichende Sicherheitsbegriffe, welche ein Protokoll dann als sicher bezeichnen, wenn es von einer gewissen Referenzfunktionalität ununterscheidbar ist.

In Kapitel 7 reißen wir Möglichkeiten an, wie bei einer vorgegebenen Quelle mit statistischen Tests die postulierten Eigenschaften der Quelle geprüft werden können. Insbesondere entwickeln wir einen Test für die Symbolgewichtung.

In Kapitel 8 wird zunächst ein im Rahmen dieser Arbeit entstandenes Testprogramm vorgestellt, welches es ermöglicht, einige der hier präsentierten Methoden auszuprobieren. Es eignet sich aber nicht für die Anwendung in der Praxis, hierzu sind optimierte und wenn möglich verifizierte Programme vonnöten.

Danach wird eine an der LMU München implementierte physikalische Quelle untersucht und die praktische Anwendbarkeit der hier entwickelten Verfahren auf diese Quelle aufgezeigt (sowohl durch eine Modellierung als CHMM, als auch durch die Schätzung der Symbolgewichtung mittels statistischer Methoden).

Die Beweise zu den Aussagen in den obigen Kapiteln sind ausgelagert, um das flüssige Lesen dieser Arbeit zu vereinfachen. Sie finden sich in Anhang A und sind von den zugehörigen Sätzen aus referenziert.

1.4 Bisherige Ergebnisse

Im folgenden betrachten wir einige bereits bekannte Ergebnisse auf dem Gebiet der Zufallsextraktion.

1.4.1 Extraktion perfekten Zufalls

Ein frühes Extraktionsverfahren wird in [vN51] beschrieben, im folgenden die Von-Neumann-Extraktion genannt. Hier wird davon ausgegangen, daß eine Quelle vorliegt, die unabhängig identisch verteilte Zufallsbits liefert, jedoch ist die Verteilung der einzelnen Bits unbekannt. Das Verfahren sieht jetzt vor, jeweils Paare von Bits zu betrachten. Sind die beiden Bits gleich, so wird das Paar verworfen, ansonsten wird das erste der beiden Bits ausgegeben. Da die Paare 01 und 10 gleich wahrscheinlich sind, erhalten wir eine gleichverteilte Ausgabe. In der hier vorgestellten Form werden noch unnötig viele Zufallsbits verworfen, in z. B. [Eli72] oder [Per92] wird erläutert, wie das Verfahren erweitert werden kann, so daß asymptotisch der gesamte der Folge innewohnende Zufall extrahiert wird.

Die Von-Neumann-Extraktion hat die folgenden Vorteile:

- Der resultierende Zufall ist perfekt zufällig (d. h. gleichverteilt, nicht nur annähernd gleichverteilt).
- Es wird kein initialer Zufall benötigt (damit ist eine kleine Menge an perfektem Zufall, die vom Extraktionsverfahren zusätzlich verwendet wird, gemeint).

Leider hat dieses Verfahren auch den großen Nachteil, daß es nur auf eine sehr kleine Klasse von Quellen anwendbar ist. Damit ist es für die meisten Anforderungen nicht geeignet.

Für Quellen, die als Markov-Prozeß n-ter Ordnung mit unbekannten Transitionswahrscheinlichkeiten beschrieben werden können, wurde in [Blu86] ein Extraktionsverfahren angegeben. Es teilt die Vorteile der Von-Neumann-Extraktion (perfekte Zufälligkeit, kein initialer Zufall), die betrachtete Klasse von Quellen ist aber wesentlich größer. Da unabhängig identisch verteilte Quellen das gleiche sind wie Markov-Prozesse nullter Ordnung, ist das Verfahren aus [Blu86] echt allgemeiner als die Von-Neumann-Extraktion.

1.4.2 Extraktion guten Zufalls

Für die exakte Simulation physikalischer Quellen sind Markov-Prozesse nicht geeignet, wie das folgende Beispiel zeigen soll: Man stelle sich eine Quelle vor, die anhand eines komplexen, chaotischen physikalischen Prozesses eine Wahrscheinlichkeit aus $I := [\frac{1}{2} - \delta, \frac{1}{2} + \delta]$ wählt und mit dieser dann eine 1 ausgibt. Um dies mit einem Markov-Prozeß zu modellieren, müßte man den chaotischen Prozeß auch als Markov-Prozeß beschreiben, was wohl nicht möglich ist.

Daher wird man über die Quelle einfach nur aussagen, daß die Wahrscheinlichkeit für die Ausgabe einer 1, gegeben alle zuvor getätigten Ausgaben, in I liegt. So beschriebene Familien von Quellen wurden in [SV86] als *slightly random sources* mit Parameter δ eingeführt. Dort wurde auch ein Extraktionsverfahren angegeben, welches beliebig guten Zufall ausgibt (d. h. beliebig nah an der Gleichverteilung liegenden). Allerdings setzt das Extraktionsverfahren eine hinreichend große Anzahl von unabhängigen Quellen voraus, es wird gezeigt, daß eine einzige nicht ausreicht.

Die größere Allgemeinheit der Quellen erkaufen wir uns durch die folgenden Nachteile (gegenüber den in den vorangehenden Absätzen beschriebenen Verfahren):

- Es genügt nicht nur eine Quelle (wir brauchen zwar noch keinen initialen Zufall, dafür aber mehrere Quellen, was evtl. sogar störender sein kann als eine kleine Menge initialen Zufalls).
- Der erzeugte Zufall ist nicht mehr perfekt. Dieser Nachteil ist aber nicht groß, da wir jede beliebige Qualität erreichen können.

In [CG88] werden die eben vorgestellten Quellen noch verallgemeinert. Wir setzen nun für die Extraktion nur noch voraus, daß jede Bitfolge einer gewissen (festen) Länge l, gegeben alle davor liegenden Bits, nicht wahrscheinlicher ist als eine gewisse (feste) Wahrscheinlichkeit b.

In [CG88] wird dann ein Extraktionsverfahren erarbeitet, welches aus zwei Quellen des obigen Typs Zufall beliebig hoher Qualität erzeugt.

Will man die Quellen weiter verallgemeinern, so bietet es sich an, von einer Quelle X lediglich eine gewisse min-Entropie zu verlangen. Dies bedeutet, daß eine Zahl $k = H_\infty(X)$ gegeben ist, und daß jede Ausgabe der Quelle höchstens die Wahrscheinlichkeit 2^{-k} hat. Extraktion aus diesen Quellen ist für die Komplexitätstheorie wichtig, für einen Überblick über diese Problematik siehe [Nis96].

Möchte man einen Extraktor für solche Quellen konstruieren, so muß man, bevor man ein Bit ausgeben kann, die gesamte Ausgabe der Quelle verarbeiten, da ob der schwachen Voraussetzungen die gesamte der Quelle innewohnende Zufälligkeit in den hinteren Bits befindlich sein kann. Damit eignet sich diese Quellenmodellierung nur für Quellen mit endlichem Wertebereich.

Hier eine kurze Übersicht über die verschiedenen in der Literatur beschriebenen Verfahren mit ihren Parametern (weitgehend übernommen aus [NTS95, Tre99]):

Referenz	k	m	d	ε
[GW94, SZ94]	$\Omega(k)$	$(1+\Omega(1)) \cdot k$	k	$2^{-\Omega(k)}$
[Zuc97]	$\Omega(n)$	$\Omega(k)$	$O(\log n \cdot \log \varepsilon^{-1})$	beliebig
[SZ94]	$\Omega(n^{1/2+\gamma})$	n^δ, $\delta \leq \gamma$	$O(\log n)$	beliebig
[NTS95]	beliebig	k	$\operatorname{poly} \log n \cdot \log \varepsilon^{-1}$	$\geq 2^{-\sqrt{n}}$
[NTS95]	$\Omega(n^\gamma)$	$\Omega(n^\delta)$, $\delta < \gamma$	$O(\log n \cdot \log\log \ldots \log n)$	$\frac{1}{n}$
[Tre99]	$n^{\Omega(1)}$	$k^{\Omega(1)}$	$O\big(\log n - \log \varepsilon \cdot \big(1 - \frac{\log \varepsilon}{\log n}\big)\big)$	beliebig
[HILL93]	beliebig	$k - O(\log \varepsilon^{-1})$	$n + O(n)$	beliebig

Hierbei bedeuten

- die Eingabelänge n die Menge an Bits, die die Quelle liefert,
- die min-Entropie k eine untere Schranke für die min-Entropie der von der Quelle gelieferten n Bit (siehe Definition 2.4),
- die Ausgabelänge m die Menge an resultierendem Zufall,
- der initiale Zufall d die Menge an perfekt zufälligen Bits, welche in die Verarbeitung zusätzlich zu dem der Quelle entnommenen Zufall einfließen muß,
- und schließlich ε den Abstand des resultierenden Zufalls zur Gleichverteilung (wir betrachten dann $-\log \varepsilon$ als die Qualität des Zufalls).

Der letzte in dieser Tabelle angegebene Extraktor (Leftover Hash Lemma [HILL93], siehe auch Lemma 3.3 in der vorliegenden Arbeit) braucht ziemlich viel initialen Zufall, dafür findet sich dieser in unveränderter Form als Teil des Resultats wieder, er wirkt gewissermaßen als Katalysator. Diese Eigenschaft kann dann sehr von Nutzen sein, wenn man diesen Extraktor als Teilkomponente anderer Extraktoren benutzen will. Auch bei uns wird das Leftover Hash Lemma eine zentrale Rolle spielen.

1.5 Adaptive Extraktion

Vergleichen wir die im vorangegangenen Abschnitt vorgestellten Extraktoren in Hinblick auf die Struktur der unterstützten Quellen, so erkennen wir zwei Typen:

- Die Extraktoren in Abschnitt 1.4.1 unterstützen Quellen verschiedenster Qualität (sogar konstante Quellen sind zugelassen), und liefern je nach Qualität unterschiedliche Mengen an Zufall.
- In Abschnitt 1.4.2 hingegen müssen alle zugelassenen Quellen ein gewisses Mindestmaß an Zufälligkeit enthalten, denn die Länge der erzeugten Zufallsfolge ist fest.

Die Quellen des ersten Typs sind also gewissermaßen in der Lage, die Qualität der Quelle zu schätzen und sich daran anzupassen. Bei Extraktoren aus Abschnitt 1.4.1 wird dies aber damit erkauft, daß die Quellen jeweils nur wenige verborgene Parameter haben (nämlich die Transitionswahrscheinlichkeiten des Markov-Prozesses).

Bei den in Abschnitt 1.4.2 präsentierten Extraktoren hingegen ist die Familie von Quellen i. a. viel größer, da z. B. zur vollständigen Beschreibung einer *slightly random source* für jeden Zeitpunkt die Wahrscheinlichkeit für die Ausgabe einer 1 abhängig von allen bisherigen Ausgaben festgelegt sein muß, und alle diese Wahrscheinlichkeiten beliebige Werte aus $[\frac{1}{2} - \delta, \frac{1}{2} + \delta]$ annehmen können.

In der vorliegenden Arbeit versuchen wir, diese beiden Vorteile zu kombinieren, und erhalten folgendes Verfahren, welches wir die *adaptive Extraktion* taufen:

- Zerlege die zu verarbeitende Folge in Blöcke.
- Zu jedem Block erstelle eine untere Abschätzung η (die *Symbolgewichtung*), wieviel Zufall in diesem enthalten ist.
- Behandle den Block wie die Ausgabe einer Quelle mit min-Entropie η und extrahiere mit dem Leftover Hash Lemma (siehe Ende des vorangegangenen Abschnittes und Satz 3.6) $\eta - c$ Zufallsbits. Hierbei ist c eine von verschiedenen Parametern des Extraktionsverfahrens (z. B. der gewünschten Qualität) abhängige Konstante.
- Konkateniere die aus den einzelnen Blöcken extrahierten Zufallsfolgen.

Es bleibt die Frage offen, wie man die in einem Block enthaltene Zufälligkeit abschätzen kann. Angelehnt an die Definition der min-Entropie schlagen wir folgendes vor:

Es sei x der zu untersuchende Block und α die dem Block vorangehende Ausgabe (die gesamte Vergangenheit). Dann bestimmen wir für jede mögliche Quelle X die Wahrscheinlichkeit, daß die Symbolfolge x nach dem Präfix α ausgegeben wird. Der negative Logarithmus dieser Wahrscheinlichkeit ist dann der Informationsgehalt I_X des Blockes x, d. h. dessen Zufälligkeit. Da wir nicht wissen, was für eine Quelle vorliegt, verwenden wir als untere Abschätzung das Minimum von I_X über alle zugelassenen Quellen X.

Eine genaue Beschreibung und formale Analyse dieser Methode findet sich in Kapitel 4.

Die so definierte adaptive Extraktion hat nun die folgenden Eigenschaften:

- Es wird zur Laufzeit bestimmt, wieviel Zufall wir der Quelle entnehmen dürfen und die Länge der Ausgabe entsprechend angepaßt.
- Das Verfahren stellt keine Voraussetzungen an die Familie von Quellen. Allerdings sind Familien denkbar, bei denen die Symbolgewichtung verschwindet, so daß der Extraktor zwar im Prinzip korrekt arbeitet, die Ausgabe allerdings die Länge 0 hat.
- Der erzeugte Zufall ist nicht perfekt (nicht exakt gleichverteilt) aber beliebig gut (beliebig nah an der Gleichverteilung).
- Es ist ein gewisses Maß an initialem Zufall nötig (das Doppelte der Blocklänge).
- Ob das Verfahren effizient ist, hängt davon ab, ob sich die Symbolgewichtung effizient berechnen läßt. In vielen Fällen aber läßt sich zumindest eine gute untere Abschätzung durch Tabellierung der Symbolgewichtung sehr effizient berechnen (siehe Lemma 4.6 und die darauf folgende Bemerkung).
- Ist die vorliegende Quelle vollständig bekannt (d. h. hat sie keine unbekannten Parameter), so kann der Erwartungswert über die Länge der Ausgabe beliebig nah an die Entropie der Quelle gebracht werden (siehe Lemma 4.11).

Zuletzt wollen wir noch betrachten, wie sich die adaptive Extraktion verhält, wenn man die in Abschnitt 1.4 erwähnten Familien von Quellen zugrundelegt.

- Bei den in Abschnitt 1.4.1 beschriebenen Familien liegt die erwartete Länge der erzeugten Zufallsfolge beliebig nah an der Entropie der verarbeiteten Folge (für hinreichend große Blocklängen).
- Bei *slightly random sources* extrahieren wir aus jedem Block gleich viel Zufall, das Verfahren degeneriert also zu einem nicht-adaptiven Spezialfall. Die erwartete Länge des erzeugten Zufalls entspricht der kleinsten Entropie aller Quellen der betrachteten Familie.
- Bei Quellen, für die lediglich die min-Entropie bekannt ist, liefert unser Verfahren nur dann eine nicht verschwindende Menge an Zufall, wenn die Blocklänge gleich der Länge der gesamten Eingabe ist. In diesem Fall entartet die adaptive Extraktion zu einer Anwendung des Leftover Hash Lemmas ([HILL93], letzte Zeile in der Tabelle auf Seite 9).
- Besondere Vorteile hat unser Verfahren vor allem bei Quellen, die ihre Parameter verändern können, d. h. manchmal guten und manchmal schlechten Zufall liefern. Ein Beispiel für eine solche Quelle wäre eine, welche zu jedem Zeitpunkt wählen kann, ob sie gleichverteilte Bits ausgibt oder konstant 0 (siehe Abschnitt 5.2.4). Aus dieser Quelle kann keines der im vorangegangenen Abschnitt vorgestellten Verfahren etwas extrahieren; mit adaptiver Extraktion aber erhalten wir solange Daten, wie die Quelle nicht konstant 0 ausgibt.

1.6 CHMM-Quellen

Um die Modellierung von Quellen und die Berechnung der zugehörigen Symbolgewichtung zu vereinfachen, stellen wir in Kapitel 5 eine Verallgemeinerung des Konzepts der HMM (*hidden Markov models*) vor. Anders als ein HMM beschreibt ein CHMM eine Quelle nicht vollständig, sondern schränkt lediglich die Wahrscheinlichkeiten für bestimmte Transitionen ein. Damit ergibt sich für jedes CHMM eine ganze Familie von Quellen.

Wir erläutern dann weiter, wie man für eine CHMM-Quelle die zugehörige Symbolgewichtung errechnen kann. Dies ist (im Rahmen der Rechengenauigkeit) exakt möglich, somit eignen sich CHMM-Quellen besonders gut für die adaptive Extraktion.

Weiterhin geben wir einige Beispiele, was für Quellen man mit CHMM modellieren kann. Dazu gehören unter anderem die *slightly random sources* (Abschnitt 5.2.5) und die im vorangegangenen Abschnitt erwähnte Quelle, die konstant 0 ausgeben kann (Abschnitt 5.2.4).

1.7 Praktische Anwendung

Als Beispiel für eine praktische Anwendung dient eine an der LMU München entwickelte und realisierte physikalische Quelle. Wir untersuchen diese in Abschnitt 8.2.

Zunächst reißen wir Möglichkeiten zur Modellierung der Quelle als CHMM an. Danach untersuchen wir die Quelle mittels statistischer Tests (und in Abschnitt 8.1 vorgestellter Software). Hierdurch erhalten wir experimentell eine Aussage darüber, wie die Symbolgewichtung für verschiedene Einstellungen der Münchner Quelle aussieht. Mit dieser Information lassen sich dann die Extraktionsraten bestimmen.

Wir kommen zu dem Schluß, daß es – eine hinreichend schnelle Hardware für die in der Extraktion vorkommenden Faltungen vorausgesetzt – ratsamer ist, die Quelle schlechten Zufall mit hoher Ausgaberate erzeugen zu lassen, als guten, bei dem die Datenrate dann wesentlich geringer ist. Dies liegt daran, daß die durch Verringerung der Ausgaberate erreichte Verbesserung im wesentlichen einfach einem Weglassen von Bits entspricht, wohingegen die adaptive Extraktion auf die Struktur der von der Quelle gelieferten Daten eingeht, und somit bessere Ergebnisse erzielt.

Kapitel 2

Notation und mathematische Grundlagen

2.1 Notation

Im folgenden werden wir einige Konventionen mathematischer Notation etablieren, die in dieser Arbeit gelten sollen. Der Inhalt dieses Abschnitts vermittelt keine Erkenntnisse, ist aber insbesondere wichtig, wenn formale Details oder die Beweise in Anhang A verstanden werden sollen. Man beachte auch das Symbolverzeichnis auf Seite 108.

Will man sich nur einen Überblick über die in dieser Arbeit untersuchten Aussagen verschaffen, so ist eine Kenntnis dieser Konventionen nicht zwingend vonnöten.

2.1.1 Zahlen und Zahlenmengen

Es seien

$\mathbb{N}$	die Menge der natürlichen Zahlen ohne 0,
$\mathbb{N}_0$	die Menge der natürlichen Zahlen einschließlich der 0,
$\mathbb{R}$	die Menge der reellen Zahlen,
$\mathbb{R}_{>0}$	die Menge der positiven reellen Zahlen,
$\mathbb{R}_{\geq 0}$	die Menge der nichtnegativen reellen Zahlen.

Es bezeichne log durchgehend den Logarithmus zur Basis 2.

Für $x \in \mathbb{R}^M$ (wobei M eine abzählbare Menge sei) bezeichne $\|x\|_1$ die Betragssummennorm von x, d. h.

$$\|x\|_1 := \sum_{i \in M} |x_i|.$$

Weiterhin bezeichne $\mathbb{R}_1^M$ die Menge der normierten Tupel in $\mathbb{R}_{\geq 0}^M$, genauer

$$\mathbb{R}_1^M := \{x \in \mathbb{R}_{\geq 0}^M : \|x\|_1 = 1\}.$$

In $\mathbb{R}^M$ (oder einer Teilmenge davon), bezeichne e_i ($i \in M$) den Einheitsvektor mit

$$(e_i)_j = \begin{cases} 1, & i = j, \\ 0, & \text{sonst.} \end{cases}$$

Das Symbol $\mathbb{1}_n$ bezeichne die Einheitsmatrix in $\mathbb{F}^{n \times n}$, wobei $\mathbb{F}$ ein aus dem Zusammenhang ersichtlicher Körper sei.

Der Ausdruck Toeplitz$(\mathbb{F}^{m \times n})$ bezeichne die Menge der $(m \times n)$-Toeplitz-Matrizen über dem Körper $\mathbb{F}$, d. h. die Matrizen T mit

$$T_{ij} = T_{i'j'} \qquad (i - j = i' - j'),$$

also die Matrizen mit konstanten Diagonalen und Nebendiagonalen.

Es bezeichne $\perp$ ein undefiniertes Ergebnis. Wir setzen fest, daß $0 \cdot \perp = \perp \cdot 0 = 0$ und $\infty + \perp = \perp + \infty = \infty$.[1]

Ist $M \subseteq \mathbb{R} \cup \{\perp\}$, so verhalten sich Mengenoperationen wie sup, inf, max oder min auf M wie auf $M \setminus \{\perp\}$. Insbesondere ist dann $\sup\{\perp\} = \sup \emptyset = -\infty$ und $\max\{\perp\} = \max \emptyset = \perp$ und inf, min analog.

[1] Dies ermöglicht es u. a., $P(A|B)P(B) = P(AB)$ zu schreiben, ohne die Bedingung $P(B) > 0$ immer erwähnen zu müssen.

2.1.2 Operationen auf Mengen, Folgen und Funktionen

Die Menge M^* bezeichne die Menge der abbrechenden Folgen (Wörter) über $M \neq \emptyset$ und $M^{\mathbb{N}}$ die der nicht abbrechenden. In einigen Fällen bevorzugen wir, die Indizierung der Folge mit 0 beginnen zu lassen, wir schreiben dann $M^{\mathbb{N}_0}$.

Ist x eine Folge, so bezeichne $|x|$ die Länge dieser Folge, also $|x| \in \mathbb{N}_0$ für abbrechende Folgen (Wörter), und $|x| = \infty$ für nicht abbrechende.

Sind x und y zwei Folgen, so bezeichne xy die Konkatenation von x und y, d. h.

$$(xy)_i := \begin{cases} x_i, & i \leq |x|, \\ y_{i-|x|}, & i > |x|. \end{cases}$$

Der Ausdruck x, y hingegen bezeichne das Paar bestehend aus x und y, also $(x, y)_1 = x$, $(x, y)_2 = y$.

Liegt eine abbrechende Folge x vor, und ist $i > |x|$, so sei $x_i = \perp$.

Ist x eine Folge, so sei $\omega_\sigma(x)$ die Anzahl der Vorkommen von σ in x, formal

$$\omega_\sigma(x) := \sum_{\substack{i \leq |x| \\ x_i = \sigma}} 1.$$

Der Spezialfall $\omega_1(x)$, $x \in \{0, 1\}^*$ ist das Hamming-Gewicht von x.

Das Symbol λ bezeichne das leere Wort (d. h. λ ist die Folge mit $|\lambda| = 0$).

2.1.3 Ereignisse und Wahrscheinlichkeiten

Ist A irgendeine Aussage, so sei $\delta(A)$ definiert durch

$$\delta(A) := \begin{cases} 1, & A \text{ ist wahr}, \\ 0, & \text{sonst}. \end{cases}$$

Zum Beispiel ist $\delta(a^2 = b) = 1$ genau dann, wenn $a^2 = b$.

Ist X eine Zufallsvariable, die deterministisch von zwei unabhängigen Zufallsvariablen A und B abhängt (also $X = f(A, B)$), so bezeichne

$$P_{B=b}(X \in M) := P(f(A, b) \in M).$$

Analoge Interpretationen von $P_{B=b}$ gelten für andere Ereignisse und Darstellungen von X. Beispiel: Es sei $X = A + B$, und A, B unabhängig gleichverteilt auf $[0, 1]$. Dann ist

$$P_{B=\frac{2}{3}}(X \leq 1) = P(A + \tfrac{2}{3} \leq 1) = \tfrac{1}{3}.$$

Eine intuitive Vorstellung von $P_{B=b}(\ldots)$ ist $P(\ldots | B = b)$, formal ist diese aber nicht zulässig, da zumeist $P(B = b) = 0$.

Außerdem muß eine kanonische Darstellung $X = f(A, B)$ existieren, damit diese Notation definiert ist. Dies ist im Einzelfall zu verifizieren.

2.2 Quellen

Im folgenden werden wir klären, was wir formal unter einer Quelle bzw. einer Familie von Quellen verstehen.

> **Definition 2.1: Quelle**
> Es sei Σ eine endliche, nichtleere Menge. Eine *Quelle X über dem Alphabet Σ* ist eine Zufallsvariable X, welche Werte in $\Sigma^* \cup \Sigma^{\mathbb{N}}$ annimmt. □

Eine Quelle ist also ein irgendwie gearteter Zufallsprozeß, welcher eine Folge von Symbolen aus einer gegebenen Menge produziert, und welche – hier unterscheidet sich unser Begriff ein wenig von üblichen Definitionen (wie z. B. der des Begriffs *discrete source* in [Sha48]) – auch abbrechen kann (aber nicht muß).

Diese Erweiterung des Begriffs ist notwendig, da wir Extraktionsverfahren untersuchen werden, die unter bestimmten Bedingungen nicht mehr in der Lage sind, weitere Symbole zu produzieren.

Ein Beispiel für ein solches Extraktionsverfahren ist das folgende: Es sei eine Quelle X über dem Alphabet $\Sigma := \{0, 1, ?\}$ gegeben mit unabhängigen X_i und $P(X_i = 0) = P(X_i = 1)$. Dann kann man Y definieren als die Teilfolge von X bestehend nur aus den Symbolen 0 und 1. Diese resultierende Folge Y ist dann perfekt zufällig (siehe Definition 2.10), aber falls z. B. $P(X_i = 0) = P(X_i = 1) = 0$ für fast alle $i \in \mathbb{N}$, so bricht diese Folge ab.

Man beachte, daß X_i auch bei einer abbrechenden Folge X für alle $i \in \mathbb{N}$ sinnvoll ist, wir schreiben bei Indizes jenseits des definierten Bereichs $X_i = \perp$.

Da wir zumeist nicht genau angeben können, welche Verteilung die Ausgabe einer vorliegenden physikalischen Quelle hat, werden wir hauptsächlich Familien von Quellen betrachten:

Definition 2.2: Familie von Quellen
Eine *Familie $\mathcal{X}$ von Quellen* ist eine Menge von Quellen, alle über dem gleichen Alphabet $\Sigma_{\mathcal{X}}$. □

Die Einschränkung, daß alle Quellen in einer Familie das gleiche Alphabet haben, ist eine natürliche, da die Darstellung des Alphabets meist bei der Konstruktion gewählt wird und somit bekannt ist.

2.3 Entropie und Zufälligkeit

In diesem Abschnitt werden wir einige Definitionen für Maße von Zufälligkeit anführen, manche davon auf beliebige diskrete Zufallsvariablen anwendbar, manche nur auf Quellen.

Das wohl wichtigste und bekannteste Maß wurde schon in [Sha48] vorgestellt:

Definition 2.3: Entropie
Die *Entropie (oder Shannon-Entropie)* $H(X)$ einer diskreten Zufallsvariable X über einer Menge M ist definiert als

$$H(X) := -\sum_{x \in M} P(X = x) \log P(X = x).$$

Für Zufallsfolgen $X = (X_1, X_2, \ldots)$ aber ist die Entropie definiert durch

$$H(X) := \lim_{n \to \infty} \tfrac{1}{n} H(X_1 \ldots X_n),$$

falls existent. □

Darüber hinaus benötigen wir noch die folgenden zwei Varianten der Entropie:

Definition 2.4: min-Entropie
Die *min-Entropie* $H_\infty(X)$ einer diskreten Zufallsvariable X über einer abzählbaren Menge M ist definiert als

$$H_\infty(X) := -\sup_{x \in M} \log P(X = x).$$ □

Für die beiden vorstehenden Definitionen gibt es die folgende Interpretation: Wenn wir $I(x) := -\log P(X = x)$ als den Informationsgehalt des Ereignisses x ansehen, so ist die Entropie von X die zu erwartende Information $\mathrm{E}\, I(X)$ und die min-Entropie die garantierte Mindestinformation $\min I(x)$.

Definition 2.5: Renyi-Entropie
Die *Renyi-Entropie* $H_{Ren}(X)$ einer Zufallsvariable X über einer Menge M ist definiert als

$$H_{\mathrm{Ren}}(X) := -\log \sum_{x \in M} P(X = x)^2.$$ □

Man beachte, daß $2^{-H_{\mathrm{Ren}}(X)} = P(X = X')$ (wobei X' eine von X unabhängige Zufallsvariable gleicher Verteilung sei), also die Renyi-Entropie ein Maß für die Kollisionswahrscheinlichkeit zweier unabhängiger Stichproben ist.

Diese drei Maße stehen in folgendem Verhältnis zueinander:

Lemma 2.6: Abschätzungen der min-Entropie
Für jede diskrete Zufallsvariable X gilt:

$$H_\infty(X) \leq H(X), \qquad H_\infty(X) \leq H_{\mathrm{Ren}}(X). \qquad \square$$

Beweis siehe Abschnitt A.2.1, Seite 52.

Eine weiteres Maß der Zufälligkeit ergibt sich durch den Abstand einer Verteilung zur Gleichverteilung, hierzu zunächst folgende Definition:

Definition 2.7: Statistischer Abstand
Der *statistische Abstand (statistical distance)* $\mathrm{SD}(X;Y)$ *zwischen zwei diskreten Zufallsvariablen* X, Y ist definiert als

$$\mathrm{SD}(X;Y) := \tfrac{1}{2} \sum_{a \in M} \bigl|P(X=a) - P(Y=a)\bigr|,$$

wobei M die Vereinigung der Wertebereiche von X und Y sei.

Hierbei müssen X und Y nicht zwingend das gleiche Wahrscheinlicheitsmaß teilen.
Ist E ein Ereignis, so schreiben wir abkürzend $\mathrm{SD}(X;Y\|E)$ für $\mathrm{SD}(X|E;Y|E)$. $\square$

Eine nützliche Interpretation des statistischen Abstands liefert das folgende Lemma:

Lemma 2.8: Statistischer Abstand
Für diskrete Zufallsvariablen X und Y gilt

$$\mathrm{SD}(X;Y) = \max_{T \subseteq M} \bigl|P(X \in T) - P(Y \in T)\bigr|,$$

wobei M die Vereinigung der Wertebereiche von X und Y sei. $\square$

Beweis siehe Abschnitt A.2.2, Seite 52.
Dieses Lemma sagt aus, daß es keinen Test T gibt, der die beiden Verteilungen mit einer Wahrscheinlichkeit größer als $\mathrm{SD}(X;Y)$ unterscheiden kann. Diese Interpretation ist sehr wichtig für die Anwendung unserer Ergebnisse in der Kryptologie, da allgemeine Definitionen der Sicherheit von Protokollen stark vereinfacht die folgende Bedingung stellen: Egal was passiert (sprich: egal welchen Test wir verwenden), das Ergebnis weicht nicht wesentlich von dem im hypothetischen Idealfall (sprich: der Gleichverteilung) ab. Siehe hierzu auch Kapitel 6.

Wichtige Eigenschaften des statistischen Abstands zeigt das folgende Lemma:

Lemma 2.9: Eigenschaften des statistischen Abstands
Es seien X, Y, Z, U Zufallsvariablen, U unabhängig von $\{X, Y, Z\}$, und f eine Funktion, die mindestens auf den Wertebereichen von X und Y definiert ist. Dann gilt

$$\mathrm{SD}(X;Y) \geq \mathrm{SD}(f(X); f(Y)), \tag{1}$$
$$\mathrm{SD}(X;Y) = \mathrm{SD}(XU; YU), \tag{2}$$
$$\mathrm{SD}(X;Z) \leq \mathrm{SD}(X;Y) + \mathrm{SD}(Y;Z), \tag{3}$$
$$\mathrm{SD}(XZ;YZ) = \sum_{z \in M_Z} P(Z=z)\, \mathrm{SD}(X;Y\|Z=z), \tag{4}$$

wobei M_Z der Wertebereich von Z sei.
Ist f injektiv, so liegt in (1) Gleichheit vor. $\square$

Beweis siehe Abschnitt A.2.3, Seite 53.

Wir kommen nun zur Definition der Zufälligkeit.

Definition 2.10: Perfekt zufällig
Sei S eine diskrete Zufallsvariable mit Werten aus M_S. Eine Quelle X über einem Alphabet Σ heißt *perfekt*

zufällig unter Kenntnis von S, wenn für alle $n \in \mathbb{N} \cup \{\infty\}$ und $s \in M_S$ mit $P(|X| = n, S = s) > 0$ gilt: $X \big| (|X| = n, S = s)$ ist gleichverteilt auf Σ^n (mit $\Sigma^\infty := \Sigma^{\mathbb{N}}$).

Wird kein S angegeben, so setzen wir $S := \lambda$. □

Diese Definition der Zufälligkeit spiegelt wieder, daß wir durchaus zulassen wollen, daß eine Zufallsquelle aufhört, Daten zu liefern, ohne dadurch als nicht zufällig klassifiziert zu werden. Man beachte aber, daß über die Verteilung der Länge $|X|$ nichts ausgesagt wird. Will man diese festlegen, so kann man Formulierungen wie „X ist perfekt zufällig und hat Länge l“ oder „X ist perfekt zufällig und bricht nicht ab“ verwenden.

Die Zufallsvariable S stellt eine Information dar, bei deren Kenntnis die Verteilung von X immer noch perfekt zufällig erscheint. Dieser Formalismus ist besonders wichtig bei der Modellierung von Seitenkanälen.

Da die obige Definition in den meisten Fällen ein unerreichbares Ziel darstellt, müssen wir uns oft mit einer abgeschwächten Fassung begnügen:

Definition 2.11: ε-zufällig
Sei S eine diskrete Zufallsvariable. Eine Quelle X über einem Alphabet Σ heißt *ε-zufällig unter Kenntnis von S* ($\varepsilon > 0$), wenn es eine unter Kenntnis von S perfekt zufällige Quelle U mit $\mathrm{SD}(SX, SU) \leq \varepsilon$ gibt. □

Man beachte, daß wir nicht $\mathrm{SD}(X, U \parallel S = s) \leq \varepsilon$ verlangen. Dies bedeutet, daß wir es tolerieren, wenn für bestimmte $s \in M_S$ der statistische Abstand $\mathrm{SD}(X, U \parallel S = s)$ groß wird, sofern $S = s$ entsprechend unwahrscheinlich ist.

Im Zusammenhang mit diesen Definitionen werden wir später das folgende Lemma benötigen:

Lemma 2.12: Konkatenation von Zufallsquellen

Es seien $S, U_1, \ldots, U_n$ diskrete Zufallsvariablen, und U_i sei perfekt zufällig über Σ unter Kenntnis von S, U_j $(j \neq i)$.

Dann ist $U_1 \ldots U_n$ perfekt zufällig unter Kenntnis von S. □

Der Beweis hierzu findet sich in Abschnitt A.2.4, Seite 54.

Kapitel 3

Das Leftover Hash Lemma

Wir werden im Verlauf dieser Arbeit einen Mechanismus brauchen, um einen Block von Symbolen, bezüglich dessen min-Entropie wir eine nichttriviale Abschätzung kennen, in einen möglichst zufälligen umzuwandeln.[2]

Zunächst wollen wir überlegen, ob dies mittels eines deterministischen Prozesses möglich ist. Dies verneint das folgende

> **Lemma 3.1: Unmöglichkeit deterministischer Extraktion**[3]
> Sei M eine Menge, Σ ein Alphabet mit $\#\Sigma =: n$, $k \in \mathbb{R}_{\geq 0}$ und $k \leq \log \#M - \log n$. Weiter sei $\mathcal{X}$ die Menge aller Zufallsvariablen X mit Werten in M und $H_\infty(X) \geq k$, und schließlich $f : M \to \Sigma \cup \{\perp\}$.
>
> Dann existiert ein $X \in \mathcal{X}$, so daß $P(f(X) \in \{\sigma, \perp\}) = 1$ für ein $\sigma \in \Sigma$, und so daß für jede über Σ perfekt zufällige Zufallsvariable U gilt:
>
> $$\mathrm{SD}(f(X); U) \geq \frac{n-1}{n}(1 - P(U = \perp)). \qquad \square$$

Beweis siehe Abschnitt A.3.1, Seite 55.

In unserem speziellen Fall ist M die Menge der Werte, die ein Block von n Symbolen annehmen kann, und f der Extraktor.

Wir sehen, daß nur solche zufälligen Quellen hoher Qualität mit deterministischer Extraktion realisiert werden können, die fast nie Daten liefern. Dies ist aber wiederum trivialerweise möglich, z. B. ist die Quelle, die immer $\perp$ liefert, perfekt zufällig.

Wir benötigen also eine gewisse Menge an initialem Zufall, der gewissermaßen als „Katalysator“ zur Verbesserung des in unserer Quelle vorhandenen Zufalls dient. Da wir diesen initialen Zufall nicht nur für einen Block, sondern zur Bearbeitung mehrerer Blöcke verwenden wollen, brauchen wir eine Extraktionsfunktion, welche den initialen Zufall wirklich nur zur „Katalyse“ nutzt, also nicht als Teil des resultierenden Zufalls mit ausgibt. Das sogenannte Leftover Hash Lemma leistet dies. Dieses soll im folgenden vorgestellt werden, wir präsentieren dafür zunächst die folgende Definition:

> **Definition 3.2: Universelle Hashfunktion**
> Es sei
> $$h : M_R \times M_X \longrightarrow M_{\hat{X}}.$$
> Dann heißt h *universelle Hashfunktion*, wenn $\#M_X > 1$ und für alle $x, x' \in M_X$, $x \neq x'$ und $a, a' \in M_{\hat{X}}$ gilt:
> $$P\big(h(R,x) = a \ \wedge \ h(R,x') = a'\big) = (\#M_{\hat{X}})^{-2},$$
> wobei R eine auf M_R gleichverteilte Zufallsvariable sei. $\square$

Mit dieser Definition können wir das Leftover Hash Lemma in der folgenden Fassung formulieren. Diese Fassung entspricht in etwa der in [HILL93].[4]

> **Lemma 3.3: Leftover Hash Lemma, 1. Fassung**
> Es seien X, R, U Zufallsvariablen mit Werten in M_X, M_R bzw. $M_{\hat{X}}$, sowie $k \in \mathbb{R}$. Hierbei sei R gleichverteilt auf M_R, U gleichverteilt auf $M_{\hat{X}}$, $H_{\mathrm{Ren}}(X) \geq k$, sowie X, R, U stochastisch unabhängig. Weiter sei $h : M_R \times M_X \to M_{\hat{X}}$ eine universelle Hashfunktion.

[2] Kapitel 4 wird dann zeigen, wie wir solche Blöcke erhalten, und wie wir die Ergebnisse des aktuellen Kapitels (welche sich auf einen einzigen Block beziehen) auf mehrere Blöcke anwenden.

[3] Die Unmöglichkeit bezieht sich natürlich nur auf das hier angegebene Szenario, andere Voraussetzungen mögen eine Extraktion zulassen.

[4] In [HILL93] sind die Wertemengen der Zufallsvariablen auf Blöcke von Bits beschränkt, der Beweis allerdings ändert sich dadurch nicht wesentlich.

Dann ist
$$\mathrm{SD}(R, h(R,X); R, U) \leq \tfrac{1}{2}\sqrt{\#M_{\hat{X}} \cdot 2^{-k}}. \qquad \square$$

Zum Beweis siehe Abschnitt A.3.2, Seite 56.

Dieses Lemma sagt nun das folgende aus: Liegt eine Zufallsvariable X vor, über die nur der Wertebereich und eine untere Schranke für die min-Entropie bekannt ist, so kann unter Verwendung von initialem Zufall aus X guter Zufall gemacht werden. Hierbei wächst die Qualität, d. h. der negative Logarithmus des statistischen Abstands zur Gleichverteilung, proportional zur der min-Entropie von X und fällt indirekt proportional mit der Länge der Ausgabe (dem Logarithmus der Kardinalität der Ausgabe).

Wir können somit jede beliebige Qualität erreichen, indem wir die Blocklänge vor Anwendung des Leftover Hash Lemmas sehr groß wählen im Vergleich zur Blocklänge nach dieser Anwendung.[5] Genauere Abschätzungen (bei Kombination mit den in Kapitel 4 vorgestellten Mechanismen) finden sich in Satz 4.8 und Korollar 4.9.

Für sich genommen bringt diese Extraktion noch nicht viel, da die Menge an initialem Zufall zumindest bei den unten vorgestellten universellen Hashfunktionen die des resultierenden Zufalls übersteigt. Der Vorteil liegt aber in der Wiederverwendbarkeit des initialen Zufalls (siehe das nächste Kapitel), der bei Anwendung auf mehrere Blöcke ein gutes Verhältnis zwischen initialem und resultierendem Zufall liefert.

Eine leichte Verbesserung in Hinblick auf die Menge an notwendigem initialem Zufall läßt sich erreichen, indem man statt universeller Hashfunktionen die folgende etwas weniger restriktive Klasse von Funktionen verlangt:

Definition 3.4: Universelle Quasi-Hashfunktion

Es sei
$$h : M_R \times M_X \longrightarrow M_{\hat{X}}.$$
Dann heißt h *universelle Quasi-Hashfunktion*, wenn es eine Familie von Bijektionen $f_{\tilde{r}} : M_{\hat{X}} \to M'_{\hat{X}}$, $\tilde{r} \in M_{\tilde{R}}$ gibt, so daß
$$\begin{array}{rccc} \tilde{h}: & (M_R \times M_{\tilde{R}}) \times M_X & \longrightarrow & M'_{\hat{X}} \\ & (r, \tilde{r}),\ x & \longmapsto & f_{\tilde{r}}(h(r,x)) \end{array}$$
eine universelle Hashfunktion ist. $\square$

Damit ergibt sich die folgende allgemeinere Fassung, in der lediglich in den Bedingungen „universelle Hashfunktion“ durch „universelle Quasi-Hashfunktion“ ersetzt wurde:

Lemma 3.5: Leftover Hash Lemma, 2. Fassung

Es seien X, R, U Zufallsvariablen mit Werten in M_X, M_R bzw. $M_{\hat{X}}$, sowie $k \in \mathbb{R}$. Hierbei sei R gleichverteilt auf M_R, U gleichverteilt auf $M_{\hat{X}}$, $H_{\mathrm{Ren}}(X) \geq k$, sowie X, R, U stochastisch unabhängig. Weiter sei $h : M_R \times M_X \to M_{\hat{X}}$ eine universelle *Quasi*-Hashfunktion.

Dann ist
$$\mathrm{SD}(R, h(R,X); R, U) \leq \tfrac{1}{2}\sqrt{\#M_{\hat{X}} \cdot 2^{-k}}. \qquad \square$$

Der Beweis findet sich in Abschnitt A.3.3, Seite 57.

Ein weitere Verallgemeinerung des Leftover Hash Lemmas ist die Einführung von Seitenkanälen. Man stelle sich vor, daß die Quelle neben ihrer Ausgabe (die unser Gesamtsystem nur in nachbearbeiteter Form verläßt), noch einen Seitenkanal hat, über den Informationen am Extraktor vorbei nach außen gelangen. Im allgemeinen wird dies die ganze Nachbearbeitung zunichte machen, denn der Seitenkanal könnte eine Kopie der Ausgabe enthalten, und dann wäre die Entropie der nachbearbeiten Ausgabe höchstens die des initialen Zufalls (und bei Wiederverwendung desselben ist sogar die Entropie aller Blöcke zusammen gleich der des initialen Zufalls). In dem Falle aber, daß der Seitenkanal nur eine beschränkte Anzahl von Werten annehmen kann, tritt dieser Effekt nur in beschränktem Maße auf, so daß wir ihn durch geeignete Maßnahmen wie Erhöhung der min-Entropie der Eingabe oder Verkürzung der Ausgabe kompensieren können. Man kann sich in diesem Fall vereinfachend vorstellen, daß die Information, die der Seitenkanal enthält, sich einfach in einer entsprechenden Verringerung der Information der Zufallsdaten niederschlägt. Formalisiert wird dieser Sachverhalt von der endgültigen Fassung des Leftover Hash Lemmas:

[5] Vorausgesetzt, die min-Entropie steigt hinreichend schnell mit der Blocklänge. Dies ist aber meist gegeben.

Satz 3.6: Leftover Hash Lemma

Es seien X, R, U und S Zufallsvariablen mit Werten in M_X, M_R, $M_{\hat{X}}$ bzw. M_S, sowie $k \in \mathbb{R}$. Dabei seien (X, S), R und U unabhängig. Es sei U auf $M_{\hat{X}}$ und R auf M_R gleichverteilt. Schließlich seien $H_{\mathrm{Ren}}(X) \geq k$ und $h : M_R \times M_X \to M_{\hat{X}}$ eine universelle Quasi-Hashfunktion.

Dann ist

$$\mathrm{SD}(S, R, h(R, X);\ S, R, U) \leq \tfrac{1}{2} \#M_S \sqrt{\#M_{\hat{X}} \cdot 2^{-k}}. \qquad \square$$

Der Beweis hierzu findet sich in Abschnitt A.3.4, Seite 58.

3.1 Hashfunktionen

Im folgenden sollen ein paar universelle Hashfunktionen und Quasi-Hashfunktionen vorgestellt werden. Weitere Beispiele finden sich u. a. in [Sti02].

Lemma 3.7: Affine Transformationen als universelle Hashfunktion

Es sei $\mathbb{F}$ ein endlicher Körper, $M_X := \mathbb{F}^n$, $M_{\hat{X}} := \mathbb{F}^m$ mit $n \geq 1$, $m \leq n$, und $M_R := \mathbb{F}^{m \times n} \times \mathbb{F}^m \cong \mathbb{F}^{m(n+1)}$. Dann ist

$$\begin{aligned} h:\quad M_R \times M_X &\longrightarrow M_{\hat{X}} \\ (M, b),\ x &\longmapsto Mx + b \end{aligned}$$

eine universelle Hashfunktion. $\square$

Beweis siehe Abschnitt A.3.5, Seite 58.

Lemma 3.8: Affine Toeplitz-Transformationen als universelle Hashfunktion

Es sei $\mathbb{F}$ ein endlicher Körper, $M_X := \mathbb{F}^n$, $M_{\hat{X}} := \mathbb{F}^m$ mit $n \geq 1$, $m \leq n$, und $M_R := \mathrm{Toeplitz}(\mathbb{F}^{m \times n}) \times \mathbb{F}^m \cong \mathbb{F}^{2m+n-1}$. Dann ist

$$\begin{aligned} h:\quad M_R \times M_X &\longrightarrow M_{\hat{X}} \\ (M, b),\ x &\longmapsto Mx + b \end{aligned}$$

eine universelle Hashfunktion. $\square$

Beweis siehe Abschnitt A.3.6, Seite 59.

Lemma 3.9: Lineare Abbildungen als universelle Quasi-Hashfunktion

Es sei $\mathbb{F}$ ein endlicher Körper, $M_X := \mathbb{F}^n$, $M_{\hat{X}} := \mathbb{F}^m$ mit $n \geq 1$, $m \leq n$, und $M_R := \mathbb{F}^{m \times n} \cong \mathbb{F}^{mn}$. Dann ist

$$\begin{aligned} h:\quad M_R \times M_X &\longrightarrow M_{\hat{X}} \\ M,\ x &\longmapsto Mx \end{aligned}$$

eine universelle Quasi-Hashfunktion. $\square$

Beweis siehe Abschnitt A.3.7, Seite 60.

Lemma 3.10: Toeplitz-Transformationen als universelle Quasi-Hashfunktion

Es sei $\mathbb{F}$ ein endlicher Körper, $M_X := \mathbb{F}^n$, $M_{\hat{X}} := \mathbb{F}^m$ mit $n \geq 1$, $m \leq n$, und $M_R := \mathrm{Toeplitz}(\mathbb{F}^{m \times n}) \cong \mathbb{F}^{m+n-1}$. Dann ist

$$\begin{aligned} h:\quad M_R \times M_X &\longrightarrow M_{\hat{X}} \\ M,\ x &\longmapsto Mx \end{aligned}$$

eine universelle Quasi-Hashfunktion. $\square$

Beweis siehe Abschnitt A.3.7, Seite 60.

Alle diese Hashfunktionen setzen ein Alphabet der Kardinalität p^n voraus (p prim), dies jedoch stellt kein größeres Problem dar, da einfach ein größeres Alphabet als das der Modellierung der Quelle zugrundeliegende angenommen werden kann, hierbei vervielfacht sich der Logarithmus der Kardinalität um maximal 1,21,[6]

[6] Wir nehmen dabei an, daß die jeweils nächstgrößere Primpotenz gewählt wird. Die Kardinalität N können wir als $N \geq 2$ annehmen. Ist $N > 32$, so ist $2^{n-1} < N \leq 2^n$ mit $n \geq 6$, also $\log 2^n / \log N < n/(n-1) \leq 1{,}2$. Für $N = 2, \ldots, 32$ stellt man durch Ausrechnen fest, daß der Faktor maximal ist für $N = 10$, nämlich $\log 16 / \log 10 \leq 1{,}21$.

daher wird sich (bei gleicher Qualität) die erreichbare Ausgabeblocklänge (und damit die Rate) in Satz 4.8 um höchstens diesen Faktor verringern.

Ist zwingend vonnöten, daß das Ausgabealphabet mit dem Eingabealphabet übereinstimmt, so muß dann noch eine nachträgliche Extraktion durchgeführt werden. Die einfachste bestünde darin, einfach alle nicht gewünschten Symbole aus dem Ausgabestrom zu entfernen, der Verlust beträgt dann höchstens 50 %. Es sind aber natürlich auch effizientere Verfahren denkbar, die sogar den Verlust, den wir uns durch die Vergrößerung des Alphabets eingehandelt haben, beliebig gut wieder kompensieren.

An die Blocklängen (sowohl der Eingabe als auch der Ausgabe) stellen alle vier Hashfunktionen keine Anforderungen außer der offensichtlich notwendigen, daß die Ausgabe nicht länger sein darf als die Eingabe.

Die universelle Quasi-Hashfunktion in Lemma 3.10 verlangt unter den vieren am wenigsten initialen Zufall, nämlich $n + m - 1$ Symbole (wobei n und m die Eingabe- bzw. Ausgabeblocklänge seien).

Für Satz 4.8 werden wir Familien von Hashfunktionen brauchen, welche zwar den gleichen Definitionsbereich haben, aber verschiedene Wertebereiche. Genauer benötigen wir universelle Quasi-Hashfunktionen

$$h_m : M_R \times \Sigma^n \to \Sigma^m \qquad (m \leq n),$$

wobei Σ^n und M_R bei all diesen Hashfunktionen gleich sein soll. Bei den am Anfang dieses Abschnitts vorgestellten jedoch hängt M_R von m ab. Abhilfe schafft das folgende Lemma:

Lemma 3.11: Vergrößerung des initialen Zufalls einer Hashfunktion
Ist $h : M_{f(R)} \times M_X \to M_{\hat{X}}$ eine universelle Quasi-Hashfunktion, und $f : M_R \to M_{f(R)}$ eine Abbildung mit $\#f^{-1}(r) = \#f^{-1}(r')$ für alle $r, r' \in M'_R$, dann ist auch

$$\begin{array}{rccc} h_f : & M_R \times M_X & \longrightarrow & M_{\hat{X}} \\ & r,\ x & \longmapsto & h(f(r), x), \end{array}$$

eine universelle Quasi-Hashfunktion. Ist h eine universelle Hashfunktion, so ist h_f auch eine universelle Hashfunktion. □

Zum Beweis siehe Abschnitt A.3.8, Seite 61.

Damit kann eine Familie von Quellen mit den gewünschten Eigenschaften z. B. wie folgt konstruiert werden: Seien $h'_m : \mathbb{F}^{m+n-1} \times \mathbb{F}^n \to \mathbb{F}^m$ $(m \leq n)$ wie in Lemma 3.10 (Anwendung von Toeplitz-Matrizen). Dann existieren nach vorstehendem Lemma universelle Quasi-Hashfunktionen $h_m : \mathbb{F}^{2n-1} \times \mathbb{F}^n \to \mathbb{F}^m$, wobei für $f : \mathbb{F}^{2n-1} \to \mathbb{F}^{n+m-1}$ jede surjektive lineare Abbildung gewählt werden kann (z. B. einfach $(r_1, \ldots, r_{2n-1}) \mapsto (r_1, \ldots, r_{n+m-1})$).

Kapitel 4

Adaptive Extraktion

In diesem Kapitel werden wir aufzeigen, wie man auch aus Quellen, die keine garantierte min-Entropie haben, Zufall extrahieren kann. Dazu werden wir zunächst ein Qualitätsmaß für von der Quelle ausgegebene Daten definieren und dann darauf basierend ein Extraktionsverfahren vorstellen.

4.1 Symbolgewichtung

Die folgende Definition dient dazu, einer Folge von Symbolen ein gewisses Qualitätsmaß zuzuordnen, um später zu entscheiden, in welchem Maße diese in die weitere Verarbeitung mit einfließen soll.

> **Definition 4.1: Symbolgewichtung**
> Eine *Symbolgewichtung über einem Alphabet* Σ ist eine partielle Funktion
>
> $$\begin{array}{rlll} \eta: & \Sigma^* \times \Sigma^* & \longrightarrow & \mathbb{R}_{\geq 0}, \\ & \alpha, x & \longmapsto & \eta(\alpha; x). \end{array}$$
>
> Die *Symbolgewichtung* $\eta^{\mathcal{X}}$ *der Familie* $\mathcal{X}$ *von Quellen* ist definiert durch
>
> $$\eta^{\mathcal{X}}(\alpha; x) := -\log \sup_{X \in \mathcal{X}} P(X_{|\alpha|+1} \dots X_{|\alpha x|} = x \mid X_1 \dots X_{|\alpha|} = \alpha),$$
>
> wobei $\eta^{\mathcal{X}}$ eine Symbolgewichtung über $\Sigma_{\mathcal{X}}$ ist. Dabei sei $\eta^{\mathcal{X}}(\alpha; x) := \perp$, falls $P(X_1 \dots X_{|\alpha|} = \alpha) = 0$ für alle $X \in \mathcal{X}$. $\square$

Die Symbolgewichtung liefert also eine obere Schranke für die Wahrscheinlichkeit einer Symbolfolge, gegeben ihren Präfix. Da in die Definition der negative Logarithmus dieser Schranke eingegangen ist, erhalten wir für seltenere Symbolfolgen höhere Werte. Unmögliche Symbolfolgen erhalten das Gewicht ∞.

Wir wollen nun einige Ungleichungen für Symbolgewichtungen aufzeigen, die uns das Berechnen derselben erleichtern sollen:

> **Lemma 4.2: Komposition von Symbolgewichtungen**
> Es sei $\mathcal{X}$ eine Familie von Quellen und $\alpha, x_1, \dots, x_n \in \Sigma_{\mathcal{X}}^*$. Dann ist
>
> $$\eta^{\mathcal{X}}(\alpha; x_1 \dots x_n) \geq \sum_{\nu=1}^{n} \eta^{\mathcal{X}}(\alpha x_1 \dots x_{\nu-1}; x_\nu).$$
>
> Ist eine Seite dieser Ungleichung definiert (d. h. nicht $\perp$), so ist es auch die andere. $\square$

Beweis siehe Abschnitt A.4.1, Seite 61.

Dieses Lemma ermöglicht die sukzessive Berechnung einer unteren Schranke für eine Symbolgewichtung η, wenn nur z. B. die Einschränkung $\eta|_{\Sigma^* \times \Sigma}$ bekannt ist.

Leider gilt in Lemma 4.2 i. a. keine Gleichheit, wie folgendes einfache Beispiel zeigt: Es sei $\mathcal{X} := \{X, Y\}$ eine Familie von Quellen über $\{0, 1\}$, wobei X gleichverteilt auf den endlichen Folgen $\{00, 01\}$ und Y gleichverteilt auf $\{00, 11\}$ sei. Dann ist $\eta^{\mathcal{X}}(\lambda; 0) = -\log P(X_1 = 0) = 0$ und $\eta^{\mathcal{X}}(0; 0) = -\log P(Y_2 = 0 | Y_1 = 0) = 0$. Allerdings ist $P(X_1 X_2 = 00) = P(Y_1 Y_2 = 00) = \frac{1}{2}$, also $\eta^{\mathcal{X}}(\lambda; 00) = 1 > \eta^{\mathcal{X}}(0) + \eta^{\mathcal{X}}(0; 0)$.[7]

Im folgenden wollen wir spezielle Klassen von Familien von Quellen definieren, für die wir noch weitere Ungleichungen für die Symbolgewichtung angeben können.

> **Definition 4.3: Links-zeitinvariante Familien von Quellen**
> Zu einer Quelle X sei $X^{(n)}$, $n \in \mathbb{N}_0$ definiert durch $X_i^{(n)} := X_{i+n}$.

[7] Man kann auch ein CHMM (siehe Kapitel 5) konstruieren, für das in Lemma 4.2 keine Gleichheit gilt. Ein Beispiel mit $\eta^{\mathcal{C}}(\lambda; 00) = 0$, $\eta^{\mathcal{C}}(00; 0) = 0$ aber $\eta^{\mathcal{C}}(\lambda; 000) = 1$ findet sich in Abschnitt 5.2.7.

Eine Familie $\mathcal{X}$ von Quellen heißt *links-zeitinvariant*, wenn für jedes $X \in \mathcal{X}$ und jedes $n \in \mathbb{N}_0$ auch $X^{(n)} \in \mathcal{X}$ ist. □

In dieser Definition ist $X^{(n)}$ die Quelle, die aus X entsteht, wenn man die ersten n Symbole wegläßt. Somit bedeutet links-zeitinvariant, daß jede Quelle nach Weglassen eines Präfixes wieder eine Quelle aus der selben Familie ist.

Definition 4.4: Rechts-zeitinvariante Familien von Quellen
Es sei $Y^{(n)}$ analog zu $X^{(n)}$ in der vorangehenden Definition.

Eine Familie $\mathcal{X}$ von Quellen heißt *rechts-zeitinvariant*, wenn für jedes $X \in \mathcal{X}$ und jedes $n \in \mathbb{N}_0$ ein $Y \in \mathcal{X}$ existiert mit $Y^{(n)} = X$. □

Diese Definition ist gewissermaßen die Umkehrung zur Links-Zeitinvarianz. Hier wird verlangt, daß zu jeder Quelle ein Präfix (nicht notwendigerweise konstant) angegeben werden kann, so daß durch Voranstellen desselben wieder eine Quelle aus der selben Familie entsteht.

Definition 4.5: Konditioniert links-zeitinvariante Familien von Quellen
Es sei $X^{(n)}$ wie in Definition 4.3.

Eine Familie $\mathcal{X}$ von Quellen heißt *konditioniert links-zeitinvariant*, wenn für jedes $X \in \mathcal{X}$, jedes $n \in \mathbb{N}_0$ und jedes $x \in \Sigma_{\mathcal{X}}^n$ mit $P(X_1 \dots X_n = x) > 0$ auch

$$X^{(n)}|(X_1 \dots X_n = x) \in \mathcal{X}$$

gilt. □

Diese Definition ist ähnlich zu der der Links-Zeitinvarianz. Die konditionierte Links-Zeitinvarianz sagt aus, daß wenn wir eine Quelle X haben und wissen, welchen Präfix sie ausgegeben hat, die noch folgenden Daten, konditioniert nach unseren Kenntnissen, dann wieder eine Quelle aus der selben Familie bilden.

In den drei vorangegangenen Definitionen genügt es, die Bedingungen für $n = 1$ zu prüfen. In den Fällen der Links- und Rechts-Zeitinvarianz ergibt sich dies direkt aus der Definition, für den Fall der konditionierten Links-Zeitinvarianz findet sich der Beweis in Abschnitt A.4.2, Seite 62.

Die drei Definitionen sind alle unabhängig in dem Sinne, daß es für jede Teilmenge aus den drei Eigenschaften eine Familie von Quellen gibt, die genau diese Teilmenge erfüllt. Die acht Beispiele finden sich in Abschnitt A.4.3, Seite 63.

Gerüstet mit diesen neuen Klassifizierungen können wir das folgende Lemma formulieren:

Lemma 4.6: Verschiebung von Symbolgewichtungen
Es sei $\mathcal{X}$ eine Familie von Quellen, $\alpha_1, \alpha_2, x \in \Sigma_{\mathcal{X}}^*$ und $n \in \mathbb{N}_0$.

Ist $\mathcal{X}$ konditioniert links-zeitinvariant, so gilt, falls $\eta^{\mathcal{X}}(\alpha_1\alpha_2; x) \neq \perp$:

$$\eta^{\mathcal{X}}(\alpha_2; x) \leq \eta^{\mathcal{X}}(\alpha_1\alpha_2; x). \tag{5}$$

Ist $\mathcal{X}$ rechts-zeitinvariant, so gilt für $\eta^{\mathcal{X}}(\alpha_2; x) \neq \perp$:

$$\eta^{\mathcal{X}}(\alpha_2; x) \geq \min_{\alpha \in \Sigma_{\mathcal{X}}^n} \eta^{\mathcal{X}}(\alpha\alpha_2; x). \tag{6}$$

Ist $\mathcal{X}$ rechts-zeitinvariant und konditioniert links-zeitinvariant, so gilt in (6) sogar Gleichheit. □

Beweis siehe Abschnitt A.4.4, Seite 64.

Dieses Lemma ermöglicht es uns, bei konditioniert links-zeitinvarianten Quellen nur eine beschränkte Anzahl von vorangegangenen Symbolen in die Berechnung der Gewichtung einfließen zu lassen (5). Bei zusätzlich rechts-zeitinvarianten Quellen sagt (6) aus, daß diese Abschätzung optimal ist.

Dank der Lemmata aus diesem Abschnitt können wir folgende Methode verwenden, um die Berechnung der Symbolgewichtungen großer Datenmengen aus links-zeitinvarianten Quellen zu beschleunigen. In der Vorverarbeitung wird $\eta^{\mathcal{X}}$ für Argumente aus $\Sigma_{\mathcal{X}}^n \times \bigcup_{i=1}^m \Sigma_{\mathcal{X}}^i$ tabelliert ($n, m \in \mathbb{N}$), dann kann mit Hilfe der Lemmata 4.2 und 4.6 aus dieser Tabelle die Symbolgewichtung $\eta^{\mathcal{X}}(\alpha; x)$ mit $|\alpha| \geq n$ nach unten abgeschätzt werden durch

$$\sum_{\nu=1}^{n} \eta^{\mathcal{X}}(\sigma_n(\alpha x_1 \dots x_{\nu-1}); x_\nu),$$

wobei $x_1 \dots x_n$ eine beliebige Zerlegung von x mit $|x_\nu| \leq m$ sei und $\sigma_n(\omega)$ die letzten n Symbole von ω bezeichne.

4.2 Extraktion

Wir haben im vorangegangenen Abschnitt ein Maß aufgestellt, welches angibt, wie zufällig eine gegebene Ausgabe der Quelle ist. Es bietet sich nun an, dies mit den Ergebnissen aus Kapitel 3 zu kombinieren. Hierzu zerlegen wir die Ausgabe der Quelle in Blöcke, gewichten die Blöcke und wenden dann das Leftover Hash Lemma (Satz 3.6) auf jeden Block an. Hierbei setzen wir die Ausgabeblocklänge abhängig von der Gewichtung des jeweiligen Block, denn das Leftover Hash Lemma sagt uns, daß wir bei Blöcken „größerer Zufälligkeit" einen größeren Ausgabeblock erhalten. Dieses Extraktionsverfahren formalisieren wir wie folgt:

Definition 4.7: Adaptiver Hash-Extraktor $\Xi_{\eta,h}^{n,m}$

Es sei η eine Symbolgewichtung, $n \in \mathbb{N}$, $m : \mathbb{R}_{\geq 0} \cup \{\infty\} \to \mathbb{N}_0$, weiter M_R, Σ und Σ_{out} endliche, nichtleere Mengen, und h eine Familie von Funktionen

$$h_{\tilde{m}} : M_R \times \Sigma^n \to \Sigma_{\text{out}}^{\tilde{m}} \qquad (\tilde{m} \in \mathcal{M} := m(\mathbb{R}_{\geq 0} \cup \{\infty\}) \setminus \{0\}).$$

Dann ist der *adaptive Hash-Extraktor*

$$\Xi_{\eta,h}^{n,m} : \Sigma^* \cup \Sigma^{\mathbb{N}} \longrightarrow \Sigma_{\text{out}}^* \cup \Sigma_{\text{out}}^{\mathbb{N}}$$

durch folgende Konstruktion definiert:

Sei $X \in \Sigma^* \cup \Sigma^{\mathbb{N}}$ und $R \in M_R$, sowie

$$B_i := \begin{cases} X_{(i-1)n+1} \dots X_{in}, & \text{falls } |X| \geq in, \\ \perp, & \text{sonst,} \end{cases}$$

$$\hat{X}_i := \begin{cases} h_{m(\eta(B_1 \dots B_{i-1};\, B_i))}(R, B_i), & \text{falls } B_i \neq \perp,\ \eta(B_1 \dots B_{i-1};\, B_i) \neq \perp \\ & \text{und } m(\eta(B_1 \dots B_{i-1};\, B_i)) > 0, \\ \lambda, & \text{sonst,} \end{cases}$$

und schließlich

$$\Xi_{\eta,h}^{n,m}(R, X) := \hat{X}_1 \hat{X}_2 \dots$$ □

Die verschiedenen Komponenten dieser Konstruktion können wir folgt interpretiert werden:

Die Zufallsvariable B_i gibt den i-ten Quelldatenblock der Länge n an. Dieser wird gewichtet, und dann wird eine universelle Quasi-Hashfunktion (gemäß Leftover Hash Lemma, mit initialem Zufall R) auf B_i angewandt, wobei die Ausgabelänge mittels der Abbildung m aus der Gewichtung errechnet wird; das Ergebnis findet sich dann in der Variablen $\hat{X}_i$. Das Endresultat (die extrahierte Zufallsfolge) ergibt sich schließlich durch Konkatenation der einzelnen Ergebnisblöcke.

Um oben beschriebenes Extraktionsverfahren einsetzen zu können, müssen wir wissen, von welcher Qualität der resultierende Zufall ist. Dies klärt der folgende Satz:

Satz 4.8: Adaptive Extraktion

Es sei $\mathcal{X}$ eine Familie von Quellen über Σ, $\eta \leq \eta^{\mathcal{X}}$ eine Symbolgewichtung über Σ, $l \in \mathbb{N}$, $n \in \mathbb{N}$, $m : \mathbb{R}_{\geq 0} \cup \{\infty\} \to \{0, \dots, n\}$, weiter M_R, M_S, Σ_{out} endliche, nichtleere Mengen und h eine Familie von universellen Quasi-Hashfunktionen

$$h_{\tilde{m}} : M_R \times \Sigma^n \to \Sigma_{\text{out}}^{\tilde{m}} \qquad (\tilde{m} \in \mathcal{M} := m(\mathbb{R}_{\geq 0} \cup \{\infty\}) \setminus \{0\}).$$

Außerdem sei R eine auf M_R gleichverteilte und von X, S unabhängige Zufallsvariable, S eine Zufallsvariable mit Werten in M_S und $X \in \mathcal{X}$.

Seien ferner

$$\hat{X} := \Xi^{n,m}_{\eta,h}(R, X_1 \dots X_l)$$

und

$$\log \varepsilon := \tfrac{1}{2} \sup_{\substack{k \in \mathbb{R}_{\geq 0} \\ m(k) \neq 0}} (m(k) \log \#\Sigma_{\text{out}} - k) + \log(l+1) + \log\lfloor l/n \rfloor + \log \#M_S + \tfrac{1}{2} \log \#\mathcal{M} - 1 \tag{7}$$

Dann ist $P(|\hat{X}| \leq l) = 1$, und $\hat{X}$ ist ε-zufällig unter Kenntnis von $R, S, |\hat{X}|$. □

Beweis siehe Abschnitt A.4.5, Seite 65.

Gegenüber Definition 4.7 ist noch der Parameter l hinzugekommen, welcher angibt, wieviele Symbole maximal der Ursprungsquelle entnommen werden.

Zu beachten ist hier, daß selbst bei Kenntnis des initialen Zufalls R die extrahierte Folge noch obiger Qualitätsabschätzung genügt.

Außerdem erlaubt dieser Satz auch noch, einen beliebigen Seitenkanal S anzunehmen, vorausgesetzt dieser nimmt nur ein endliches Repertoire von Werten an,[8] dann sinkt die Qualität nur logarithmisch in der Anzahl der möglichen Werte.

Intuitiv läßt sich obiger Satz etwa wie folgt begründen: Die Symbolgewichtung $\eta := \eta(B_1 \dots B_{i-1}; B_i)$ eines Blocks läßt eine Abschätzung der min-Entropie dieses Blocks zu. Von dieser min-Entropie ist das Wissen, daß uns der Seitenkanal S vermittelt, (höchstens $\log \#M_S$) abzuziehen. Wir kommen auf eine min-Entropie von $O(\eta - \log \#M_S)$. Da auch die Länge der Ausgabefolge einen Seitenkanal darstellt, welcher $l+1$ Werte annehmen kann (min. Länge 0, max. Länge l), verringert sich die min-Entropie noch weiter um $\log(l+1)$. Durch Anwendung des Leftover Hash Lemmas erhalten wir dann eine Qualität (d. h. $-\log \varepsilon$) proportional zur min-Entropie abzüglich der Ausgabeblocklänge (in Bit, daher müssen wir noch mit $\log \#\Sigma_{\text{out}}$ multiplizieren), also $O(-m \log \#\Sigma_{\text{out}} + k - \log(l+1) - \log \#M_S)$, wobei k die Symbolgewichtung darstellt. Da wir vom schlimmsten Fall ausgehen müssen, ist hier das Infimum über alle Symbolgewichtungen zu nehmen. Da wir maximal $\lfloor l/n \rfloor$ Blöcke aus der Ursprungquelle erhalten, ist der statistische Abstand ε über ebensoviele Blöcke zu summieren, also die Qualität noch um $\log\lfloor l/n \rfloor$ zu senken. Damit sind alle wesentlichen Terme aus (7) erklärt (denn $\#\mathcal{M} \leq n$ ist klein im Vergleich zu den anderen Größen).

Da diese Abschätzung der Qualität ob der vielen Faktoren etwas unübersichtlich ist, soll das folgende Korollar helfen:

Korollar 4.9: Adaptive Extraktion

Es sei $\mathcal{X}$ eine Familie von Quellen über Σ, $\eta \leq \eta^{\mathcal{X}}$ eine Symbolgewichtung über Σ, $l \in \mathbb{N}$, $n \in \mathbb{N}$, $\varepsilon > 0$, weiter M_R, M_S endliche, nichtleere Mengen und h eine Familie von universellen Quasi-Hashfunktionen $h_{\tilde{m}} : M_R \times \Sigma^n \to \{0,1\}^{\tilde{m}}$ ($\tilde{m} = 1, \dots, n$). Weiter sei R eine auf M_R gleichverteilte Zufallsvariable, S eine Zufallsvariable mit Werten in M_S und $X \in \mathcal{X}$.

Wir setzen

$$\begin{aligned} c &:= -2\log\varepsilon + 4\log l - \log n + 2\log \#M_S, \\ m(k) &:= \begin{cases} 0, & (k - c \leq 0), \\ \lfloor k - c \rfloor, & (0 \leq k - c \leq n), \\ n, & (k - c \geq n), \end{cases} \\ \hat{X} &:= \Xi^{n,m}_{\eta,h}(R, X_1 \dots X_l), \end{aligned}$$

dann ist $\hat{X}$ ε-zufällig unter Kenntnis von $R, S, |\hat{X}|$. □

Der Beweis findet sich in Abschnitt A.4.6, Seite 68.

[8] Achtung: Hiermit ist nicht eine Quelle mit endlichem Alphabet gemeint, sondern eine Zufallsvariable, die nur endlich viele Werte annimmt.

Gewappnet mit diesem Korollar kann man nun wie folgt vorgehen, um aus einer gegebenen Quelle Zufall zu extrahieren.

- Zunächst modelliert man eine Familie von Quellen, von der man postuliert, daß die vorliegende Quelle dazugehört. Siehe auch Kapitel 5 für konkretere Modellierungsmethoden.
- Dann bestimmt man eine untere Abschätzung für die Symbolgewichtung dieser Familie.
- Als nächstes schätzt man die Lebensdauer der Quelle großzügig ab. Liegt z. B. eine binäre Quelle vor, die 20 MBit/s an Daten liefert, und nehmen wir an, daß diese maximal 1000 Jahre in Betrieb sein wird, so können wir folgern, daß die Quelle nicht mehr als $l := 2^{60}$ Symbole ausgeben wird.
- Dann wählen wir eine Blocklänge n. Größere Blocklängen liefern laut Korollar eine höhere Extraktionsrate, allerdings nimmt i. a. der Aufwand für die Berechnung der Hashfunktionen zu. Es gilt also abzuwägen. Um dies zu erleichtern, sei folgende Heuristik vorgeschlagen: Zunächst schätzt man experimentell den Erwartungswert R von $\eta(B_1 \dots B_{i-1}; B_i)/n$ (durch generieren von Testdaten und Berechnung von η). Bei realen Quellen wird dieser Erwartungswert für hinreichend große n üblicherweise nicht stark von i oder n abhängen. Dann weiß man, daß die Extraktionsrate laut obigem Korollar ungefähr

  $$\frac{\mathrm{E}\, m(\eta(B_1 \dots B_{i-1}; B_i))}{n} \approx \frac{nR - c}{n} \overset{n \ll l}{\approx} \frac{nR - \overbrace{(-2\log\varepsilon + 4\log l + 2\log \#M_S)}^{=:\tilde{c}}}{n}$$

  sein wird. Dies konvergiert für $n \to \infty$ von unten gegen R, man wähle nun n so groß, daß die Rate möglichst nahe an R kommt, ohne dabei n allzu groß werden zu lassen. (Mit $n := 10\tilde{c}/R$ erreicht man beispielsweise bereits 90 % der maximalen Rate R.)

  Die Tatsache, daß die Wahl von n nur von heuristischen Überlegungen abhängt, ist nicht weiter kritisch, da dies nur einen Einfluß auf die Effizienz der Extraktion hat, nicht aber auf die Qualität des resultierenden Zufalls.
- Zuletzt wählen wir eine Familie von Hashfunktionen. Liegt eine binäre Quelle vor, bieten sich wegen der geringen Menge an benötigtem initialen Zufall die in Lemma 3.10 untersuchten Toeplitz-Transformationen an. Mit Lemma 3.11 kreieren wir eine Familie von universellen Quasi-Hashfunktionen mit $2n - 1$ Bit initialem Zufall.
- Wir extrahieren $\hat{X}$ gemäß Korollar 4.9.

In vorstehenden Überlegungen haben wir gesehen, daß die folgende Kenngröße von Interesse ist:

Definition 4.10: Rate

Sei η eine Symbolgewichtung über einem Alphabet Σ. Die *Rate $R(X,\eta)$ von $X \in \mathcal{X}$ mit η* ist definiert durch

$$R(X,\eta) := \lim_{n\to\infty} \lim_{l\to\infty} \tfrac{1}{l} \sum_{i=1}^{\lfloor l/n \rfloor} \mathrm{E}\,\eta(X_1 \dots X_{(i-1)n};\ X_{(i-1)n+1} \dots X_{in}).$$

Die *Rate $R(\mathcal{X},\eta)$ von $\mathcal{X}$ mit η* ist dann

$$R(\mathcal{X},\eta) := \inf_{X\in\mathcal{X}} R(X,\eta),$$

die *Rate $R(\mathcal{X})$ von $\mathcal{X}$*

$$R(\mathcal{X}) := R(\mathcal{X},\eta^{\mathcal{X}}),$$

und die *Rate $R(X,\mathcal{X})$ von X in $\mathcal{X}$*

$$R(X,\mathcal{X}) := R(X,\eta^{\mathcal{X}}). \qquad \square$$

Man beachte, daß die Rate nur ein erster Richtwert ist, wie gut sich Zufall aus einer Familie von Quellen extrahieren läßt, es könnte z. B. die Rate nur für $n \approx l$ erreicht werden, was bedeuten würde, daß mehr initialer Zufall benötigt wird, als letztlich extrahiert wird.

Für den Spezialfall einelementiger Familien von Quellen können wir die Rate direkt angeben:

Lemma 4.11: Rate einelementiger Quellen
Sei X eine Quelle über Σ und $\mathcal{X} := \{X\}$. Dann ist

$$\eta^{\mathcal{X}}(\alpha;\, x) = \sum_{\nu=1}^{|x|} \eta^{\mathcal{X}}(\alpha x_1 \dots x_{\nu-1};\, x_\nu) \qquad (\alpha, x \in \Sigma^*) \tag{8}$$

und, falls $H(X)$ existiert,

$$R(\mathcal{X}) = R(X, \mathcal{X}) = H(X). \qquad \square$$

Zum Beweis siehe Abschnitt A.4.7, Seite 69.

Nebenbei erkennen wir hieran auch noch

$$R(X, \mathcal{X}) \le R(X, \{X\}) \overset{4.11}{=} H(X)$$

für beliebige Familien $\mathcal{X}$ von Quellen.

4.3 Beispiele

Im folgenden wollen wir einige Beispiele für Symbolgewichtungen verschiedener Familien von Quellen vorstellen.

4.3.1 Quelle mit festem Bias

Es seien X_i unabhängig identisch verteilt auf $\Sigma := \{0, 1\}$ mit $P(X_i = 1) = \gamma$, $\gamma \in [0, 1]$. Es sei dann $\mathcal{X} := \{X\}$. Es ist

$$P(X_i \dots X_j = x | X_1 \dots X_{i-1} = \alpha) = \gamma^{\omega_1(x)}(1-\gamma)^{\omega_0(x)} \qquad (1 \le i \le j,\ \alpha \in \Sigma^{i-1},\ x \in \Sigma^{j-i+1}),$$

also folgt direkt aus Definition 4.1:

$$\eta^{\mathcal{X}}(\alpha; x) = -\omega_1(x) \log \gamma - \omega_0(x) \log(1-\gamma) \qquad (\alpha, x \in \Sigma^*).$$

Daraus ergibt sich direkt die Rate

$$R(\mathcal{X}) = R(X, \mathcal{X}) = -\gamma \log \gamma - (1-\gamma) \log(1-\gamma) = H(X),$$

in Übereinstimmung mit Lemma 4.11.

4.3.2 Quelle abschnittsweise garantierter min-Entropie

Es sei Σ nichtleer und endlich, $k \in \mathbb{R}_{\ge 0}$, $n \in \mathbb{N}$, $k \le n \log \#\Sigma$. Dann sei $\mathcal{X}$ die Familie aller Quellen X mit

$$H_\infty(X_i \dots X_{i+n-1} | X_1 \dots X_{i-1} = \alpha) \ge k \qquad (i \in \mathbb{N}, \alpha \in \Sigma^i).$$

Dann ist nach der Definition der min-Entropie (Definition 2.4) und der Symbolgewichtung (Definition 4.1) sofort

$$\eta^{\mathcal{X}}(\alpha; x) \ge k \qquad (x \in \Sigma^n),$$

und mit Lemma 4.2 ergibt sich schließlich

$$\eta^{\mathcal{X}}(\alpha; x) \ge k \left\lfloor \frac{|x|}{n} \right\rfloor \qquad (x \in \Sigma^*).$$

Direkt aus Definition 4.10 ergibt sich dann für jedes $X \in \mathcal{X}$

$$R(X, \mathcal{X}) = \frac{k}{n},$$

welches die minimale min-Entropie pro Symbol ist.

4.3.3 Von-Neumann-Quelle

In [vN51] wurde die Familie von Quellen eingeführt, die aus allen unabhängig identisch verteilten binären Quellen besteht. Wir wollen diese Familie ein wenig verallgemeinern und mit unseren Mittel untersuchen.

Sei Σ endlich und nichtleer, sowie $\mathcal{X}$ die Familie aller unabhängig identisch verteilten Quellen über Σ. Für jede Quelle $X \in \mathcal{X}$ gilt dann:

$$P(X_i \dots X_j = x \mid X_1 \dots X_{i-1} = \alpha) = \prod_{\sigma \in \Sigma} P(X_1 = \sigma)^{\omega_\sigma(x)} \qquad (1 \le i \le j, \alpha \in \Sigma^{i-1}, x \in \Sigma^{j-i+1}),$$

damit ist

$$\eta^{\mathcal{X}}(\alpha; x) = -\sup_{p \in \mathbb{R}_1^\Sigma} \sum_{\sigma \in \Sigma} \omega_\sigma(x) \log p_\sigma \overset{(*)}{=} -|x| \sum_{\sigma \in \Sigma} \frac{\omega_\sigma(x)}{|x|} \log \frac{\omega_\sigma(x)}{|x|} \qquad (\alpha, x \in \Sigma^* \setminus \{\lambda\}),$$

die Gleichheit $(*)$ wird in Abschnitt A.4.8, Seite 70 gezeigt.

Da $\frac{1}{|x|}\eta^{\mathcal{X}}(\alpha; x)$ für unabhängig identisch verteilte Zufallsfolgen ein asymptotisch erwartungstreuer Schätzer für die Entropie ist (siehe Abschnitt A.4.9, Seite 71), gilt

$$R(X, \mathcal{X}) = H(X),$$

aber

$$R(\mathcal{X}) = 0.$$

Man beachte, daß man, wenn man $\eta^{\mathcal{X}}$ symbolweise berechnet und mit Lemma 4.2 als

$$\eta(\alpha;\, x_1 \dots x_i) := \sum_{\nu=1}^{i} \eta^{\mathcal{X}}(\alpha x_1 \dots x_{\nu-1};\, x_\nu)$$

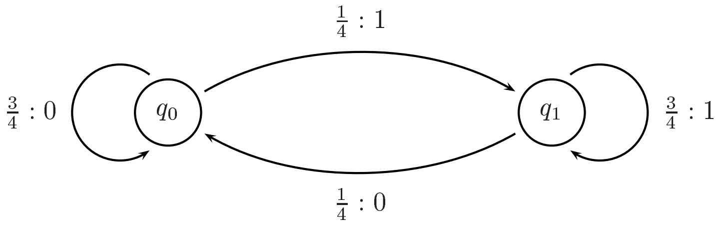
zusammensetzt, $\eta = 0$ erhält.

Kapitel 5

CHMM-Quellen

5.1 Modellierung

In diesem Kapitel wollen wir eine spezielle Klasse von Familien von Quellen untersuchen, die CHMM-Quellen. Hierzu rufen wir uns zunächst die Funktionsweise eines HMM (*hidden Markov model*) ins Gedächtnis (eine kurze Einführung in die Theorie der HMM findet sich z. B. in [Rab90]). Man betrachte das folgende Beispiel:[9]



Hier ist ein HMM mit zwei Zuständen q_0, q_1 dargestellt. Aus jedem dieser Zustände geht das HMM mit Wahrscheinlichkeit $\frac{1}{4}$ in den anderen Zustand über, ansonsten verbleibt es im vorhergehenden Zustand. Bei jedem Übergang zu q_0 gibt es 0 aus, bei jedem zu q_1 wird 1 emittiert.[10]

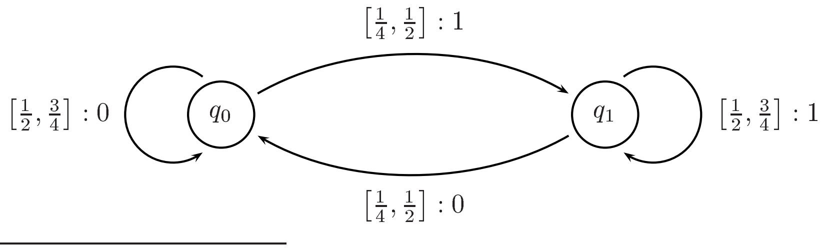
Durch dieses HMM wird nun eine Quelle beschrieben, bei welcher nur mit einer Wahrscheinlichkeit von $\frac{1}{4}$ die Ausgabe wechselt, welche also eine starke Tendenz zu gleichbleibenden Ausgabesequenzen hat.

Wenngleich man mit HMM viele Quellen beschreiben kann, haben sie für unsere Anwendung zwei schwerwiegende Mängel:

- Ein HMM beschreibt nur eine einzige Quelle,[11] wollen wir eine Familie von Quellen beschreiben, müssen wir eine Familie von HMM angeben. Auch dann ist die Quelle von ihrem ersten Symbol an auf gewisse Übergangswahrscheinlichkeiten festgelegt. Es ist dann nicht möglich, daß eine Quelle mit einer Übergangsverteilung beginnt und später eine andere annimmt.

- Chaotische Prozesse lassen sich – wenn überhaupt – nur mit gigantischen HMM modellieren. Liegt z. B. eine Quelle vor, die anhand eines chaotischen oder zumindest komplexen Prozesses aus der bisherigen Ausgabe entscheidet, ob eine Übergangswahrscheinlichkeit p_1 oder p_2 vorliegt, dann kann die Quelle nicht mehr modelliert werden, wenn sich dieser Entscheidungsprozess einer Modellierung widersetzt, obwohl wir über die Quelle immerhin eine sehr klare Aussage treffen können: Die besagte Übergangswahrscheinlichkeit wird immer p_1 oder p_2 sein.

Um dem abzuhelfen, führen wir kontrollierte HMM ein, kurz CHMM. Diese erlauben es, eine (bzgl. der Rechenkapazität) beliebig mächtige Instanz (den Quellen-Adversary, kurz Adversary) anzunehmen, welche vor jeder Ausgabe die Übergangsverteilungen neu wählen darf. Damit aber nicht einfach die Familie aller Quellen dabei herauskommt, ist die Wahl der Verteilungen für jeden Zustand auf eine gewisse Teilmenge aller möglichen beschränkt. Obiges HMM können wir z. B. wie folgt zu einem CHMM erweitern:[12]



[9] Dateiname `hmmexample.chmm` (siehe Abschnitt B.4).

[10] Man beachte, daß die im obigen Graphen verwandte Notation es auch zuläßt, beim Übergang zu einem Zustand abhängig davon, über welchen Pfeil die Transition ging, die Verteilung über die möglichen Ausgaben zu variieren. Diese Möglichkeit ist bei HMM nicht gegeben. Wir haben uns aber hier für diese Notation entschieden, weil damit die HMM klar als Spezialfall der weiter unten eingeführten CHMM erkennbar sind.

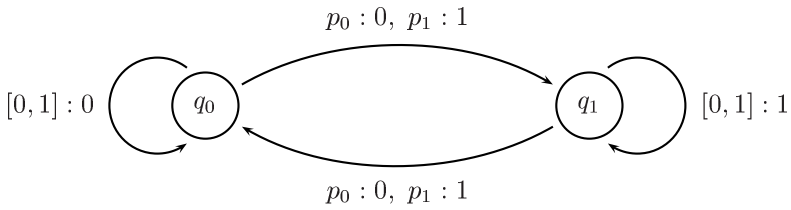
[11] Genaugenommen schon mehrere, wenn wir keinen Startzustand festlegen, aber meist gehen unsere Ansprüche an die Größe der Familie darüber hinaus.

[12] Dateiname `chmmexample.chmm` (siehe Abschnitt B.4).

Dieses CHMM kann in jedem Schritt eine Wahrscheinlichkeit für das Verweilen im aktuellen Zustand aus dem Interval $[\frac{1}{2}, \frac{3}{4}]$ frei wählen.[13] Wählt der Adversary $\frac{3}{4}$, so entspricht das CHMM dem HMM oben. Wählt er $\frac{1}{2}$, so verhält sich die Quelle wie eine perfekt zufällige. Jede Abstufung dazwischen kann angenommen werden, das Verhalten kann nach jedem Ausgabesymbol neu bestimmt werden.

Diagramme der obigen Form sind nun wie folgt zu lesen: Ist das CHMM in Zustand q, so betrachte man die von q abgehenden Pfeile. Jeder Pfeil ist mit einer Wahrscheinlichkeitsmenge und einem Ausgabesymbol versehen. (Ist ein Pfeil mit mehreren solchen Paaren versehen, so ist er als mehrere Pfeile mit gleichem Ursprung und gleichem Ziel zu verstehen.) Vor der Bestimmung der nächsten Ausgabe kann der Adversary eine Wahrscheinlichkeitsverteilung auf den betrachteten Pfeilen festlegen, mit der Einschränkung, daß jeder Pfeil eine Wahrscheinlichkeit zugeteilt bekommt, die innerhalb der ihm zugeordneten Menge liegt. Dann wird entsprechend der Verteilung ein Pfeil gewählt, und ihm entsprechend eine Ausgabe und ein neuer Zustand.

Man beachte, daß diese Darstellungsform noch nicht volle Allgemeinheit hat. Nehmen wir z. B. ein CHMM, welches folgendes Verhalten zeigt: Aus einem Zustand kann es mit einer frei wählbaren Wahrscheinlichkeit in den anderen Zustand wechseln. Tut es dies, ist die Ausgabe mit doppelt so großer Wahrscheinlichkeit 0 wie 1. Verbleibt es im aktuellen Zustand, so ist die Ausgabe 0 für q_0 und 1 für q_1. Dieses CHMM läßt sich nicht nach unserem Schema darstellen, man bedient sich dann folgender Syntax: Anstelle von Mengen können an manche Pfeile auch Variablen geschrieben werden. Dann annotiert man das CHMM mit beliebig gearteten Bedingungen (z. B. Gleichungen) in diesen Variablen, die vom Adversary wählbaren Verteilungen müssen dann diese Bedingungen erfüllen. Eben beschriebenes Beispiel sieht dann wie folgt aus:[14]



$$p_0 = 2p_1$$

Man sollte noch beachten, daß nicht jedes Diagramm ein CHMM darstellt, denn wenn für einen Zustand keine vom Adversary wählbare Verteilung existiert (z. B. weil die möglichen Wahrscheinlichkeiten sich in keinem Fall zu 1 addieren, oder weil die angefügten Gleichungen keine Lösung haben), dann liegt kein CHMM vor.

Formal besteht ein CHMM natürlich nicht aus Pfeilen und Annotationen, sondern genügt einfach der folgenden Definition:

> **Definition 5.1: CHMM**
> Ein *CHMM $\mathcal{C}$ (kontrolliertes HMM, controlled hidden Markov model)* besteht aus einem Alphabet $\Sigma_\mathcal{C}$, einer endlichen Menge $Q_\mathcal{C}$ von Zuständen und einer Familie von *Transitionsbereichen* $\mathcal{C}_q$ $(q \in Q_\mathcal{C})$ mit
> $$\mathcal{C}_q \subseteq \mathbb{R}_1^{\Sigma_\mathcal{C} \times Q_\mathcal{C}}, \qquad \mathcal{C}_q \neq \emptyset. \qquad \square$$

Diese Definition entspricht wie folgt den oben erläuterten Objekten:

- Die Menge der Zustände $Q_\mathcal{C}$ wurde dargestellt durch Kreise mit den Namen der Zustände.
- Weiter ist $\Sigma_\mathcal{C}$ die Menge der Symbole, welche bei einem Übergang ausgegeben werden können. Sie standen an den Pfeilen nach dem Doppelpunkt.
- Für einen gegebenen Zustand q ist $\Sigma_\mathcal{C} \times Q_\mathcal{C}$ also die Menge der von q ausgehenden Pfeile, da ein solcher aus einem Ursprung (q), einem Ziel $(\in Q_\mathcal{C})$ und einer Ausgabe $(\in \Sigma_\mathcal{C})$ besteht.[15] (Im Diagramm sind es meist weniger Pfeile, da die mit verschwindender Wahrscheinlichkeit i. a. nicht dargestellt werden.)
- Für die vom Zustand q ausgehenden Pfeile ist $\mathcal{C}_q$ die Menge der vom Adversary wählbaren Verteilungen. Hierbei ist eine Verteilung p dargestellt als Tupel in $\mathbb{R}_1^{\Sigma_\mathcal{C} \times Q_\mathcal{C}}$, es stellt dann $p_{x,q'}$ die Wahrscheinlichkeit für einen Übergang nach q' mit Ausgabe x dar.

[13] Die Wahrscheinlichkeit für den Übergang in den anderen Zustand darf aus $[\frac{1}{4}, \frac{1}{2}]$ gewählt werden, da aber die Gesamtwahrscheinlichkeit 1 betragen muß, ist diese Übergangswahrscheinlichkeit gerade 1 abzüglich der Verweilwahrscheinlichkeit.

[14] Dateiname: `notinterval.chmm` (siehe Abschnitt B.4).

[15] Die Wahrscheinlichkeitsmenge, die im Diagramm am Pfeil notiert ist, ist strenggenommen nicht Teil des Pfeils, da sich die Verteilungen immer auf mehrere Pfeile beziehen.

Wir haben oben von einer Instanz gesprochen, die die Auswahl der Übergangsverteilung vornimmt, dem Adversary. Um diesen Adversary A möglichst allgemein zu gestalten, modellieren wir ihn als Funktion A, welche die folgenden Eingaben bekommt:

- Den aktuellen Zeitpunkt, d. h. eine laufende Nummer i beginnend bei 1 für das erste Symbol.
- Ein unbeschränktes Zufallsband (r_ν) von *reellen* Zahlen im Bereich 0 bis 1. Dieses Zufallsband wird bei sukzessiven Aufrufen des Adversaries nicht neu initialisiert.
- Den vorangegangenen Zustand q. Der Adversary darf nur Verteilungen aus $\mathcal{C}_q$ wählen.
- Alle vorangegangenen Zustände $(q_0, \ldots, q_i)$.
- Alle bislang ausgegebenen Symbole $(x_1, \ldots, x_i)$.

Man beachte, daß die Zustände bei Index 0 beginnen, die ausgegebenen Symbole hingegen bei 1. Dies liegt darin begründet, daß für die Ausgabe des ersten Symbols zunächst ein initialer Zustand festgelegt werden muß. Hier lassen wir dem Adversary freie Wahl über die Verteilung der Anfangszustände, modelliert als Funktion A^* über dem Zufallsband mit Ausgaben aus ganz $\mathbb{R}_1^{Q_\mathcal{C}}$.

Ein interner Zustand des Adversaries, welcher zwischen den Aufrufen gespeichert wird, würde keine größere Allgemeinheit bewirken, da der Adversary (bei Kenntnis des Zufallsbands) deterministisch ist, und somit zu einem späteren Zeitpunkt alle von früheren Aufrufen generierten Informationen neu errechnen kann (der Adversary ist in keiner Weise bzgl. der Rechenleistung beschränkt).

Formal fügt sich dies zu folgendem Konstrukt:

Definition 5.2: CHMM-Adversary
Ein *CHMM-Adversary A zu einem CHMM $\mathcal{C}$* besteht aus zwei meßbaren Abbildungen

$$A^* : [0,1)^{\mathbb{N}_0} \longrightarrow \mathbb{R}_1^{Q_\mathcal{C}}$$

und

$$A : \mathbb{N} \times [0,1)^{\mathbb{N}_0} \times Q_\mathcal{C} \times Q_\mathcal{C}^* \times \Sigma_\mathcal{C}^* \longrightarrow \mathbb{R}_1^{\Sigma_\mathcal{C} \times Q_\mathcal{C}}$$

mit

$$A(i, (r_\nu), q, (q_\nu), (\sigma_\nu)) \in \mathcal{C}_q$$

für alle $i \in \mathbb{N}$, $(r_\nu) \in [0,1)^{\mathbb{N}_0}$, $q \in Q_\mathcal{C}$, $(q_\nu) \in Q_\mathcal{C}^*$, $(\sigma_\nu) \in \Sigma_\mathcal{C}^*$.
Die Menge aller Adversaries zu $\mathcal{C}$ nennen wir $\mathrm{Adv}_\mathcal{C}$. □

Wir haben oben in Begriffen von Zuständen und Pfeilen zwischen denselben beschrieben, wie aus einem CHMM und einem Adversary eine Quelle entsteht. Dies wird nun formal definiert:

Definition 5.3: CHMM-Quelle
Sei $\mathcal{C}$ ein CHMM und $A \in \mathrm{Adv}_\mathcal{C}$. Dann ist die *CHMM-Quelle* X^A durch den folgenden Zufallsprozeß definiert:

Es seien R und R' unabhängige, auf $[0,1)^{\mathbb{N}_0}$ gleichverteilte Zufallsvariablen. Wir nehmen auf $Q_\mathcal{C}$ und $\Sigma_\mathcal{C} \times Q_\mathcal{C}$ eine beliebige aber feste Halbordnung an.

Setze dann

$$T_*^A := A^*(R')$$

und

$$Q_0^A = q \quad :\Longleftrightarrow \quad \sum_{\substack{q' \in Q_\mathcal{C} \\ q' < q}} (T_*^A)'_q \;\le\; R_0 \;<\; \sum_{\substack{q' \in Q_\mathcal{C} \\ q' \le q}} (T_*^A)'_q \tag{9}$$

Weiter seien für $i \in \mathbb{N}$, $x \in \Sigma_\mathcal{C}$ und $q \in Q_\mathcal{C}$:

$$T_i^A := A(i, R', Q_{i-1}^A, (Q_0^A, \ldots, Q_{i-1}^A), (X_1^A, \ldots, X_{i-1}^A))$$

und

$$(X_i^A, Q_i^A) = (x, q) \quad :\Longleftrightarrow \quad \sum_{\substack{(x',q') \in \Sigma_\mathcal{C} \times Q_\mathcal{C} \\ (x',q') < (x,q)}} (T_i^A)_{x',q'} \;\le\; R_i \;<\; \sum_{\substack{(x',q') \in \Sigma_\mathcal{C} \times Q_\mathcal{C} \\ (x',q') \le (x,q)}} (T_i^A)_{x',q'}. \tag{10}$$

Eine Quelle X heißt $\mathcal{C}$-Quelle, wenn X für ein $A \in \mathrm{Adv}_{\mathcal{C}}$ die gleiche Verteilung wie X^A hat.

Die Familie aller $\mathcal{C}$-Quellen schreiben wir $\mathcal{X}^{\mathcal{C}}$, und die *Symbolgewichtung* $\eta^{\mathcal{C}}$ *zu* $\mathcal{C}$ ist durch $\eta^{\mathcal{C}} := \eta^{\mathcal{X}^{\mathcal{C}}}$ definiert.
Eine Quelle X heißt *CHMM-Quelle,* wenn ein CHMM $\mathcal{C}$ existiert, so daß X eine $\mathcal{C}$-Quelle ist. □

In dieser Definition haben die verschiedenen Zufallsvariablen folgende Interpretation:

- Das Zufallsband des Adversaries wird dargestellt durch die Folge R'.
- Für jeden Zeitpunkt i stellt Q_i^A den aktuellen Zustand dar (mit Q_0^A als initialem Zustand), und X_i^A das beim Übergang von Q_{i-1}^A nach Q_i^A ausgegebene Symbol.
- Die Zufallsvariable T_i^A stellt die vom Adversary ausgewählte Verteilung für (Q_i^A, X_i^A) dar.
- Die Zufallsvariable T_*^A stellt die vom Adversary ausgewählte Verteilung des Anfangszustands dar.
- Die Zufallsvariablen R_i werden mittels der Gleichungen (9) und (10) verwandt, damit Q_0^A bzw. (X_i^A, Q_i^A) auch die vom Adversary verlangte Verteilung bekommen (für festes T_*^A bzw. T_i^A).

Da für verschiedene Adversaries ein CHMM $\mathcal{C}$ verschiedene Quellen erzeugt, definiert ein CHMM in natürlicher Weise eine Familie von Quellen $\mathcal{X}^{\mathcal{C}}$.

Damit wir von CHMM erzeugte Familien von Quellen besser in die Begrifflichkeiten aus Abschnitt 4.1 einordnen können, werde noch das folgende Lemma präsentiert:

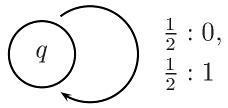
Lemma 5.4: Zeitinvarianz von CHMM-Familien
Sei $\mathcal{C}$ ein CHMM. Dann ist $\mathcal{X}^{\mathcal{C}}$ links-zeitinvariant und konditioniert links-zeitinvariant. □

Beweis siehe Abschnitt A.5.1, Seite 72.
Man beachte aber, daß $\mathcal{X}^{\mathcal{C}}$ i. a. nicht rechts-zeitinvariant ist. Ein Gegenbeispiel wird in Abschnitt A.5.2, Seite 76 gegeben.
Intuitiv begründen wir das Lemma dadurch, daß – im Falle der Links-Zeitinvarianz – ein Adversary, der die verschobene Folge $X^{(n)}$ (Notation wie in Definition 4.3) erzeugen will, einfach den Adversary der Ursprungsfolge simuliert, die Verteilung von $Q_0^{(n)}$ der von Q_n entsprechend wählt, und danach die Übergangswahrscheinlichkeiten des simulierten Adversaries übernimmt. Für die konditionierte Links-Zeitinvarianz gilt entsprechendes, nur muß hier der Simulator noch die durch die Konditionierung des Wahrscheinlichkeitsraums veränderte Verteilung des Zufallsbands des simulierten Adversaries berücksichtigen.

5.2 Beispiele

Im folgenden stellen wir einige CHMM vor, um ein erstes Gefühl für diese Konstrukte zu bekommen. Zusätzlich zu diesen einfachen Beispielen findet sich in Abschnitt 8.2.2 noch eine kommentierte Modellierung einer physikalischen Quelle.

5.2.1 Gleichverteilung

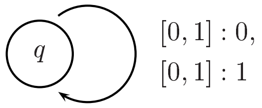


Dieses CHMM $\mathcal{C}$ modelliert eine auf $\{0,1\}^{\mathbb{N}}$ gleichverteilte Quelle. Da es nur einen einzigen Zustand gibt und der dazugehörige Transitionsbereich einelementig ist, hat der Adversary keinen Einfluß auf das Verhalten der Quelle.
Direkt aus der Definition der Symbolgewichtung (Definition 4.1) ergibt sich dann

$$\eta^{\mathcal{C}}(\alpha; x) = |x| \qquad (\alpha, x \in \{0,1\}^*).$$

Dateiname: `uniform.chmm` (siehe Abschnitt B.4).

5.2.2 Uneingeschränkter Adversary

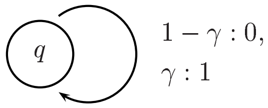


Bei diesem CHMM $\mathcal{C}$ kann der Adversary in jedem Schritt die Verteilung der Ausgabe frei wählen. Somit ist $\mathcal{X}^{\mathcal{C}}$ die Familie aller Quellen über $\{0,1\}$. Damit ergibt sich wieder direkt aus Definition 4.1

$$\eta^{\mathcal{C}}(\alpha;x) = 0 \qquad\qquad (\alpha, x \in \{0,1\}^*).$$

Dateiname: `all.chmm` (siehe Abschnitt B.4).

5.2.3 Quelle mit festem Bias



Dieses CHMM $\mathcal{C}$ beschreibt für festes $\gamma \in [0,1]$ eine Quelle, die unabhängig identisch verteilte Symbole erzeugt, wobei die Ausgabe 1 die Wahrscheinlichkeit γ hat. Auch hier hat der Adversary keinen Einfluß auf die Daten.

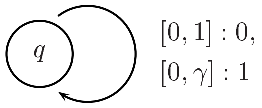
Für $\gamma = \frac{1}{2}$ entspricht $\mathcal{C}$ dem CHMM aus Abschnitt 5.2.1.

Da eine Teilsequenz $x \in \{0,1\}^*$ die Wahrscheinlichkeit $\gamma^{\omega_1(x)}(1-\gamma)^{\omega_0(x)}$ hat, unabhängig von zuvor ausgegebenen Symbolen, ist

$$\eta^{\mathcal{C}}(\alpha;x) \;=\; -\,\omega_1(x)\log\gamma \;-\; \omega_1(x)\log(1-\gamma) \qquad\qquad (\alpha, x \in \{0,1\}^*).$$

Dateiname (mit $\gamma := 0{,}6$): `biased.chmm` (siehe Abschnitt B.4).

5.2.4 Einseitig beschränkte Quelle



Dieses CHMM $\mathcal{C}$ beschreibt für festes $\gamma \in [0,1]$ Quellen, bei denen der Adversary jederzeit die Ausgabe einer 0 erzwingen kann, die Ausgabe einer 1 aber nur mit einer Wahrscheinlichkeit von höchstens γ verlangen. Es kann also z. B. (bei geeignetem Adversary) die Folge $000\ldots$ mit beliebig hoher Wahrscheinlichkeit auftreten, die Folge $111\ldots$ aber nur mit in der Länge exponentiell fallender (außer wenn $\gamma = 1$). Natürlich kann der Adversary seine Strategie nach jedem Symbol ändern.

Für $\gamma = 1$ erhalten wir das CHMM aus Abschnitt 5.2.2.

Die Gleichverteilung auf $\{0,1\}^{\mathbb{N}}$ ist genau dann in $\mathcal{X}^{\mathcal{C}}$, wenn $\gamma \geq \frac{1}{2}$.

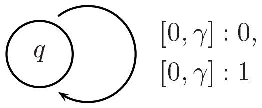
Es ist

$$\eta^{\mathcal{C}}(\alpha;x) = -\omega_1(x)\log\gamma \qquad (\alpha, x \in \{0,1\}^*),$$

der Beweis hierzu findet sich in Abschnitt A.5.3, Seite 76.

Dateiname (mit $\gamma := 0{,}6$): `oneside.chmm` (siehe Abschnitt B.4).

5.2.5 Symmetrisch beschränkte Quelle



Dieses CHMM $\mathcal{C}$ beschreibt für festes $\gamma \in [\frac{1}{2}, 1]$ Quellen, bei denen der Adversary eine 0 oder 1 jeweils mit einer Wahrscheinlichkeit von maximal γ verlangen kann, jedoch nicht erzwingen.

Die Familie $\mathcal{X}^{\mathcal{C}}$ entspricht den *slightly random sources* mit Parameter γ aus [SV86].

Ist $\gamma = \frac{1}{2}$, so ergibt sich das CHMM aus Abschnitt 5.2.1, für $\gamma = 1$ das aus Abschnitt 5.2.2. Für $\gamma < \frac{1}{2}$ läge kein CHMM vor.

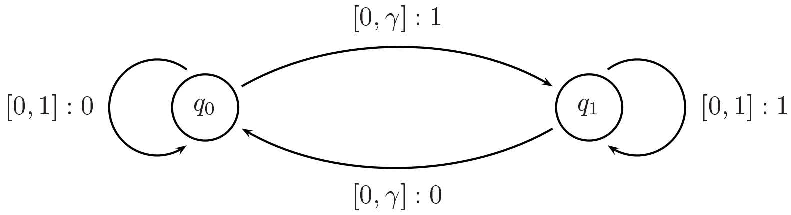
Es ist

$$\eta^{\mathcal{C}}(\alpha;x) = -|x|\log\gamma \qquad\qquad (\alpha, x \in \{0,1\}^*),$$

der Beweis steht in Abschnitt A.5.3, Seite 76.

Dateiname (mit $\gamma := 0{,}6$): `slightlyrandom.chmm` (siehe Abschnitt B.4).

5.2.6 Blockierende Quelle



Dieses CHMM $\mathcal{C}$ beschreibt für festes $\gamma \in [0,1]$ eine Quelle, bei der der Adversary jederzeit erzwingen kann, daß das gleiche Symbol wie im vorangegangenen Schritt ausgegeben wird, aber einen Wechsel der Ausgabe nur mit einer Wahrscheinlichkeit von maximal γ anfordern kann. Es sind also z. B. (bei geeignetem Adversary) die Folgen $000\ldots$ und $111\ldots$ mit beliebig hoher Wahrscheinlichkeit möglich, die Folge $010101\ldots$ aber kann maximal mit einer in ihrer Länge exponentiell fallender Wahrscheinlichkeit auftreten (sofern $\gamma \neq 1$).

Für $\gamma = 1$ erhalten wir ein CHMM mit gleichem Verhalten wie das aus Abschnitt 5.2.2.

Die Gleichverteilung auf $\{0,1\}^{\mathbb{N}}$ ist genau dann in $\mathcal{X}^{\mathcal{C}}$, wenn $\gamma \geq \frac{1}{2}$.

Es bezeichne $\kappa(x)$ für $x = x_1 \ldots x_n \in \{0,1\}^*$ die Anzahl der $i \in \{1, \ldots, n-1\}$ mit $x_i \neq x_{i+1}$. Dann ist

$$\eta^{\mathcal{C}}(\alpha; x) = \begin{cases} -\kappa(x)\log\gamma, & \text{falls } \alpha = \lambda, \\ -\kappa(\alpha_{|\alpha|}x)\log\gamma, & \text{sonst}, \end{cases} \qquad (\alpha, x \in \{0,1\}^*),$$

der Beweis hierzu findet sich in Abschnitt A.5.4, Seite 77.

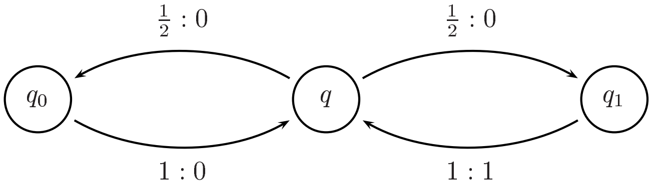
Dateiname (mit $\gamma := 0{,}6$): `stalling.chmm` (siehe Abschnitt B.4).

5.2.7 Ungleichheit in Lemma 4.2

In Lemma 4.2 haben wir gesehen, daß immer

$$\eta^{\mathcal{X}}(\alpha; x_1x_2) \geq \eta^{\mathcal{X}}(\alpha; x_1) + \eta^{\mathcal{X}}(\alpha x_1; x_2)$$

gilt. Daß selbst für von CHMM erzeugte Familien hier i. a. keine Gleichheit gilt (wie bereits in Fußnote 7, Seite 21 erwähnt), zeigt das folgende CHMM $\mathcal{C}$:



Hier ist

$$\begin{aligned} \eta^{\mathcal{C}}(\lambda; 00) &= 0, \\ \eta^{\mathcal{C}}(00; 0) &= 0, \\ \eta^{\mathcal{C}}(\lambda; 000) &= 1. \end{aligned}$$

Zum Beweis siehe Abschnitt A.5.5, Seite 78.

Dateiname: `nocompose.chmm` (siehe Abschnitt B.4).

5.3 Berechnung der Symbolgewichtung

Wie in Kapitel 4 gesehen, brauchen wir, um aus einer Familie von CHMM-Quellen Zufall zu extrahieren, eine möglichst gute untere Abschätzung der Symbolgewichtung. Wie die folgenden Sätze zeigen, können wir bei CHMM-Quellen die Symbolgewichtung sogar (im Rahmen der Rechengenauigkeit) exakt bestimmen.

Satz 5.5: Berechnung der Symbolgewichtung von CHMM

Es sei $\mathcal{C}$ ein CHMM und $x \in \Sigma^{\mathbb{N}}$. Weiterhin seien die folgenden Abbildungen auf $2^{\mathbb{R}_{\geq 0}^{Q_\mathcal{C}}}$ definiert:

$$\mathcal{N}(M) := \left\{\tfrac{p}{\|p\|_1} : p \in M \setminus \{0\}\right\},$$
$$\mathcal{T}_x^\mathcal{C}(M) := \left\{\Big(\sum_{q' \in Q_\mathcal{C}} t_{x,q}^{(q')} p_{q'}\Big)_q \,:\, t^{(q')} \in \overline{\overline{\mathcal{C}}}_{q'},\ p \in M\right\},$$

wobei $\overline{\overline{\mathcal{C}}}_{q'}$ den topologischen Abschluß (im folgenden immer kurz Abschluß genannt) der konvexen Hülle von $\mathcal{C}_{q'}$ bezeichne.

Betrachte folgende Rekursion:

$$\begin{aligned} \mathcal{P}_{1,1} &:= \mathcal{T}_{x_1}^\mathcal{C}(\mathbb{R}_1^{Q_\mathcal{C}}), \\ \mathcal{P}_{j,j} &:= \mathcal{T}_{x_j}^\mathcal{C} \circ \mathcal{N}(\mathcal{P}_{i,j-1}) && (1 \leq i < j), \\ \mathcal{P}_{i,j} &:= \mathcal{T}_{x_j}^\mathcal{C}(\mathcal{P}_{i,j-1}) && (1 \leq i < j). \end{aligned} \tag{11}$$

Hierbei ist (11) wohldefiniert.

Dann ist

$$\eta^\mathcal{C}(x_1 \ldots x_{i-1};\ x_i \ldots x_j) = -\log \max_{p \in \mathcal{P}_{i,j}} \|p\|_1 \qquad (1 \leq i \leq j).$$ □

Beweis siehe Abschnitt A.5.6, Seite 78.

Diese Rekursionsformel erlaubt es uns, in $O(j)$ Anwendungen von $\mathcal{T}_x^\mathcal{C}$ und $\mathcal{N}$ die Symbolgewichtung $\eta^\mathcal{C}(x_1 \ldots x_{i-1};\, x_i \ldots x_j)$ zu bestimmen. Nichtsdestotrotz ist dieser Satz für die algorithmische Implementation noch nicht sehr zweckdienlich, da die einzelnen Schritte auf Mengen von Verteilungen operieren, welche i. a. unendlich sein werden. Wir müssen deshalb eine endliche und algorithmisch handhabbare Repräsentation dieser Mengen finden. Wir werden feststellen, daß ein konvexes Erzeugendensystem eine solche Repräsentation ist, welche zwar die jeweilige Menge nur bis auf konvexe Äquivalenz (s. u.) festlegt, was aber auf das Endergebnis keinen Einfluß hat, wie Lemma 5.10 zeigen wird.

Zunächst stellen wir fest, daß die Repräsentation der Transitionsbereiche eines CHMM durch konvexe Erzeugendensysteme zulässig ist:

Definition 5.6: Konvexe Äquivalenz

Zwei Teilmengen A, B eines $\mathbb{R}$-Moduls M heißen *konvex-äquivalent,* wenn ihre konvexen Hüllen gleich sind.

Zwei CHMM $\mathcal{C}$ und $\mathcal{C}'$ heißen *konvex-äquivalent,* wenn $\Sigma_\mathcal{C} = \Sigma_{\mathcal{C}'}$, $Q_\mathcal{C} = Q_{\mathcal{C}'}$, und zusätzlich $\mathcal{C}_q$ und $\mathcal{C}'_q$ für jedes $q \in Q_\mathcal{C}$ konvex-äquivalent sind.

Die CHMM $\mathcal{C}$ und $\mathcal{C}'$ heißen *fast konvex-äquivalent,* wenn $\Sigma_\mathcal{C} = \Sigma_{\mathcal{C}'}$, $Q_\mathcal{C} = Q_{\mathcal{C}'}$, und zusätzlich $\mathcal{C}_q$ und $\mathcal{C}'_q$ für jedes $q \in Q_\mathcal{C}$ den gleichen Abschluß der konvexen Hülle haben. □

Lemma 5.7: Konvex-äquivalente CHMM

Sind $\mathcal{C}$ und $\mathcal{C}'$ konvex-äquivalente CHMM, so ist $\mathcal{X}^\mathcal{C} = \mathcal{X}^{\mathcal{C}'}$. □

Beweis siehe Abschnitt A.5.7, Seite 82.

Die Grundidee des Beweises ist recht einfach. Will ein Adversary eine Verteilung p erzeugen, die in der konvexen Hülle der erlaubten Verteilungen liegt, so ermittelt er eine Konvexkombination $p = \sum r_i p_i$ und wählt mit Wahrscheinlichkeit r_i die Verteilung p_i.

Dieses Lemma (5.7) wird genaugenommen nicht benötigt, um zu zeigen, daß wir für die Berechnung der Symbolgewichtung die Transitionsbereiche konvex repräsentieren können, die ergibt sich einfacher bereits daraus, daß in Satz 5.5 nur der Abschluß der konvexen Hülle von $\mathcal{C}_q$ verwandt wird, nie $\mathcal{C}_q$ direkt. Das Lemma mit seiner stärkeren Aussage sei aber der Vollständig halber angegeben (und weil es im Beweis von Satz 5.5 benutzt wird).

Hiermit können wir noch ein paar Klassen von CHMM spezifizieren, die sich für eine algorithmische Bearbeitung besonders eignen.

Definition 5.8: Endlich repräsentierbare CHMM

Ein CHMM $\mathcal{C}$ heißt *endlich,* wenn alle $\mathcal{C}_q$ $(q \in Q_\mathcal{C})$ endlich sind.

Ein CHMM $\mathcal{C}$ heißt *endlich repräsentierbar,* wenn ein endliches CHMM $\mathcal{C}'$ existiert, welches konvex-äquivalent zu $\mathcal{C}$ ist.

Ein CHMM $\mathcal{C}$ heißt *fast endlich repräsentierbar,* wenn ein endliches CHMM $\mathcal{C}'$ existiert, welches fast konvex-äquivalent zu $\mathcal{C}$ ist. □

Der letzte Teil dieser Definition ist klar an die Details von Satz 5.5 angelehnt, da dort von einem CHMM nur die Abschlüsse der konvexen Hüllen verwendet werden. Zu einem fast endlich repräsentierbaren CHMM gibt es also insbesondere ein endliches mit gleicher Symbolgewichtung.

Interessant ist noch der folgende Sachverhalt:

Lemma 5.9: Repräsentierbarkeit von durch Diagramme definierten CHMM

Läßt sich ein CHMM durch ein Diagramm mit beigefügten Gleichungen und Ungleichungen darstellen (wie vor Definition 5.1 erläutert), und sind diese Gleichungen und Ungleichungen linear, sowie alle an den Pfeilen notierten Wahrscheinlichkeitsmengen Intervalle, so ist das CHMM fast endlich repräsentierbar.

Sind zusätzlich alle an den Pfeilen angegebenen Wahrscheinlichkeitsmengen abgeschlossen, und kommen in den Gleichungen und Ungleichungen nur die Relationen $\leq$, $\geq$ und $=$ vor (nicht $<$ oder $>$), so ist das CHMM sogar endlich repräsentierbar. □

Ein Beweis findet sich in Abschnitt A.5.8, Seite 84.

Zu guter Letzt präsentieren wir das Lemma, welches uns eine algorithmische Implementation von Satz 5.5 ermöglicht:

Lemma 5.10: Konvexität der Rekursion in Satz 5.5

Sind $\mathcal{C}$ und $\mathcal{C}'$ zwei fast konvex-äquivalente CHMM, $x \in \Sigma_\mathcal{C}$, $\mathcal{T}_x$ und $\mathcal{N}$ wie in Satz 5.5, sowie $\mathcal{P}, \mathcal{P}' \subseteq \mathbb{R}_1^{Q_\mathcal{C}}$ konvex-äquivalent, dann sind

$$\mathcal{N}(\mathcal{P}) \approx \mathcal{N}(\mathcal{P}'), \qquad \mathcal{T}_x^\mathcal{C}(\mathcal{P}) \approx \mathcal{T}_x^{\mathcal{C}'}(\mathcal{P}') \qquad \text{und} \qquad -\log \sup_{p \in \mathcal{P}} \|p\|_1 = -\log \sup_{p \in \mathcal{P}'} \|p\|_1,$$

wobei $\approx$ konvexe Äquivalenz meine. □

Zum Beweis siehe Abschnitt A.5.9, Seite 84.

Ist also $\mathcal{C}$ ein fast endlich repräsentierbares CHMM, so kann der folgende Algorithmus zur Berechnung von $\eta^\mathcal{C}(x_1, \ldots, x_{i-1};\, x_i \ldots x_j)$ herangezogen werden (es bedeute $\approx$ konvexe Äquivalenz):

1. Es sei $\mathcal{C}'$ ein endliches, zu $\mathcal{C}$ fast konvex-äquivalentes CHMM.
2. Setze $P := \{e_i : i \in Q_\mathcal{C}\}$. (Da die Einheitsvektoren von $\mathbb{R}^{Q_\mathcal{C}}$ nach Definition von $\mathbb{R}_1^{Q_\mathcal{C}}$ ebendiese Menge konvex aufspannen, ist $P \approx \mathbb{R}_1^{Q_\mathcal{C}}$.)
3. Für $\nu := 1$ bis einschließlich i wiederhole Schritte 4 und 5. (Nach Beendigung der Schleife ist $P \approx \mathcal{P}_{i,i}$.)
4. Setze $P := \mathcal{N}(P)$ und reduziere P (s. u.). (Dann ist $P \approx \mathcal{N}(\mathcal{P}_{\nu-1,\nu-1})$ für $i > 1$ bzw. $P \approx \mathbb{R}_1^{Q_\mathcal{C}} = \mathcal{N}(\mathbb{R}_1^{Q_\mathcal{C}})$ für $i = 1$. Dieser Schritt kann für $i = 1$ ohne Änderung der Invarianten auch weggelassen werden.)
5. Setze $P := \mathcal{T}_{x_\nu}^{\mathcal{C}'}(P)$ und reduziere P. (Dann ist $P \approx \mathcal{P}_{\nu,\nu}$.)
6. Für $\nu := i+1$ bis einschließlich j wiederhole den folgenden Schritt. (Ist $j = i$, so wird die Schleife keinmal durchlaufen. Nach Beendigung der Schleife ist $P \approx \mathcal{P}_{i,j}$.)
7. Setze $P := \mathcal{T}_{x_\nu}^{\mathcal{C}'}(P)$ und reduziere P. (Dann ist $P \approx \mathcal{P}_{i,\nu}$.)
8. Berechne $\eta := -\log \max_{p \in P} \|p\|_1$. (Dann ist $\eta = \eta^\mathcal{C}(x_1, \ldots, x_{i-1};\, x_i \ldots x_j)$.)

Steht in diesem Algorithmus „reduziere P“, so ist damit gemeint, daß P durch eine (möglichst, aber nicht notwendig) kleinere Menge mit gleicher konvexer Hülle zu ersetzen ist. Die Reduktion kann auch die Identität sein (also weggelassen werden). Algorithmen zur Reduktion behandeln wir in dieser Arbeit nicht.

Bei jeder Operation in diesem Algorithmus darf P auch durch eine Menge ersetzt werden, welche eine Obermenge von P als konvexes Erzeugnis hat. Dann sind alle Invarianten entsprechend neu zu formulieren („die konvexe Hülle von P ist eine Obermenge der konvexen Hülle von M“ statt „$P \approx M$“), und am Ende erhalten wir $\eta \leq \eta^{\mathcal{C}}(x_1, \ldots, x_{i-1}; x_i \ldots x_j)$, also eine untere Abschätzung der Symbolgewichtung.

Diese Variante kann u. U. von Nutzen sein, wenn Rundungsfehler keine exakte Berechnung des neuen Inhalts von P ermöglichen, oder wenn dadurch die Reduktion von P wesentlich besser möglich ist.

Wird die Reduktion weggelassen, so wird dieser Algorithmus offensichtlich schlechtestenfalls eine in i exponentielle Laufzeit haben, da $\#P$ exponentiell in i wachsen kann. Bei Ersetzung von P durch jeweils ein minimales Erzeugendensystem ist uns kein Beispiel dafür bekannt, daß $\#P$ stärker als polynomiell in i wächst, allerdings auch kein Gegenbeweis.

Die Auswertung von $\mathcal{N}$ ist offenbar linear in i bei geeigneter Datenstruktur für P, die Auswertung von $\mathcal{T}_x^{\mathcal{C}'}$ aber hat bei naiver Auswertung der Summenformel eine Laufzeit von

$$O\Big(\#Q_{\mathcal{C}} \prod_{q \in Q_{\mathcal{C}}} \#C'_q\Big),$$

wahrscheinlich ist es i. a. auch nicht besser möglich.

Zusammenfassend kann über die Laufzeit des Algorithmus gesagt werden, daß er nur für CHMM mit wenig Pfeilen geeignet ist, unter diesen aber möglicherweise auch für große i, j und $j - i$ effizient (bei geeigneter Reduktionsmethode).

Kapitel 6

Formale Sicherheit

6.1 Klassische Sicherheitsdefinition

In der Kryptologie ist es wichtig, formal zeigen zu können, daß ein vorliegendes Verfahren sicher ist. Daher sind im Laufe der Zeit für verschiedene Anwendungen verschiedene Sicherheitsbegriffe entworfen worden, die meist aus Aufzählungen verschiedener Eigenschaften bestehen, die spezifisch für die jeweilige Anwendung sind. Dies bedeutet aber, daß beispielsweise der Begriff der Sicherheit von Verschlüsselungsverfahren ein ganz anderer ist als der von Authentifikation. Man kann dies nun fortführen und Generatoren von Zufallsfolgen, die abbrechen können, als sicher bezeichnen, wenn die von ihnen erzeugten Folgen ε_k-zufällig sind, wobei $\varepsilon_k \in \mathbb{R}$ in einem Sicherheitsparameter k hinreichend schnell gegen Null gehe. Eine exakte Formulierung bilden die nachfolgenden zwei Definitionen.

Definition 6.1: Parametrische Familie von Quellen
Sei $\mathcal{B}$ eine Familie von Quellen. Eine parametrische Familie $\mathcal{X}$ von Quellen besteht aus einer nichtleeren Menge $I_{\mathcal{X}}$, der *Indexmenge,* und einer Abbildung

$$\mathcal{X} : \mathbb{N} \times I_{\mathcal{X}} \longrightarrow \mathcal{B}. \qquad \square$$

Wir interpretieren diese Definition wie folgt: Der Anwender eines durch $\mathcal{X}$ beschriebenen Zufallsgenerators kann einen Sicherheitsparameter $k \in \mathbb{N}$ frei wählen; je größer dieser ist, desto zufälliger soll die Quelle sein. Der Index $i \in I_{\mathcal{X}}$ ist dem Anwender i. a. unbekannt und von ihm nicht beeinflußbar. Dann ist die Ausgabe des Zufallsgenerators die Quelle $\mathcal{X}(k, i)$.

Eine solche Quelle kann einer oder mehreren der folgenden Sicherheitsdefinitionen genügen:

Definition 6.2: Exponentiell/superpolynomiell/perfekt zufällig
Sei $\mathcal{X}$ eine parametrische Familie von Quellen. Dann heißt $\mathcal{X}$ *exponentiell zufällig,* wenn ein $c > 1$ existiert, so daß für hinreichend große $k \in \mathbb{N}$ für jedes $i \in I_{\mathcal{X}}$ die Quelle $\mathcal{X}(k, i)$ eine (c^{-k})-zufällige ist.

Weiter heißt $\mathcal{X}$ *superpolynomiell zufällig,* wenn eine superpolynomielle Funktion $f : \mathbb{N} \to \mathbb{R}_{>0}$ existiert,[16] so daß für hinreichend große $k \in \mathbb{N}$ für jedes $i \in I_{\mathcal{X}}$ die Quelle $\mathcal{X}(k, i)$ eine $(f(k)^{-1})$-zufällige ist.

Schließlich heißt $\mathcal{X}$ *perfekt zufällig,* wenn jedes $\mathcal{X}(k, i)$ ($k \in \mathbb{N}$, $i \in I_{\mathcal{X}}$) perfekt zufällig ist. $\square$

Die superpolynomielle und die exponentielle Variante unterscheiden sich dadurch, wie stark der Sicherheitsparameter in der Qualität steigt.

Der stärkste dieser Begriffe ist der der perfekten Zufälligkeit, er impliziert die beiden anderen.

Hiernach folgt der der exponentiellen Zufälligkeit, diese impliziert die superpolynomielle. Exponentielle Zufälligkeit läßt sich mit Satz 4.8 bzw. Korollar 4.9 aus geeigneten Quellen extrahieren.

6.2 Vergleichende Sicherheitsdefinition

Die oben beschriebene Methode, Sicherheitsbegriffe zu finden, hat zwei schwerwiegende Nachteile:

- Für jede Klasse von kryptographischen Verfahren muß ein neuer Sicherheitsbegriff entwickelt werden (wie z. B. der obige).
- In vielen Fällen ist die Liste der geforderten Eigenschaften eine immer weiter wachsende, da im Laufe der Zeit immer neue Anforderungen an das Verfahren entdeckt werden.

Um diesen Problemen beizukommen, sind in den letzten Jahren allgemeine Sicherheitsmodelle aufgekommen, die den folgenden Ansatz verfolgen:

- Es wird zunächst eine *ideale Funktionalität* (oder auch *trusted party*) formuliert, welche als sicher definiert wird. (Im Falle der Verschlüsselung beispielsweise würde diese Funktionalität von einer Partei Daten entgegennehmen und einer anderen diese wieder ausliefern, ohne daß dritte etwas über diese Daten erfahren (außer vielleicht der Länge). Einer solchen Funktionalität können wir großes Vertrauen entgegenbringen.)

[16] Das heißt für jedes Polynom p gilt für hinreichend großes k, daß $f(k) > p(k)$.

- In einem von vornherein definierten Kommunikationsmodell agieren nun eine Umgebung (*environment*), ein Adversary, evtl. eine oder mehrere Funktionalitäten und eine Anzahl von Parteien. Alle Teilnehmer kommunizieren in beliebiger Weise miteinander, mit der Einschränkung, daß die Umgebung und die Funktionalitäten nicht miteinander kommunizieren können. Zum Schluß gibt die Umgebung ein Bit aus.

- Soll ein Protokoll (das *Real-Life-Protokoll*) als sicher gelten, so muß es die folgende Anforderung erfüllen:

  Für jeden Adversary $\mathcal{A}$ (den *real life adversary*) muß ein Adversary $\mathcal{S}$ (der *ideal adversary*) existieren, so daß für jede Umgebung $\mathcal{Z}$ die Ausgaben der Umgebung in den folgenden beiden Konfigurationen ungefähr die gleiche Verteilung haben:[17]

  – Die Umgebung $\mathcal{Z}$ zusammen mit dem Real-Life-Adversary $\mathcal{A}$ und dem zu überprüfenden Protokoll.

  – Die Umgebung $\mathcal{Z}$ zusammen mit dem idealen Adversary $\mathcal{S}$ und der idealen Funktionalität.

Dieses Verfahren führt zu der folgenden Anforderung an jedes Protokoll, daß als sicher eingestuft werden soll: Was auch immer der Real-Life-Adversary bei einer Ausführung des Protokolls erreichen kann, das kann der ideale Adversary auch mit der idealen Funktionalität erreichen. Da wir aber die ideale Funktionalität a priori als sicher eingestuft haben, und beim realen Protokoll nichts passieren kann, was nicht auch bei der idealen Funktionalität geschehen könnte, dürfen wir mit Fug und Recht auch das Protokoll als sicher bezeichnen.

Man beachte, daß das Attribut *sicher* allein hier noch nicht aussagekräftig ist, man muß (sofern dies aus dem Kontext nicht klar ersichtlich ist) immer mit angeben, welche Funktionalität sicher realisiert wird.

Wir haben oben die Hauptnachteile der klassischen Sicherheitsdefinitionen aufgezählt, darum wollen wir auch die der vergleichenden (oder simulierenden) Sicherheitsmodelle nicht verschweigen:

- Für jede Anwendung muß eine neue Funktionalität definiert werden. Hier muß mit großer Vorsicht vorgegangen werden, denn hat die Funktionalität Schwachstellen, so erfüllen auch Protokolle mit den gleichen Schwachstellen die Sicherheitsdefinitionen. In manchen Fällen ist die Definition von Funktionalitäten relativ einfach, in anderen aber sind viele Details zu beachten.

- Das zugrundeliegende Kommunikationsmodell ist für die Sicherheitsdefinition von großer Relevanz. Ist das Kommunikationsmodell unrealistisch, so ist auch die Sicherheitsdefinition nicht viel wert. Aufgrund des relativ hohen Detailreichtums dieser Modelle (es müssen auch komplexe Aktionen wie Korruption von Parteien berücksichtigt werden), ist es schwierig, die Qualität eines Kommunikationsmodells einzuschätzen.

- Aufgrund des Detailreichtums der Kommunikation tendieren Sicherheitsbeweise zu hoher Komplexität, sie werden dann nicht formal geführt und können deshalb leicht unentdeckte Fehler enthalten.

Vorschläge für Sicherheitsmodelle finden sich u. a. in [Can00, PW94, Bea91, Unr02]. Wir legen dieser Arbeit das in [Can00] dargestellte zugrunde (im folgenden kurz das *Canetti-Modell* genannt), dieses Kapitel und die zugehörigen Beweise in Anhang A sollten aber auch mit obigen Erläuterungen allein weitgehend verständlich sein.

Es stellt sich nun die Frage, inwiefern sich die Ergebnisse der vorliegenden Arbeit im Rahmen einer Definition der Sicherheit von *Protokollen* einordnen lassen, und inwieweit sich dieser Aufwand lohnt. Das Generieren von Zufall kann als Ein-Parteien-Protokoll aufgefaßt werden, auf Anfrage beginnt die Partei zu rechnen und gibt ein oder mehrere Symbole zurück. Für sich allein stehend ist dann ein Sicherheitsergebnis in einem solchen Modell noch nicht sehr gewinnbringend, jedoch ist in vielen Sicherheitsmodellen eine sogenannte *Komposition* von Protokollen möglich (siehe z. B. [Can00, Unr02]). Dies besagt in etwa das folgende:

- Ist bewiesen, daß ein Protokoll π, welches eine Funktionalität $\mathcal{F}$ benutzt, sicher ist (d. h. eine Funktionalität $\mathcal{G}$ sicher realisiert), und ist weiterhin bewiesen, daß ein Protokoll ϱ die Funktionalität $\mathcal{F}$ realisiert, so ist auch das Protokoll π^ϱ sicher, welches dadurch entsteht, daß Aufrufe von $\mathcal{G}$ durch Aufrufe von ϱ ersetzt werden.

Die eben vorgestellte Form der Komposition ist die *universelle Komposition,* es existieren auch schwächere Formen der Komposition, die z. B. einfach nur die Hintereinander- oder Parallelausführung mehrerer Instanzen desselben Protokolls erlauben, diese sind aber für unserem Fall nicht mächtig genug.

Die Komposition läßt sich nun wie folgt anwenden: Wenn wir gezeigt haben, daß eine Zufallsquelle sicher ist (was dies genau heißt, d. h. welche Funktionalität wir zugrunde legen, sehen wir weiter unten), und wenn

[17] Das heißt der statistische Abstand fällt hinreichend schnell (z. B. exponentiell oder superpolynomiell, je nach Definition).

wir ein Protokoll haben, welches unter Benutzung einer idealen Zufallsquelle sicher ist, so ist dieses Protokoll auch sicher unter Benutzung der realen Quelle.

Wir müssen nun definieren, was wir als sichere Quelle definieren. Hierzu schlagen wir zunächst die folgende Variante vor:

Definition 6.3: Funktionalität $\mathcal{F}_{\mathrm{Rnd},\Sigma}$: Nicht abbrechende Zufallsquelle
Sei Σ nichtleer und endlich. Dann hat die ideale Funktionalität $\mathcal{F}_{\mathrm{Rnd},\Sigma}$ das folgende Verhalten:

- Für jede Partei P_j initialisiere die Variable $p_j := 1$ (Nummer des nächsten Symbols).
- Bei Empfang der Nachricht (random) von Partei P_j, wähle zufällig gleichverteilt $\sigma \in \Sigma$ und sende $(\mathit{data}, p_j, \sigma)$ an P_j. Setze $p_j := p_j + 1$.
- Ignoriere alle anderen Nachrichten. □

Es ist offensichtlich, daß wir diese Funktionalität mit unseren Mechanismen i. a. nicht realisieren können, da unsere Quellen in manchen Fällen aufhören, Daten zu liefern, was der Funktionalität $\mathcal{F}_{\mathrm{Rnd},\Sigma}$ aber untersagt ist. Die Stärke unserer Quellen liegt darin, daß, *falls* Daten ausgegeben werden, diese von hoher Qualität sind. Um dies widerzuspiegeln, definieren wir die folgende Funktionalität:

Definition 6.4: Funktionalität $\mathcal{F}_{\mathrm{ARnd},\Sigma}$: Abbrechende Zufallsquelle
Sei Σ nichtleer und endlich. Dann zeigt die ideale Funktionalität $\mathcal{F}_{\mathrm{ARnd},\Sigma}$ das folgende Verhalten:

- Für jede Partei P_j initialisiere die Variablen $p_j := 1$ (Nummer des nächsten Symbols) und $q_j := 0$ (wenn $q_j = 1$, hält die Quelle für P_j an).
- Bei Empfang der Nachricht (init) von Partei P_j sende (init, j) an den Adversary.
- Bei Empfang der Nachricht (stop, j) vom Adversary, setze $q_j := 1$ und sende (nodata, p_j) an P_j.
- Bei Empfang einer Nachricht der Form (random) von Partei P_j unterscheide die folgenden Fälle:
  - Es wurde noch keine Nachricht (init) von P_j empfangen. Dann ignoriere die aktuelle Nachricht.
  - Es ist $q_j = 1$. Dann sende (nodata, p_j) an P_j. Setze $p_j := p_j + 1$.
  - Andernfalls wähle $\sigma \in \Sigma$ zufällig gleichverteilt und sende $(\mathit{data}, p_j, \sigma)$ an P_j. Setze $p_j := p_j + 1$.
- Ignoriere alle anderen Nachrichten. □

Diese Funktionalität erlaubt es dem Adversary, jederzeit die weitere Auslieferung von Daten an eine Partei zu unterbinden. Dabei werden die Parteien getrennt gehandelt, um die Tatsache wiederzuspiegeln, daß jede Partei eine eigene, von den anderen unabhängige Zufallsquelle benutzt. Da das Scheduling im Canetti-Modell den Adversary erst aktiviert, wenn die Funktionalität die Anforderung von Zufall beantwortet hat, ist das Unterbinden einer weiteren Auslieferung von Daten wie folgt vorgesehen:

- Die Funktionalität erhält eine Nachricht (random).
- Sie antwortet mit einer Nachricht $(\mathit{data}, \ldots)$.
- Der Adversary ist für die Auslieferung dieser Nachricht zuständig. Will er, daß keine weiteren Daten geliefert werden, so unterbindet er die Auslieferung der Nachricht und teilt dies mittels $(\mathit{stop}, \ldots)$ der Funktionalität mit.
- Die Funktionalität sendet bei Empfang nun erneut eine Antwort, nämlich $(\mathit{nodata}, \ldots)$, welche vom Adversary dann ausgeliefert werden kann.

Von jeder Partei wird außerdem vor der ersten Benutzung der Quelle eine Initialisierung mittels der Nachricht (*init*) verlangt. Dies ist gefordert, damit die Benutzung dieser Funktionalität aus Sicht der Parteien (d. h. der Umgebung) der der Funktionalität $\mathcal{F}_\mathcal{X}$ (siehe Definition 6.5) gleicht.

Zu guter Letzt muß noch eine Definition folgen, welche beliebige Familien von Quellen modelliert. Aufgrund der Struktur von Sicherheitsmodellen bietet sich als zugrundeliegendes Objekt eine parametrische Familie von Quellen mit einer Menge von Wörtern als Indexmenge an. Wir erhalten dann folgende Funktionalität:

Definition 6.5: Funktionalität $\mathcal{F}_\mathcal{X}$: Quellenfamilie $\mathcal{X}$
Sei Σ_I nichtleer und endlich, sowie $\mathcal{X}$ eine parametrische Quellenfamilie mit $I_\mathcal{X} \subseteq \Sigma_I^*$. Die Quellenfamilien $\mathcal{X}^{(j)}$ seien dann Kopien von $\mathcal{X}$, d. h. es haben $\mathcal{X}^{(j)}(k,i)$ und $\mathcal{X}^{(j')}(k,i)$ die gleiche Verteilung, aber alle Quellen $\mathcal{X}^{(j)}(k,i)$ sind stochastisch unabhängig.

Dann zeigt die ideale Funktionalität $\mathcal{F}_\mathcal{X}$ bei Sicherheitsparameter k das folgende Verhalten:

- Für jede Partei P_j initialisiere die Variablen $p_j := 1$ (Nummer des nächsten Symbols) und $s_j := o$ (Index der zu verwendenden Quelle), wobei o ein festes Element von I sei (z. B. das lexikalisch kleinste).
- Bei Empfang der Nachricht (*init*) von Partei P_j sende (init, j) an den Adversary.
- Bei Empfang der Nachricht (source, j, i) vom Adversary mit $i \in I_\mathcal{X}$ setze $s_j := i$, es sei denn $p_j > 1$.
- Bei Empfang einer Nachricht der Form (*random*) von Partei P_j unterscheide die folgenden Fälle:
  - Es wurde noch keine Nachricht (*init*) von P_j empfangen. Dann ignoriere die aktuelle Nachricht.
  - Sei $\sigma := \big(\mathcal{X}^{(j)}(k,s_j)\big)_{p_j}$. Ist $\sigma = \perp$, so sende (nodata, p_j) an P_j, sonst $(\mathit{data}, p_j, \sigma)$. Setze $p_j := p_j + 1$.
- Andere Nachrichten ignoriere. □

Diese Funktionalität überläßt dem Adversary die Wahl der Quelle, wie dies in der Interpretation parametrischer Familien von Quellen auf Seite 37 bereits angedeutet wurde.

Damit der Adversary die Möglichkeit hat, die Quelle festzulegen, muß er noch vor der ersten (*random*)-Nachricht aktiviert werden. Um dies zu garantieren, wurde in die Spezifikation mit aufgenommen, daß jede Partei ihre Quelle mit einer (*init*)-Nachricht initialisieren muß.

Mit diesen Definitionen gerüstet können wir nun folgendes untersuchen: Sei $\mathcal{X}$ eine parametrische Familie von Quellen, die einen der Sicherheitsbegriffe aus Definition 6.2 erfüllt. Realisiert dann die Funktionalität $\mathcal{F}_\mathcal{X}$ [18] sicher die ideale Funktionalität $\mathcal{F}_{\mathrm{ARnd},\Sigma_\mathcal{X}}$? Leider müssen wir dies verneinen, wie das Beispiel in Abschnitt A.6.1, Seite 86 zeigt.

Es ist also nötig, noch die folgende Bedingung zu stellen:

Definition 6.6: Simulierbare Familie von Quellen
Eine parametrische Familie $\mathcal{X}$ von Quellen heißt *simulierbar,* wenn $I_\mathcal{X}$ eine effizient entscheidbare Sprache über einem Alphabet Σ_I ist und es eine probabilistische Turingmaschine M mit folgenden Eigenschaften gibt:

- Es existiert ein Polynom p, so daß für alle $k \in \mathbb{N}$, $i \in I$, $n \in \mathbb{N}$, $d \in \{0,1\}^*$ die Turingmaschine M bei Eingabe (k,i,n,d) höchstens $p(k + |i| + n)$ Schritte läuft und eine Ausgabe der Form (σ, d') liefert mit $\sigma \in \Sigma_\mathcal{X}$ und $d' \in \{0,1\}^*$.[19]
- Sei D_0 eine konstante Zufallsvariable mit Wert λ. Es entstehe $Y^{(k,i)}$ ($i \in I_\mathcal{X}$, $k \in \mathbb{N}$) durch folgenden Zufallsprozess:
  $$(D_n^{(k,i)}, Y_n^{(k,i)}) := M(k,i,n,D_{n-1}^{(k,i)}) \qquad (n \geq 1).$$
  Dann existieren eine von k,i unabhängige superpolynomielle Funktion $f : \mathbb{N} \to \mathbb{R}_{>0}$ und ein von k,i unabhängiges Polynom p, so daß für hinreichend großes $k \in \mathbb{N}$ für jedes $l \in \mathbb{N}$ gilt:
  $$\mathrm{SD}\big(Y_1^{(k,i)} \ldots Y_l^{(k,i)};\ \big(\mathcal{X}(k,i)\big)_1 \ldots \big(\mathcal{X}(k,i)\big)_l\big) \leq \frac{p(l)}{f(k)}. \tag{12}$$

[18] Das heißt genaugenommen das Protokoll, welches alle Anfragen direkt an diese Funktionalität weiterleitet.
[19] Die Ein- und Ausgaben seien in geeigneter Weise kodiert.

Ist in (12) das Polynom $p = 0$, so sprechen wir von einer *exakt simulierbaren Familie von Quellen.* □

Diese Eigenschaft besagt, daß die Quelle durch eine strikt polynomielle probabilistische Turingmaschine simulierbar ist.

Ob die Eigenschaft, bezogen auf eine tatsächlich implementierte Quelle, realistisch ist, oder ob man den bei der Auswahl des Symbols eventuell auftretenden chaotischen Prozessen superpolynomielle Rechenleistung zuschreibt, ist eine Frage, deren Untersuchung den Rahmen dieser Arbeit sprengen würde; die Entscheidung bleibt dem Leser überlassen.

Es ergibt sich der schließlich der folgende Satz:

Satz 6.7: Sicherheit von $\mathcal{F}_{\mathcal{X}}$
Ist $\mathcal{X}$ eine simulierbare und superpolynomiell zufällige Familie von Quellen, so wird $\mathcal{F}_{\mathrm{ARnd},\Sigma_{\mathcal{X}}}$ von $\mathcal{F}_{\mathcal{X}}$ sicher realisiert (im Canetti-Modell). □

Der Beweis findet sich in Abschnitt A.6.2, Seite 87. Er dürfte auch auf die meisten anderen vergleichenden Sicherheitsmodelle übertragbar sein, eventuell muß Definition 6.6 leicht angepaßt werden (z. B. f als exponentielle Funktion gefordert werden o. ä.).

Wenn das Kommunikationsmodell so erweitert wird, daß man den Adversary zwingen kann, die Quelle festzulegen (d. h. die Nachricht $(source, j, i)$ zu senden), bevor er mit der Umgebung oder irgendeiner Partei kommuniziert hat, greift das Beispiel aus Abschnitt A.6.1, Seite 86 nicht mehr. Es ist uns unbekannt, ob in diesem Fall in Satz 6.7 die Bedingung der Simulierbarkeit der Quellenfamilie fallengelassen werden kann.

Kapitel 7

Statistische Tests

In der Praxis ist es meist nicht möglich, bestimmte Eigenschaften wie die Qualität des Zufalls einer Quelle zu beweisen. In diesem Fall ist es notwendig, mittels statistischer Tests zumindest starke Indizien für das Zutreffen der gewünschten Eigenschaften zu erlangen.

Bei der Untersuchung eines Tests gehen wir von der folgenden Situation aus: Es liegt ein parametrisiertes Wahrscheinlichkeitsmaß P_ϑ ($\vartheta \in \Theta$) vor, sowie eine Aufteilung $\Theta = \Theta_0 \cup \Theta_1$ in disjunkte Mengen. Ist eine P_ϑ-verteilte Zufallsvariable X (mit unbekanntem ϑ) gegeben, so interessiert uns die Frage, welche der beiden folgenden Aussagen zutrifft:

- *Hypothese:* Es ist $\vartheta \in \Theta_0$.
- *Alternative:* Es ist $\vartheta \in \Theta_1$.

Ein *statistischer Test* besteht nun einfach aus eine Menge $\mathcal{K}$, dem *kritischen Bereich,* und wir verfahren wie folgt:

- Ist $X \notin \mathcal{K}$ (liegt die *Stichprobe* nicht in $\mathcal{K}$), so entscheiden wir uns für die Hypothese,
- ist $X \in \mathcal{K}$, so entscheiden wir uns für die Alternative.

Natürlich wird i. a. kein Test immer die korrekte Aussage treffen, es können die beiden folgenden Fehler auftreten:

- *Fehler erster Art:* Die Hypothese trifft zu, aber es wird sich für die Alternative entschieden.
- *Fehler zweiter Art:* Die Alternative trifft zu, aber es wird sich für die Hypothese entschieden.

Zumeist gibt man einen maximalen Fehler erster Art vor und sucht dann einen dazu passenden Test. Ist der Fehler erster Art garantiert kleiner-gleich α, so spricht man von einem Test *zum Niveau* α.

Oftmals gibt man keinen kritischen Bereich an, sondern eine *Testfunktion* f_T mit Werten in $\mathbb{R}$, so daß $f_T(X)$ bei zutreffender Hypothese eine bekannte Verteilung hat (z. B. Standardnormalverteilung). Dann kann man daraus leicht zu beliebigem Niveau einen kritischen Bereich konstruieren.

Ein mit einem Parameter N (z. B. der Stichprobenlänge) parametrisierter Test heißt *konsistent*, wenn für wachsendes N die maximale Wahrscheinlichkeit eines Fehlers zweiter Art gegen 0 geht.

7.1 Tests für Zufälligkeit

Nun wollen wir kurz einige einfache und verbreitete Tests der Zufälligkeit von binären Quellen vorstellen. Die Hypothese ist also immer „X ist gleichverteilt auf $\{0,1\}^{\mathbb{N}}$“.

Alle diese Tests sind mindestens mit der Länge der Stichprobe parametrisiert und verlangen dann auch eine Stichprobe mindestens dieser Länge. Wir werden im folgenden die Tatsache vernachlässigen, daß Quellen nach unserer Definition 2.1 die Länge der Stichprobe beschränken können. In der Praxis wird man dann so vorgehen, daß man, wenn die Quelle keine hinreichend große Stichprobe liefert, entweder die Hypothese als abgelehnt ansieht (vorsichtige Vorgehensweise), oder den Testparameter auf die erhaltene Stichprobenlänge setzt (heuristisch-mutige Vorgehensweise). Um formal aussagekräftige Argumente zu erhalten, müßte diese Vorgehensweise allerdings im Beweis berücksichtigt werden.

In vielen Fällen ist eine Verallgemeinerung auf nicht binäre Quellen ohne weiteres möglich und hier nur der Einfachheit halber unterlassen worden.

Einen kurzen Überblick über Tests der Zufälligkeit bietet auch [Mau92].

7.1.1 Häufigkeitstest

Der einfachste Test ist der *Häufigkeitstest* (*frequency test*). Dieser zählt einfach die relative Häufigkeit der 1. Bei einer zufälligen Quelle müßte diese ungefähr $\frac{1}{2}$ betragen.

Bei einer Stichprobenlänge N ist die Testfunktion

$$f_T(x) := \frac{2}{\sqrt{N}}\Big(\sum_{i=1}^{N} x_i - \frac{N}{2}\Big) \qquad (x \in \{0,1\}^N),$$

und $f_T(X)$ hat bei zutreffender Hypothese für große N approximativ Standardnormalverteilung (gemäß dem Zentralen Grenzwertsatz).

Läßt die Alternative nur unabhängig identisch verteilte Quellen zu, so ist dieser Test konsistent.

7.1.2 Serientest

Der *Serientest* (*serial test*) ist eine Verallgemeinerung des Häufigkeitstests. Hier werden Blöcke einer Länge L untersucht und die relativen Häufigkeiten aller Sequenzen dieser Länge berechnet. Für zufällige Quellen ist zu erwarten, daß jede Sequenz die relative Häufigkeit 2^{-L} hat.

Die Stichprobenlänge sei wieder N, sowie $f_i(x)$ für $x \in \{0,1\}^N$ und $i \in \{0,\dots,2^L-1\}$ die Anzahl der Blöcke von x mit binärer Repräsentation i. Dann ist die Testfunktion definiert durch

$$f_T(x) := \frac{L2^L}{N}\sum_{i=0}^{2^L-1}\Big(f_i(x) - \frac{N}{L2^L}\Big)^2 \qquad (x \in \{0,1\}^N).$$

Es hat für $N \gg L$ bei zutreffender Hypothese $f_T(X)$ approximativ Chi-Quadrat-Verteilung mit $2^L - 1$ Freiheitsgraden.

7.1.3 Lauflängentest

Der *Lauflängentest* (*run test*) zählt die Anzahl der Läufe (konstante Teilsequenzen) der verschiedenen Längen. Es sei $r_i^\sigma(x)$ für $\sigma \in \{0,1\}$ und $i \in \mathbb{N}$ die Anzahl der σ-Läufe der Länge i in x.

Es sei weiter N die Stichprobenlänge und L ein Parameter, welcher die maximale zu berücksichtigende Lauflänge angibt. Dann ist die Testfunktion:

$$f_T(x) \;:=\; \sum_{\sigma\in\{0,1\}}\sum_{i=1}^{L}\frac{\big(r_i^\sigma(x) - \frac{N}{2^{i+2}}\big)^2}{\frac{N}{2^{i+2}}} \qquad (x \in \{0,1\}^N),$$

und $f_T(X)$ ist für große N approximativ χ^2-verteilt mit $2L$ Freiheitsgraden.

7.1.4 Autokorrelationstest

Der *Autokorrelationstest* (*autocorrelation test*) hat als Parameter die Verzögerung $\tau \in \mathbb{N}$ und die Stichprobenlänge N. Dieser Test basiert darauf, daß die Zufallsfolge $Y^{(\tau)} := (X_i \oplus X_{i+\tau})_i$ für zufälliges X auch zufällig ist. Der Autokorrelationstest für X ist dann genau der Häufigkeitstest für $Y^{(\tau)}$.

7.1.5 Maurers Universaltest

Ein weiterer Test, der laut [Mau92] alle obigen Tests einschließt, ist *Maurers Universaltest* (*Maurer's universal test*). Wir verweisen auf [Mau92] für eine Beschreibung dieses Tests, hier seien nur die Parameter aufgezählt:

- Die Stichprobenlänge N, wie bei den anderen Tests auch,
- eine Blocklänge L wie beim Serientest,
- eine Präfixlänge Q, welche angibt, wieviel der Stichprobe für eine Vorverarbeitung verwendet werden soll.

7.2 Test der Symbolgewichtung

Liegt nun eine Quelle vor, für die wir ein Modell aufgestellt haben, und aus der wir mittels adaptiver Extraktion (Satz 4.8) Zufall extrahieren wollen, so gilt es, dieses Modell zu testen, oder zumindest die Behauptung, daß unser Extraktionsverfahren aus dieser Quelle guten Zufall extrahiert.

Man ist nun versucht, wie folgt vorzugehen:

- Anhand der Modellierung konstruieren wir einen Extraktor.
- Mit dem Extraktor generieren wir aus der Quelle eine (hoffentlich zufällige) Symbolfolge.
- Dann wenden wir oben genannte (und evtl. andere) Tests der Zufälligkeit auf das Resultat an.

Dieses Vorgehen hat drei Nachteile:

- Zum einen benötigen wir für hohe Qualität des Zufalls einen exponentiell oder zumindest superpolymiell geringen statistischen Abstand zur Gleichverteilung, aber die Definition des statistischen Abstands sagt ja gerade aus, daß die Wahrscheinlichkeit eine perfekte von einer schlechten Zufallsfolge zu unterscheiden proportional zu diesem Abstand ist. Somit werden wir bei polynomiell vielen Testläufen alle Quellen akzeptieren, deren stochastischer Abstand zur perfekten Zufälligkeit mit einem Polynom höheren Grades wächst.
- Die Quelle kann – egal wie gut der Test ist – immer noch versteckte Eigenschaften haben, die die Zufälligkeit zerstören, aber von unserem Test nicht bemerkt werden. Hier können wir drei Typen unterscheiden:
  - Eigenschaften, die zwar in den Testsequenzen bereits auftreten, von unseren Tests aber nicht bemerkt werden. Dieses Risiko kann reduziert (aber nicht ausgeschlossen) werden durch die Anwendung immer komplexerer Tests,[20] welche aber immer mehr Testdaten benötigen.
  - Eigenschaften, die erst in der Zukunft auftreten. Diese könnten z. B. von äußeren Umwelteinwirkungen abhängen (z. B. Änderung des Luftdrucks, der Gravitation etc.) oder einfach von inneren Verschleißerscheinungen. Dieses Risiko kann man versuchen zu verringern, indem man einerseits versucht, die Quelle gut abzuschirmen, und andererseits die Quelle mit internen Überprüfungseinrichtungen versieht, die testen sollen, ob die Quelle noch so funktioniert wie zu Beginn (zur Testzeit).
  - Eigenschaften, die von einer gegnerischen Partei beeinflußt werden können (z. B. wegen der Empfindlichkeit gegenüber bestimmten Magnetfeldern). Diese stellen deshalb ein besonders hohes Risiko dar, da diese Partei die Eigenschaften erst zutage treten lassen wird, wenn sie sie benötigt, aber sicherlich nicht während des Testlaufs. Auch hier ist eine Abschirmung ein Weg zur Verringerung des Risikos.
- Den dritten Nachteil erläutere das folgende Experiment:
  - Man nehme eine schlechte Zufallsfolge (beispielsweise eine unabhängig identisch verteilte binäre Folge mit $P(X_0 = 1) = 0{,}51$).
  - Man wähle eine zufällige quadratische Toeplitz-Matrix, welche auf der Diagonale konstant 1 und unter der Diagonale konstant 0 ist (z. B. der Größe 1001×1001).
  - Dann wende man auf die Zufallsfolge blockweise die Matrix an und teste das Ergebnis mit den verschiedenen Zufallstests.

  Wir haben dieses Experiment mit obigen Beispielwerten durchgeführt und sind zu folgenden Ergebnissen gekommen:

Test	Stichprobenlänge N	Sonst. Parameter	akzeptiert für Niveaus
Häufigkeitstest	10^9		0,75 , 0,76 , 0,99
Autokorrelationstest	10^9	$\tau = 1$	0,70 , 0,68 , 0,96
Serientest	10^9	$L = 16$	0,34 , 0,25 , 0,32
Lauflängentest	10^9	$L = 15$	0,59 , 0,03 , 0,58
Maurers Universaltest	10^9	$L = 16$, $Q = 655360$	0,21 , 0,56 , 0,62

  Dabei wurde jeder Test dreimal durchgeführt, angegeben sind immer die kleinstmöglichen Niveaus, für die die jeweiligen Tests noch akzeptiert hätten, gerundet auf zwei Nachkommastellen.

  Die Ursprungsfolge hingegen wird sogar bei einer Stichprobenlänge von nur 10^7 und einem Niveau von 10^{-6} vom Häufigkeitstest, vom Serientest und vom Lauflängentest nur sehr selten akzeptiert.[21]

[20] Ein Test, welcher zumindest im Grenzfall alle Eigenschaften erkennt, ist der Kolmogorov-Test, welcher zu einer Folge unter den diese Folge generierenden Turingmaschinen die mit kleinsten Gödelnummer wählt. Der Test akzeptiert dann, wenn Logarithmus der Gödelnummer nicht wesentlich kleiner als die Länge der Folge ist. Leider ist dieser Test nicht berechenbar.

[21] Beim Autokorrelationstest und bei Maurers Universaltest sind größere Stichproben nötig.

Da unsere Nachbearbeitung die Anwendung einer Bijektion darstellt, muß die erhaltene Folge genauso schlecht sein wie die Ursprungsfolge. Jedoch merken dies die Tests nicht. Deshalb dürfen wir erwarten, daß auch bei falsch gewählter Modellierung nach Anwendung des adaptiven Hash-Extraktors die Tests die resultierende Folge für gut befinden, denn ein Teil der Nachbearbeitung besteht aus dem Anwenden von Toeplitz-Matrizen, welche oft auch noch größer sind als die oben benutzte 1001×1001-Matrix. Also ist ein Test *nach* Anwendung des Extraktors nicht sinnvoll.

Die ersten beiden Punkte sind prinzipieller Natur, dem dritten wollen wir in diesem Kapitel beikommen, indem wir einen Test vorstellen, der eine Quelle nicht auf Zufälligkeit testet (denn unsere unbearbeitete Quelle ist nicht zufällig), sondern darauf, ob eine gegebene Symbolgewichtung ihr Verhalten korrekt beschreibt.

Definition 7.1: Gewichtungstest

Es sei Σ nichtleer und endlich, $F > 0$, $\pi, \varrho \in \Sigma^*$, $|\varrho| > 0$, $\varepsilon \in \mathbb{R}_{>0}$, $N \in \mathbb{N}$, $M \in \mathbb{N}_0$, $L \in \mathbb{N}$, $L \geq |\pi\varrho|$. Dann sei $b_i(x)$ für $x \in \Sigma^N$ der i-te Block der Länge L in x, also

$$b_i(x) := x_{(i-1)L+1} \dots x_{iL}$$

und $n_\omega(x)$ für $\omega \in \Sigma^L$ die Anzahl der $b_i(x)$ mit $b_i(x) = \omega$, also

$$n_\omega(x) := \sum_{i=1}^{\lfloor N/L \rfloor} \delta(b_i(x) = \omega).$$

Weiter seien für $\varphi \in \Sigma^{L^*}$, $L^* := L - |\pi\varrho|$

$$\begin{aligned}
\hat{n}_\varphi(x) &:= \sum_{\tilde{\varrho} \in \Sigma^{|\varrho|}} n_{\varphi\pi\tilde{\varrho}}(x), \\
\hat{n}(x) &:= \sum_{\tilde{\varphi} \in \Sigma^{L^*}} n_{\tilde{\varphi}}(x), \\
f_\varphi(x) &:= \frac{n_{\varphi\pi\varrho}(x) - 2^{-\varepsilon}\hat{n}_\varphi(x)}{\sqrt{(2^{-\varepsilon} - 2^{-2\varepsilon})\hat{n}_\varphi(x)}}
\end{aligned}$$

mit $\frac{0}{0} := 0$.

Dann ist die Testfunktion definiert durch

$$f_T(x) \;:= \sum_{\varphi \in \Sigma^{L^*}} \max\{0, f_\varphi(x)\}^2$$

und der kritische Bereich $\mathcal{K}$ des *Gewichtungstests für $\eta(\dots\pi;\varrho) \geq \varepsilon$ mit Schranke F, Stichprobengröße M und Blocklänge L* durch

$$\mathcal{K} = \{x \in \Sigma^L : \; f_T(x) \geq F \text{ oder } \hat{n}(x) < M\}.$$

Weiterhin ist der kritische Bereich des *Gewichtungstests für $\eta(\dots\pi;\varrho) = \infty$ mit Stichprobengröße M und Blocklänge L*

$$\{x \in \Sigma^L : \; \exists\varphi \in \Sigma^{L^*}\colon n_{\varphi\pi\varrho}(x) \neq 0 \text{ oder } \hat{n}(x) < M\}.$$ □

Dieser Gewichtungstest hat, gegeben eine Quelle X, das Ziel, die Hypothese „$\eta^{\{X\}}(\xi\pi;\varrho) \geq \varepsilon$ für alle $\xi \in \Sigma^*$“ zu testen. Hierzu wird die Stichprobe x in Blöcke $b_i(x)$ der Länge L zerlegt. Aus diesen Blöcken wird eine Statistik erstellt, mit welcher absoluten Häufigkeit $n_{\varphi\pi\varrho}(x)$ nach $\varphi\pi$ die Symbolfolge ϱ folgt. Stimmt die Hypothese, so läßt sich $n_{\varphi\pi\varrho}(X)$ als Summe von $\hat{n}_\varphi(X)$ $\mathrm{Bin}(p)$-verteilten Zufallsvariablen auffassen, mit $p \leq 2^{-\varepsilon}$. Wir können $n_{\varphi\pi\varrho}(X)$ nach oben abschätzen,[22] wenn wir $p = 2^{-\varepsilon}$ annehmen. Damit wird $f_\varphi(X)$ durch eine approximativ standardnormalverteilte Zufallsvariable nach oben abgeschätzt. Also ist (mit N_φ als unabhängigen, standardnormalverteilten Zufallsvariablen)

$$P(X \in \mathcal{K}) = P(f_T(X) \geq F) \leq P\Big(\sum_\varphi \max\{0, N_\varphi\}^2 \geq F\Big) =: \alpha_F,$$

[22] Wir sagen, die Zufallsvariable X schätzt die Zufallsvariable Y nach oben ab, wenn für alle $t \in \mathbb{R}$ gilt: $P(X \geq t) \geq P(Y \geq t)$.

wenn wir zunächst einmal $M = 0$ annehmen. Es liegt also dann ein Test zum Niveau α_F vor.[23]

Die Zusatzbedingung $\hat{n}(X) \geq M$ führt zwar dazu, daß der Test für $M > 0$ nicht mehr das Niveau α hat, jedoch kann es ohne diese Bedingung passieren, daß der Test die Hypothese akzeptiert, wenn gar keine oder nur sehr wenige Blöcke der Form $\varphi\pi\tilde{\varrho}$ gefunden wurden, also die Hypothese eigentlich gar nicht geprüft wurde.

Unser Ergebnis wird von der folgenden heuristischen Aussage exakter dargestellt und zusammengefaßt:

Heuristik 7.2: Niveau des Gewichtungstests

Sei X eine Quelle über Σ, $\alpha \in [0,1]$, $\pi, \varrho \in \Sigma^*$, $|\varrho| > 0$, $\varepsilon \in \mathbb{R}_{>0} \cup \{\infty\}$, $N \in \mathbb{N}$, $M \in \mathbb{N}_0$, $L \in \mathbb{N}$, $L \geq |\pi\varrho|$, $L^* := L - |\pi\varrho|$.

Es sei $F \in \mathbb{R}_{>0}$ mit

$$2^{-\#\Sigma^{L^*}} \sum_{i=1}^{\#\Sigma^{L^*}} \binom{\#\Sigma^{L^*}}{i} (1 - \chi_i^2(F)) \leq \alpha, \tag{13}$$

wobei χ_i^2 die Verteilungsfunktion der Chi-Quadrat-Verteilung mit i Freiheitsgraden sei.

Es bezeichne $\mathcal{K}$ den kritischen Bereich des Gewichtungstests für $\eta(\ldots\pi;\varrho) \leq \varepsilon$ mit Schranke F, Stichprobengröße M und Blocklänge L, und $\hat{n}$ sei wie in Definition 7.1.

Ist $\eta^{\{X\}}(\xi\pi;\varrho) \geq \varepsilon$ für alle $\xi \in \Sigma^*$, so gilt für große M approximativ

$$P\big(X \in \mathcal{K} \text{ und } \hat{n}(X) \geq M\big) \leq \alpha. \qquad \square$$

Eine ausführliche Beweisskizze (aber kein formaler Beweis) findet sich in Abschnitt A.7.1, Seite 92.

Die Einschränkung $\hat{n}(X) \geq M$ mag das Ergebnis theoretisch weniger schön machen, in der Praxis ist sie aber nicht sehr hinderlich, wie folgende Überlegungen zeigen sollen:

Der Test wird i. a. angewandt werden, um die Gültigkeit von postulierten unteren Schranken für die zu einer Quelle gehörige Symbolgewichtung zu prüfen. Lehnt der Test eine dieser Schranken (z. B. $\eta(\ldots\pi;\varrho) \geq \varepsilon$) ab, so verwenden wir z. B. bei der Konstruktion eines Extraktors die direkt aus der Definition folgende untere Schranke $\eta(\ldots\pi;\varrho) \geq 0$. Wurde die ursprüngliche Schranke abgelehnt, obwohl sie korrekt ist, und zwar nur wegen $\hat{n}(X) < M$, so hat der Extraktor mit Schranke $\eta(\ldots\pi;\varrho) \geq 0$ fast die gleich Rate wie der mit $\eta(\ldots\pi;\varrho) \geq \varepsilon$, denn da die Quelle die Symbolfolge π sowieso fast nie ausgibt, fließt diese Schranke in die Extraktion kaum ein.[24]

Eine zu einem Niveau α passende Schranke F läßt sich durch binäre Suche mit vertretbarem Aufwand beliebig nah am Optimum finden, sofern $\#\Sigma^{L^*}$ nicht allzu groß wird,[25] da die linke Seite von (13) streng monoton in F fällt.

[23] Die quadrierten normalverteilten Zufallsvariablen deuten an, daß es sich hier um eine Variante des Chi-Quadrat-Tests handelt.

[24] Es sei denn, es tritt der unwahrscheinliche Fall ein, daß die Quelle gerade so arbeitet, daß π zwar oft vorkommt, aber selten so, daß es an der Position im Block liegt, an der es von unserem Gewichtungstest erwartet wird (denn dieser prüft nur $b_i(X) = \tilde{\varphi}\pi\tilde{\varrho}$ für alle $\tilde{\varphi}$, $\tilde{\varrho}$ von *fester* Länge.)

[25] Ist $\#\Sigma^{L^*}$ sehr groß, so wird schon das Berechnen von f_T sehr aufwendig.

Kapitel 8

Extraktion in der Praxis

8.1 Software

Im Rahmen dieser Arbeit ist ein kleines Programm entstanden, welches es ermöglicht, die vorgestellten Ergebnisse auszuprobieren. Allerdings sollte dieses Programm nicht dazu verwendet werden, um in kritischen Anwendungen Zufall zu erzeugen; hierzu wird empfohlen, ein minimales, aber sehr gut überprüftes Programm zu schreiben.

Die Quellen zu diesem Programm finden sich auf der beigefügten CD und unter

`http://www.unruh.de/DniQ/randomextraction/`

Um das Programm zu kompilieren, werden die folgenden Bibliotheken benötigt:

- C-XSC 2.0. Es handelt sich hierbei um eine Klassenbibliothek für Intervallarithmetik, verfügbar unter `http://www.math.uni-wuppertal.de/~xsc/xsc/cxsc.html`.
- Qhull. Dies ist eine Bibliothek zum Rechnen mit konvexen Hüllen, verfügbar unter `http://www.geom.uiuc.edu/software/qhull/`.

Das Programm wurde unter Linux entwickelt und läuft möglicherweise nur darunter.

In kompilierter Form besteht die Software aus zwei Komponenten:

- `randomextract`. Dieses Programm hat die folgende Aufrufsyntax:

  `randomextract [datei]`

  Hierbei ist `datei` eine Textdatei (Format siehe Anhang B), die angibt, welche Aufgaben das Programm erledigen soll.
- `RandomExtraction`. Dieses Programm stellt eine graphische Oberfläche zur Verfügung, um `randomextract` komfortabler zu verwenden. Als zusätzliche Hilfe erstellt `RandomExtraction` bei jedem Aufruf von `randomextract` eine Datei namens `gui.spec`, welche die zuletzt an `randomextract` geleitete Eingabe enthält.

8.2 Die Münchner Quelle

Im folgenden wollen wir die in dieser Arbeit entwickelten Methoden auf eine real existierende Quelle anwenden. Es handelt sich hierbei um eine am Institut für Physik der Ludwig-Maximilians-Universtität München von Prof. Harald Weinfurter und Dr. Christian Kurtsiefer entwickelte Quelle mit einer Ausgaberate von bis zu 20 Mbit/s, im folgenden die Münchner Quelle genannt.

8.2.1 Versuchsaufbau

Die hier präsentierten Details über die Münchner Quelle entstammen [Haa02].

Die Münchner Quelle verwendet folgenden Versuchsaufbau:

- Eine Leuchtdiode emittiert Lichtblitze mit einer bekannten durchschnittlichen Frequenz f_E.
- Diese passieren eine getönte Scheibe mit einer unbekannten Absorbtionsrate θ.
- Nicht absorbierte Photonen treffen auf einen Photodetektor.
- Jeder vom Photodetektor gemessene Lichtblitz triggert ein angeschlossenes Flipflop, welches darauf seinen Zustand wechselt.
- Mit einer von der Emissionsfrequenz unabhängigen, bekannten Lesefrequenz f_L wird der Zustand des Flipflops ausgelesen.
- Die ausgelesenen Daten bilden die Zufallsfolge.

Offenbar liefert die Quelle bei idealer Hardware für $\frac{f_L}{f_E} \to \infty$ perfekten Zufall. In praxi jedoch treten die folgenden Probleme auf:

- Die Lesefrequenz f_L liegt aus Effizienzgründen nah an der Emissionsfrequenz f_E.
- Die Absorbtionsrate θ ist nicht bekannt.
- Der Detektor kann zeitweilig ausfallen, z. B. weil zu viele Lichtblitze in Folge aufgetreten sind (Blendung).
- Das Flipflop kann imperfekt sein in der Hinsicht, daß das Signal zum Umschalten ignoriert wird (evtl. abhängig vom aktuellen Zustand), oder daß beim Auslesen der falsche Wert übertragen wird.

8.2.2 Modellierung als CHMM

Wir wollen nun die Münchner Quelle mit den zuvor beschriebenen Problemen als CHMM modellieren. Dazu betrachten wir den Zeitraum zwischen zweimaligem Auslesen des Flipflops. In dieser Zeit treten N Lichtblitze auf, wobei N $\mathrm{Po}(f_E/f_L)$-verteilt ist (Poisson-verteilt mit Parameter f_E/f_L). Jeder Blitz wird mit einer Wahrscheinlichkeit von θ absorbiert. Die Anzahl der beim Detektor eintreffenden Lichtblitze ist also $\mathrm{Po}(\theta f_E/f_L)$-verteilt.

Der Detektor kann mit einer Wahrscheinlichkeit $p_B \in B$ während des betrachteten Zeitraums geblendet werden oder sein, in diesem Fall nehmen wir an, daß der Detektor beliebige Ausgaben produzieren kann (sprich selbst entscheiden, ob das Flipflop nach dem betrachteten Zeitraum im gleichen oder in einem neuen Zustand ist).

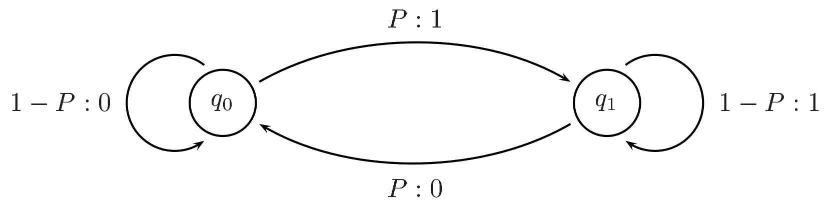
Weiterhin kann das Flipflop mit einer Wahrscheinlichkeit $p_I \in I$ während des betrachteten Zeitraums ein- oder mehrmals nicht umschalten. Wie schon beim Detektor nehmen wir dann an, daß der neue Zustand des Flipflops dann beliebig sein kann.

Es ergibt sich eine Wahrscheinlichkeit p für den Wechsel des Zustands des Flipflops (gegeben alles vor diesem Zeitraum geschehene), die innerhalb der folgenden Menge liegt.

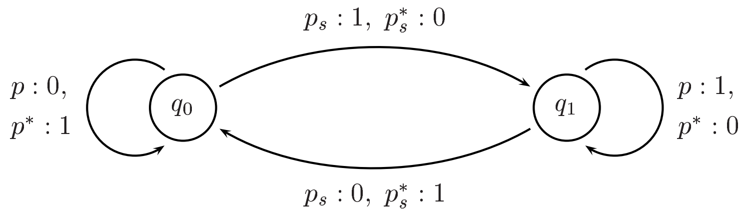
$$\left\{ \bar{p}_I \bar{p}_B p + (1 - \bar{p}_I \bar{p}_B)t \ : \ \bar{p}_I \in 1 - I,\ \bar{p}_B \in 1 - B,\ t \in [0,1] \right\} =: P,$$

wobei p die Wahrscheinlichkeit dafür sei, daß eine $\mathrm{Po}(\theta f_E/f_L)$-verteilte Zufallsvariable einen ungeradzahligen Wert annimmt (die ideale Umschaltwahrscheinlichkeit).

Wir modellieren die Zustandsübergänge also wie folgt als CHMM, wobei wir zunächst noch annehmen, die Ausgabe des Flipflops entspräche exakt seinem Zustand:[26]



Nun müssen wir noch modellieren, daß das Flipflop falsch ausgelesen werden kann. Wir nehmen an, daß die Ausgabe des Flipflops mit einer Wahrscheinlichkeit von maximal ε_F fehlerhaft gelesen wird, dann erhalten wir:[27]



$$p + p^* \in 1 - P, \qquad p_s + p_s^* \in P, \qquad p^* \leq \varepsilon_F(p + p^*), \qquad p_s^* \leq \varepsilon_F(p_s + p_s^*)$$

Das Problem bei obiger Analyse ist, daß die Werte von P und ε_F schwierig zu bestimmen sind; auch sollte die gesamte obige Analyse von einem Experten verifiziert werden. Wir werden deshalb im nächsten Abschnitt einen anderen Weg gehen.

[26] Für den (willkürlich gewählten) Beispielsfall $P = [0{,}4, 0{,}55]$ findet sich dieses CHMM in der Datei `muenchen1.chmm`, siehe Abschnitt B.4.

[27] Für den (willkürlich gewählten) Beispielsfall $P = [0{,}4, 0{,}55]$, $\varepsilon_F = 0{,}03$ findet sich dieses CHMM in der Datei `muenchen2.chmm`, siehe Abschnitt B.4.

8.2.3 Schätzung der Symbolgewichtung

Mit Hilfe des in Abschnitt 7.2 vorgestellten Tests haben wir eine Schätzung für die Symbolgewichtung der Münchner Quelle aufgestellt. Dabei sind wir wie folgt vorgegangen:

- Für jede Wahl der folgenden Parameter

$$\begin{aligned}(f_L, f_E) \in \{\, &(10\,\text{MHz}, 12{,}5\,\text{MHz}),\ (10\,\text{MHz}, 25\,\text{MHz}),\ (10\,\text{MHz}, 125\,\text{MHz}),\ (10\,\text{MHz}, 200\,\text{MHz}),\\ &(20\,\text{MHz}, 12{,}5\,\text{MHz}),\ (20\,\text{MHz}, 25\,\text{MHz}),\ (20\,\text{MHz}, 50\,\text{MHz}),\ (20\,\text{MHz}, 200\,\text{MHz})\,\},\end{aligned}$$

$$\alpha := 0{,}999, \qquad M := 10^6, \qquad L := 8 + i, \qquad i \in \{0, \ldots, 4\}, \qquad \pi \in \{0,1\}^i, \qquad \varrho \in \{0,1\}$$

  suchen wir jeweils das größte $\varepsilon = \varepsilon^{\pi,\varrho}_{f_L,f_E}$, so daß die mit Lesefrequenz f_L und Emissionsfrequenz f_E erzeugte Stichprobe x der Länge 128 MB (entspricht ca. 10^9 Symbolen) den Gewichtungstest für $\eta(\ldots\pi; \varrho) \geq \varepsilon$ mit Stichprobengröße M und Blocklänge L zum Niveau α besteht.

  Man beachte, daß je näher α an 1 liegt, die Schätzung von ε desto niedriger, sprich sicherer wird. Daher die ungewöhnlich hohe Wahl von α.

- Für alle im vorangegangenen Schritt zugelassenen f_L, f_E und i definieren wir folgende Schätzung der Symbolgewichtung:

$$\eta^{(i)}_{f_L,f_E}(\xi\pi; \varrho) := \varepsilon^{\pi,\varrho}_{f_L,f_E} \qquad\qquad (\pi \in \{0,1\}^i,\ \varrho \in \{0,1\},\ \xi \in \{0,1\}^*).$$

  Im Fall $|\beta| > 1$ schätzen wir $\eta(\alpha; \beta)$ mittels Lemma 4.2 ab.

  Die Ergebnisse der Schätzungen liegen unter den Dateinamen `muenchen-*.weight` in einem für `random-extract` verständlichen Format vor (Nichtterminal ⟨*weighting*⟩, siehe Abschnitt B.2).

- Wir approximieren die Rate der vorliegenden Quelle mit der Symbolgewichtung $\eta^{(i)}_{f_L,f_E}$ durch

$$R^{(i)}_{f_L,f_E} := \frac{\eta^{(i)}_{f_L,f_E}(\lambda; x)}{N} = \frac{1}{N}\sum_{\nu=1}^{N} \eta^{(i)}_{f_L,f_E}(x_1 \ldots x_{\nu-1};\ x_\nu)$$

  (vergleiche hierzu die Bemerkungen auf Seite 25).

  Es ergeben sich die folgenden Werte (gerundet auf zwei Nachkommastellen):

f_L	f_E	$R^{(0)}_{f_L,f_E}$	$R^{(1)}_{f_L,f_E}$	$R^{(2)}_{f_L,f_E}$	$R^{(3)}_{f_L,f_E}$	$R^{(4)}_{f_L,f_E}$
10 MHz	12,5 MHz	0,69	0,95	0,95	0,95	0,95
10 MHz	25 MHz	0,88	0,99	0,99	0,99	0,99
10 MHz	125 MHz	0,99	1,00	1,00	1,00	1,00
10 MHz	200 MHz	0,99	1,00	0,99	1,00	0,99
20 MHz	12,5 MHz	0,41	0,78	0,80	0,80	0,79
20 MHz	25 MHz	0,61	0,89	0,93	0,93	0,93
20 MHz	50 MHz	0,79	0,95	0,98	0,98	0,98
20 MHz	200 MHz	0,99	0,99	1,00	1,00	0,99

Die in obiger Tabelle angegeben Raten sind die asymptotischen Raten, die wir nur bei gegen unendlich strebender Blocklänge erreichen können. Nehmen wir jetzt eine Lebensdauer der Quelle von maximal 1000 Jahren an, so wissen wir, daß die Quelle nicht mehr als $l = 10^{60}$ Symbole ausgeben wird. Verlangen wir weiterhin, daß das Ergebnis der Extraktion $\varepsilon = 10^{-80}$-zufällig sein soll, so ergibt sich, wieder gemäß der Überlegungen auf Seite 25, für die tatsächliche Rate bei Blocklänge n:

$$R^{(i,n)}_{f_L,f_E} \approx R^{(i)}_{f_L,f_E} - \frac{\tilde{c}}{n}$$

mit $\tilde{c} := -2\log\varepsilon + 4\log l = 400$. Um die asymptotische Rate zu 90 % auszunutzen, müssen wir dann als Blocklänge $n := 10\tilde{c}/R$ verwenden. Für die verschiedenen Parameter haben wir die vorgeschlagene Blocklänge und die zugehörige tatsächliche Rate in folgender Tabelle zusammengestellt:

f_L	f_E	n	$R^{(0,n)}_{f_L,f_E}$	n	$R^{(1,n)}_{f_L,f_E}$	n	$R^{(2,n)}_{f_L,f_E}$	n	$R^{(3,n)}_{f_L,f_E}$	n	$R^{(4,n)}_{f_L,f_E}$
10 MHz	12,5 MHz	5837	0,62	4192	0,86	4191	0,86	4193	0,86	4202	0,86
10 MHz	25 MHz	4542	0,80	4029	0,90	4030	0,90	4030	0,90	4038	0,89
10 MHz	125 MHz	4024	0,90	4007	0,90	4013	0,90	4013	0,90	4018	0,90
10 MHz	200 MHz	4039	0,89	4019	0,90	4030	0,90	4019	0,90	4031	0,90
20 MHz	12,5 MHz	9818	0,37	5141	0,70	5007	0,72	4982	0,73	5078	0,71
20 MHz	25 MHz	6600	0,55	4489	0,80	4287	0,84	4288	0,84	4290	0,84
20 MHz	50 MHz	5036	0,72	4203	0,86	4068	0,89	4082	0,89	4082	0,88
20 MHz	200 MHz	4040	0,89	4023	0,90	4012	0,90	4018	0,90	4021	0,90

Wir können aus diesen Daten nun die folgenden Schlüsse ziehen:

- Für eine gute untere Abschätzung der Symbolgewichtung genügt es, das letztausgegebene Bit zu betrachten, ein Präfix größerer Länge bringt keine nennenswerte Verbesserung mehr (im besten Fall vier Prozentpunkte).[28] Dies macht es zum einen sehr einfach, die Symbolgewichtung auch für große Datenmengen schnell zu berechnen, und zum anderen deutet es darauf hin, daß die Münchner Quelle ein Gedächtnis von einem Symbol hat.

- Da sich die Extraktionsrate für steigendes Verhältnis f_E/f_L nur unterproportional erhöht, erkennen wir, daß es sinnvoll ist, für festgelegte Extraktionsrate die Auslesefrequenz so hoch zu wählen, wie die Hardware es zuläßt.[29]

- Ein interessantes Phänomen ist die Tatsache, daß bei $f_L = 20\,\text{MHz}$, $f_E = 12,5\,\text{MHz}$ die asymptotische Extraktionsrate bei etwa 80 % liegt. Damit ergibt sich eine resultierende Datenrate von bis zu 16 Mbit/s, wir erhalten somit mehr als ein Bit pro emittiertem Lichtblitz. Damit erkennen wir, daß der Zufall, den die Münchner Quelle erzeugt, nicht allein aus der zufälligen Absorption durch die getönte Scheibe entsteht (dies könnte höchstens ein Bit pro Lichtblitz erklären), sondern auch mit der ungleichmäßigen Emission durch die Leuchtdiode.

  Diese These wird dadurch noch gestärkt, daß wir die Anzahl der Lichtblitze pro Zeitintervall τ als $\mathrm{Po}(\tau f_E)$-verteilt modellieren können, dann ist die Anzahl der die Scheibe passierenden Blitze $\mathrm{Po}(\theta\tau f_E)$-verteilt. Dies ist aber auch die Verteilung, die wir für eine Leuchtdiode mit Emissionsfrequenz θf_E und ohne getönte Scheibe erwarten würden.

Die hier vorgestellte Analyse ist natürlich mit großer Vorsicht zu genießen und dient nur als Überblick über die zu erwartenden Parameter des Extraktionsverfahrens.

Vor einer tatsächlichen Implementierung müssen einerseits die den Schätzungen und Tests zugrundeliegenden Annahmen genauer spezifiziert und begründet werden (welche Bedeutung hat z. B. die Wahl von $\alpha = 0{,}999$ als Niveau), und zweitens wesentlich mehr Tests in verschiedenen Umgebungen gefahren werden.

[28] In einigen Fällen ist sogar eine Verschlechterung von einem Prozentpunkt zu bemerken, dies ist aber damit zu erklären, daß für längere Präfixe die Stichprobe pro Präfix geringer wird und die Schätzung aus diesem Grund vorsichtiger ausfällt.

[29] Wobei nicht zu vernachlässigen ist, daß auch die für die adaptive Extraktion verwendete Hardware der limitierende Faktor sein kann, da die Anwendung der Toeplitz-Matrizen aufwendig ist. (Die asymptotische Laufzeit beträgt zwar $O(n \log n)$ bei Implementation durch schnelle Fouriertransformation, aber für die vorliegenden Blockgrößen ist die Benutzung dieses Algorithmus vermutlich noch nicht lohnend; hier bietet sich eher eine Faltung nach Karatsuba in $O(n^{1,58})$ an.)

Kapitel 9

Schlußbemerkungen

Wir haben in dieser Arbeit gesehen, wie man zu Quellen verschiedenster Art adaptive Extraktoren konstruieren kann. Weitere Forschung könnte z. B. in die folgenden Richtungen gehen:

- Man kann untersuchen, inwiefern die resultierende Folge noch sicher ist, wenn man zur Laufzeit und abhängig von den ausgegebenen Daten die Parameter ändert (wie z. B. die Blocklänge oder die gewünschte Qualität).
- Es ist denkbar, den Extraktor mit anwachsender Blocklänge zu konstruieren. Damit kann man einen beispielsweise quadratisch abfallenden statistischen Abstand pro Block zur Gleichverteilung realisieren, so daß der Gesamtabstand (in etwa die Summe über die Einzelabstände) konvergiert. Dies würde es ermöglichen, adaptive Extraktoren zu gestalten, die keine Beschränkung der Eingabelänge haben.
- Anstelle des Leftover Hash Lemmas sind vielleicht andere blockweise Extraktoren möglich, man interessiert sich dann dafür, wie diese zu integrieren sind, und welche Auswirkungen auf das Gesamtsystem auftreten.
- Nehmen wir an, daß die Quelle ein beschränktes Erinnerungsvermögen hat, so ist es vielleicht möglich, Seitenkanäle gewisser geringer Übertragungsrate zuzulassen. (Die derzeitige Modellierung, bei der der Seitenkanal insgesamt nur eine beschränkte (und sehr kleine) Menge an Daten übertragen darf, liegt darin begründet, daß er sonst zuviel Information über einen einzelnen Block ausliefern könnte. Dies gilt nicht mehr, wenn das Gedächtnis zu kurz ist, um diese Informationen lange genug zu speichern, um sie übertragen zu können.) Es stellt sich vor allem die Frage nach einer guten Modellierung einer solchen Beschränkung des Gedächtnisses.
- Statt initialen Zufalls könnte man auch eine zweite unabhängige Quelle verwenden und untersuchen, wie man das Verfahren dann anpassen muß (evtl. angelehnt an [CG88]).
- Die Sicherheit bei der Verwendung von geschätzten Symbolgewichtungen kann erhöht werden, indem wir zur Laufzeit immer wieder unsere Annahmen über die Quelle prüfen. Eine genauere Formulierung und theoretische Modellierung dieser Idee wäre wünschenswert.
- Auf der praktischen Seite benötigt man schnelle Implementierungen insbesondere der bei der Anwendung der Toeplitz-Matrizen anfallenden Faltungen.
- Und schließlich gilt es, existierende physikalische Quellen zu untersuchen, zu modellieren und zu testen.

Abschließend möchte ich noch denen danken, die mich bei der Erstellung dieser Diplomarbeit tatkräftig unterstützt haben: Dr. Jörn Müller-Quade wegen anregender Diskussionen, Анне Бачинске für Unterstützung daheim, und Manuel Kauers wegen schnellen und vor allem umfangreichen Korrekturlesens. Weiterhin sei auch noch all jenen gedankt, welche hier nicht einzeln genannt wurden, aber dennoch den einen oder anderen kleinen Fehler entdeckt haben.

Anhang A

Beweise

A.2 Zu Kapitel 2

A.2.1 Lemma 2.6

> **Lemma 2.6: Abschätzungen der min-Entropie**
> Für jede diskrete Zufallsvariable X gilt:
> $$H_\infty(X) \leq H(X), \qquad H_\infty(X) \leq H_{\mathrm{Ren}}(X). \qquad \square$$

Beweis: Es sei M der Wertebereich von X, dann ist

$$H(X) = \sum_{x \in M} P(X=x)\big(-\log P(X=x)\big) \geq \sum_{x \in M} P(X=x) H_\infty(X) = H_\infty(X)$$

und für eine von X unabhängige Zufallsvariable X' gleicher Verteilung gilt

$$P(X=X') = \sum_{x' \in M} P(X'=x'|X=x')P(X=x') \leq \sum_{x' \in M} P(X'=x') \max_{x \in M} P(X=x) = \max_{x \in M} P(X=x),$$

woraus sich

$$H_{\mathrm{Ren}}(X) = -\log P(X=X') \geq -\log \max_{x \in M} P(X=x) = H_\infty(X)$$

ergibt. ∎

A.2.2 Lemma 2.8

> **Lemma 2.8: Statistischer Abstand**
> Für diskrete Zufallsvariablen X und Y gilt
> $$\mathrm{SD}(X;Y) = \max_{T \subseteq M} \big|P(X \in T) - P(Y \in T)\big|,$$
> wobei M die Vereinigung der Wertebereiche von X und Y sei. $\square$

Beweis: Zu zeigen ist

$$\frac{1}{2} \sum_{a \in M} \big|P(X=a) - P(Y=a)\big| = \max_{T \subseteq M} \big|P(X \in T) - P(Y \in T)\big|,$$

wobei M die Vereinigung der Wertebereiche von X und Y sei.
Es sei $T^* := \{a \in M : P(X=a) > P(Y=a)\}$. Damit ist

$$\begin{aligned}
\frac{1}{2} \sum_{a \in M} \big|P(X=a) - P(Y=a)\big| &= \frac{1}{2} \sum_{a \in T^*} \big(P(X=a) - P(Y=a)\big) + \frac{1}{2} \sum_{a \in M \setminus T^*} \big(P(Y=a) - P(X=a)\big) \\
&= \tfrac{1}{2}\big(P(X \in T^*) - P(Y \in T^*)\big) + \tfrac{1}{2}\big(P(Y \notin T^*) - P(X \notin T^*)\big) \\
&= \tfrac{1}{2}\big|P(X \in T^*) - P(Y \in T^*)\big| + \tfrac{1}{2}\big|(1 - P(Y \in T^*)) - (1 - P(X \in T^*))\big| \\
&= \big|P(X \in T^*) - P(Y \in T^*)\big| \\
&\leq \max_{T \subseteq M} \big|P(X \in T) - P(Y \in T)\big|. \qquad (14)
\end{aligned}$$

Weiterhin gilt

$$\begin{aligned}
\max_{T\subseteq M}\bigl|P(X\in T)-P(Y\in T)\bigr| &= \max_{T\subseteq M}\Bigl(\tfrac{1}{2}\bigl|P(X\in T)-P(Y\in T)\bigr|+\tfrac{1}{2}\bigl|P(X\notin T)-P(Y\notin T)\bigr|\Bigr)\\
&\le \max_{T\subseteq M}\Bigl(\tfrac{1}{2}\sum_{a\in T}\bigl|P(X=a)-P(Y=a)\bigr|+\tfrac{1}{2}\sum_{a\in M\setminus T}\bigl|P(X=a)-P(Y=a)\bigr|\Bigr)\\
&= \tfrac{1}{2}\sum_{a\in M}\bigl|P(X=a)-P(Y=a)\bigr|. \qquad (15)
\end{aligned}$$

Aus (14) und (15) zusammen folgt die Behauptung. ■

A.2.3 Lemma 2.9

Lemma 2.9: Eigenschaften des statistischen Abstands
Es seien X, Y, Z, U Zufallsvariablen, U unabhängig von $\{X,Y,Z\}$, und f eine Funktion, die mindestens auf den Wertebereichen von X und Y definiert ist. Dann gilt

$$\begin{aligned}
\mathrm{SD}(X;Y) &\ge \mathrm{SD}(f(X);f(Y)), && (1)\\
\mathrm{SD}(X;Y) &= \mathrm{SD}(XU;YU), && (2)\\
\mathrm{SD}(X;Z) &\le \mathrm{SD}(X;Y)+\mathrm{SD}(Y;Z), && (3)\\
\mathrm{SD}(XZ;YZ) &= \sum_{z\in M_Z} P(Z=z)\,\mathrm{SD}(X;Y\|Z=z), && (4)
\end{aligned}$$

wobei M_Z der Wertebereich von Z sei.

Ist f injektiv, so liegt in (1) Gleichheit vor. □

Beweis: Es seien M_X, M_Y, M_Z, M_U die Wertebereiche von X, Y, Z, U. Weiter sei f o. B. d. A. nur auf $M := M_X \cup M_Y$ definiert. Dann ist nach Lemma 2.8:

$$\begin{aligned}
\mathrm{SD}(f(X);f(Y)) &= \max_{T_f\subseteq f(M)}\bigl|P(f(X)\in T_f)-P(f(Y)\in T_f)\bigr|\\
&= \max_{T_f\subseteq f(M)}\bigl|P(X\in f^{-1}(T_f))-P(Y\in f^{-1}(T_f))\bigr|\\
&\le \max_{T\subseteq M}\bigl|P(X\in T)-P(Y\in T)\bigr|.
\end{aligned}$$

Damit ist (1) gezeigt.

Ist f injektiv, so folgt aus

$$\mathrm{SD}(X;Y)\ge \mathrm{SD}(f(X);f(Y)) \qquad \text{und} \qquad \mathrm{SD}(f(X);f(Y))\ge \mathrm{SD}(f^{-1}\circ f(X);f^{-1}\circ f(Y))$$

Gleichheit in (1).

Es sei $M' := M_X \cup M_Y \cup M_Z$. Gleichung (3) beweisen wir wie folgt:

$$\begin{aligned}
\mathrm{SD}(X;Z) &= \tfrac{1}{2}\sum_{a\in M_X\cup M_Z}\bigl|P(X=a)-P(Z=a)\bigr|\\
&\le \tfrac{1}{2}\sum_{a\in M_X\cup M_Z}\Bigl(\bigl|P(X=a)-P(Y=a)\bigr|+\bigl|P(Y=a)-P(Z=a)\bigr|\Bigr)\\
&\le \tfrac{1}{2}\sum_{a\in M'}\bigl|P(X=a)-P(Y=a)\bigr|+\tfrac{1}{2}\sum_{a\in M'}\bigl|P(Y=a)-P(Z=a)\bigr|\\
&= \mathrm{SD}(X;Y)+\mathrm{SD}(Y;Z)
\end{aligned}$$

Es ergibt sich (4) wie folgt:

$$\begin{aligned}
\mathrm{SD}(XZ;YZ) &= \tfrac{1}{2}\sum_{\substack{a\in M\\ z\in M_Z}} \big|P(X=a,Z=z)-P(Y=a,Z=z)\big|\\
&= \tfrac{1}{2}\sum_{\substack{a\in M\\ z\in M_Z}} \big|P(X=a|Z=z)\,P(Z=z)-P(Y=a|Z=z)\,P(Z=z)\big|\\
&= \sum_{z\in M_Z} P(Z=z)\ \tfrac{1}{2}\sum_{a\in M}\big|P(X=a|Z=z)-P(Y=a|Z=z)\big|\\
&= \sum_{z\in M_Z} P(Z=z)\ \ \mathrm{SD}(X;Y\|Z=z).
\end{aligned}$$

Und schließlich ergibt sich (2) wegen der Unabhängigkeit von U von $\{X,Y\}$ und

$$\mathrm{SD}(XU;YU) \overset{(4)}{=} \sum_{u\in M_U} P(U=u)\ \mathrm{SD}(X;Y\|U=u) = \sum_{u\in M_U} P(U=u)\ \mathrm{SD}(X;Y) = \mathrm{SD}(X;Y). \qquad \blacksquare$$

A.2.4 Lemma 2.12

> **Definition 2.10: Perfekt zufällig**
> Sei S eine diskrete Zufallsvariable mit Werten aus M_S. Eine Quelle X über einem Alphabet Σ heißt *perfekt zufällig unter Kenntnis von S,* wenn für alle $n \in \mathbb{N}\cup\{\infty\}$ und $s\in M_S$ mit $P(|X|=n,S=s)>0$ gilt: $X\big|(|X|=n,S=s)$ ist gleichverteilt auf Σ^n (mit $\Sigma^\infty := \Sigma^{\mathbb{N}}$).
> Wird kein S angegeben, so setzen wir $S:=\lambda$. □

> **Lemma 2.12: Konkatenation von Zufallsquellen**
> Es seien $S,U_1,\ldots,U_n$ diskrete Zufallsvariablen, und U_i sei perfekt zufällig über Σ unter Kenntnis von S, U_j $(j\neq i)$.
> Dann ist $U_1\ldots U_n$ perfekt zufällig unter Kenntnis von S. □

Beweis: Wir zeigen die Aussage für $n=2$, der allgemeine Fall folgt induktiv.

Es seien $U:=U_1U_2$, $l\in\mathbb{N}_0$ und $u\in\Sigma^l$.

Da U_1 perfekt zufällig unter Kenntnis von U_2 und S ist, gilt für $a=0,\ldots,l$ mit $P(|U_1|=a,\ |U|=l,\ S=s)>0$:

$$P\big(U_1=u_1\ldots u_a \;\big|\; |U_1|=a,\ |U|=l,\ S=s\big) = \#\Sigma^{-a}. \tag{16}$$

Und da U_2 perfekt zufällig unter Kenntnis von U_1 und S ist, gilt für $a=0,\ldots,l$ mit $P(|U_1|=a,U_1=u_1\ldots u_a,$ $|U|=l,S=s)>0$:

$$P\big(U_2=u_{a+1}\ldots u_l \;\big|\; |U_1|=a,\ U_1=u_1\ldots u_a,\ |U|=l,\ S=s\big) = \#\Sigma^{a-l}. \tag{17}$$

Damit folgt für $P(|U|=l,\ S=s)>0$:

$$\begin{aligned}
P\big(U=u \;\big|\; |U|=l,\ S=s\big) \ &=\ \sum_{a=0}^{l} P\big(U_1=u_1\ldots u_a \;\big|\; |U_1|=a,\ |U|=l,\ S=s\big)\cdot\\
&\qquad P\big(U_2=u_{a+1}\ldots u_l \;\big|\; |U_1|=a,\ U_1=u_1\ldots u_a,\ |U|=l,\ S=s\big)\cdot\\
&\qquad P\big(|U_1|=a \;\big|\; |U|=l,\ S=s\big)\\
&\overset{(16,17)}{=} \sum_{a=0}^{l} \#\Sigma^{-a}\,\#\Sigma^{a-l}\,P\big(|U_1|=a \;\big|\; |U|=l,\ S=s\big)\\
&=\ \#\Sigma^{-l}.
\end{aligned}$$

■

A.3 Zu Kapitel 3

A.3.1 Lemma 3.1

> **Lemma 3.1: Unmöglichkeit deterministischer Extraktion**
> Sei M eine Menge, Σ ein Alphabet mit $\#\Sigma =: n$, $k \in \mathbb{R}_{\geq 0}$ und $k \leq \log \#M - \log n$. Weiter sei $\mathcal{X}$ die Menge aller Zufallsvariablen X mit Werten in M und $H_\infty(X) \geq k$, und schließlich $f : M \to \Sigma \cup \{\perp\}$.
>
> Dann existiert ein $X \in \mathcal{X}$, so daß $P(f(X) \in \{\sigma, \perp\}) = 1$ für ein $\sigma \in \Sigma$, und so daß für jede über Σ perfekt zufällige Zufallsvariable U gilt:
>
> $$\mathrm{SD}(f(X); U) \geq \frac{n-1}{n}(1 - P(U = \perp)). \qquad \square$$

Beweis: Es gilt ein $X \in \mathcal{X}$ zu konstruieren, das der obigen Bedingung genügt.

Es sei $S_i := f^{-1}(i)$ für $i \in \Sigma \cup \{\perp\}$. Dann sei $\sigma \in \Sigma$ so gewählt, daß $\#S_\sigma \geq \#S_{\tilde{\sigma}}$ für alle $\tilde{\sigma} \in \Sigma$.

Es ist damit

$$\#S_\sigma + \#S_\perp \geq \tfrac{\#M}{n}. \tag{18}$$

Es sei nun X eine Zufallsvariable mit Werten in M und folgender Verteilung:

$$P(X = x) = \begin{cases} \frac{1}{\#S_\sigma + \#S_\perp}, & \text{falls } f(x) \in \{\sigma, \perp\}, \\ 0, & \text{sonst.} \end{cases}$$

Nach (18) ist dann $P(X = x) \leq \frac{n}{\#M}$, also $H_\infty(X) \geq k$, und somit $X \in \mathcal{X}$.

Offensichtlich ist $P(f(X) \in \{\sigma, \perp\}) = 1$.

Es bleibt $\mathrm{SD}(f(X), U)$ nach unten abzuschätzen.

Nach Lemma 2.9 (1) können wir U durch U' ersetzen, wobei $U' := U$ für $U \in \Sigma \cup \{\perp\}$ und $U' := \perp$ sonst, da $f(X)$ unter dieser Abbildung invariant ist, und der statistische Abstand wird sich nicht vergrößern. Somit können wir o. B. d. A. $U \in \Sigma \cup \{\perp\}$ annehmen.

Es sei $\varepsilon := P(U = \perp)$ und $\delta := P(f(X) = \perp)$. Dann ist

$$\begin{aligned}
2 \cdot \mathrm{SD}(f(X), U) &= \big|P(f(X) = \perp) - P(U = \perp)\big| \\
&\quad + \big|P(f(X) = \sigma) - P(U = \sigma)\big| \\
&\quad + \sum_{\tilde{\sigma} \in \Sigma \setminus \{\sigma\}} \big|P(f(X) = \tilde{\sigma}) - P(U = \tilde{\sigma})\big| \\
&= \big|\delta - \varepsilon\big| + \big|(1-\delta) - \tfrac{1-\varepsilon}{n}\big| + (n-1)\tfrac{1-\varepsilon}{n} \\
&\geq (n-1)\tfrac{1-\varepsilon}{n} + \min \begin{cases} \varepsilon + 1 - \frac{1-\varepsilon}{n}, & (\delta = 0), \\ 1 - \varepsilon + \frac{1-\varepsilon}{n}, & (\delta = 1), \\ 1 - \varepsilon - \frac{1-\varepsilon}{n}, & (\delta = \varepsilon), \\ 1 - \frac{1-\varepsilon}{n} - \varepsilon, & (\delta = 1 - \frac{1-\varepsilon}{n}) \end{cases} \\
&= (n-1)\tfrac{1-\varepsilon}{n} + 1 - \varepsilon - \tfrac{1-\varepsilon}{n} = \frac{2n-2}{n}(1-\varepsilon).
\end{aligned} \tag{19}$$

Wir haben bei der Abschätzung nur vier mögliche Werte für δ betrachtet, dies ist zulässig, da zwischen diesen Werten der statistische Abstand monoton in δ ist, und somit das Minimum an einem dieser Punkte angenommen wird.

Es ergibt sich aus (19) die noch ausstehende Eigenschaft

$$\mathrm{SD}(f(X), U) \geq \frac{n-1}{n}(1 - P(U = \perp)). \qquad \blacksquare$$

A.3.2 Leftover Hash Lemma, 1. Fassung

Definition 3.2: Universelle Hashfunktion
Es sei
$$h : M_R \times M_X \longrightarrow M_{\hat{X}}.$$
Dann heißt h *universelle Hashfunktion*, wenn $\# M_X > 1$ und für alle $x, x' \in M_X$, $x \neq x'$ und $a, a' \in M_{\hat{X}}$ gilt:
$$P\big(h(R,x) = a \ \wedge \ h(R,x') = a'\big) = (\# M_{\hat{X}})^{-2},$$
wobei R eine auf M_R gleichverteilte Zufallsvariable sei. □

Lemma 3.3: Leftover Hash Lemma, 1. Fassung
Es seien X, R, U Zufallsvariablen mit Werten in M_X, M_R bzw. $M_{\hat{X}}$, sowie $k \in \mathbb{R}$. Hierbei sei R gleichverteilt auf M_R, U gleichverteilt auf $M_{\hat{X}}$, $H_{\text{Ren}}(X) \geq k$, sowie X, R, U stochastisch unabhängig. Weiter sei $h : M_R \times M_X \to M_{\hat{X}}$ eine universelle Hashfunktion.

Dann ist
$$\mathrm{SD}(R, h(R,X); R, U) \leq \tfrac{1}{2}\sqrt{\# M_{\hat{X}} \cdot 2^{-k}}.$$ □

Beweis: Für beliebige $x, x' \in M_X$, $a \in M_{\hat{X}}$ gilt
$$P(h(R,x) = a) = \sum_{a \in M_{\hat{X}}} P\big(h(R,x) = a,\ h(R,x') = a'\big) = \sum_{a \in M_{\hat{X}}} \# M_{\hat{X}}^{-2} = \# M_{\hat{X}}^{-1}. \tag{20}$$

Es sei X' eine von R und X unabhängige Zufallsvariable gleicher Verteilung wie X.

Wir setzen
$$\begin{aligned} Z_{r,a} &:= P(h(R,X) = a \mid R = r) - P(U = a) \\ &= P(h(r,X) = a) - P(U = a) \end{aligned}$$

und
$$E_a(C) := \mathrm{E}\big((\delta(h(R,X) = a) - P(U = a)) \cdot (\delta(h(R,X') = a) - P(U = a)) \mid C\big),$$

wobei wir E_a statt $E_a(C)$ schreiben, falls $P(C) = 1$.

Es ist dann
$$(\mathrm{E}|Z_{R,a}|)^2 \leq \mathrm{E}\, Z_{R,a}^2 = E_a \tag{21}$$

Um E_a zu errechnen, untersuchen wir zunächst die folgenden bedingten Erwartungswerte (mit $x, x' \in M_X$, $x \neq x'$):
$$\begin{aligned} E_a(X = x, X' = x') &= P(h(R,x) = a, h(R,x') = a) - P(h(R,x) = a)P(U = a) \\ &\quad - P(h(R,x') = a)P(U = a) + P(U = a)^2 \\ &\overset{(20)}{=} \# M_{\hat{X}}^{-2} - \# M_{\hat{X}}^{-1} \# M_{\hat{X}}^{-1} - \# M_{\hat{X}}^{-1} \# M_{\hat{X}}^{-1} + \# M_{\hat{X}}^{-2} \\ &= 0 \end{aligned} \tag{22}$$
$$\begin{aligned} E_a(X = X' = x) &= P(h(R,x) = a) - 2P(h(R,x) = a)P(U = a) + P(U = a)^2 \\ &= \# M_{\hat{X}}^{-1} - 2\# M_{\hat{X}}^{-1} \# M_{\hat{X}}^{-1} + \# M_{\hat{X}}^{-2} \\ &\leq \# M_{\hat{X}}^{-1} \end{aligned} \tag{23}$$

Diese Ungleichungen gelten natürlich nur unter der Bedingung, daß die jeweiligen bedingten Erwartungswerte definiert sind.

Es ist also

$$
\begin{aligned}
E_a &= \sum_{x \in M_X} P(X = X' = x)\, E_a(X = X' = x) \\
&\quad + \sum_{\substack{x,x' \in M_X \\ x \neq x'}} P(X = x, X' = x')\, E_a(X = x, X' = x') \\
&\overset{(22),(23)}{\leq} \sum_{x \in M_X} P(X = X' = x)\, \# M_{\hat{X}}^{-1} \\
&= P(X = X')\, \# M_{\hat{X}}^{-1} = 2^{-H_{\text{Ren}}(X)} \# M_{\hat{X}}^{-1} \\
&\leq 2^{-k} \# M_{\hat{X}}^{-1}. \qquad (24)
\end{aligned}
$$

Damit ergibt sich schließlich

$$
\begin{aligned}
\mathrm{SD}(R, h(R,X);\ R, U) &= \tfrac{1}{2} \sum_{\substack{a \in M_{\hat{X}} \\ r \in M_R}} \left| P(R = r, h(R,X) = a) - P(R = r)\, P(U = a) \right| \\
&= \tfrac{1}{2} \sum_{a \in M_{\hat{X}}} \sum_{r \in M_R} P(R = r) |Z_{r,a}| = \tfrac{1}{2} \sum_{a \in M_{\hat{X}}} \mathrm{E}|Z_{R,a}| \\
&\overset{(21)}{\leq} \tfrac{1}{2} \# M_{\hat{X}} \sqrt{E_a} \\
&\overset{(24)}{\leq} \tfrac{1}{2} \# M_{\hat{X}} \sqrt{2^{-k} \# M_{\hat{X}}^{-1}} \\
&= \tfrac{1}{2} \sqrt{\# M_{\hat{X}} \cdot 2^{-k}}. \qquad \blacksquare
\end{aligned}
$$

A.3.3 Leftover Hash Lemma, 2. Fassung

Definition 3.4: Universelle Quasi-Hashfunktion
Es sei

$$h : M_R \times M_X \longrightarrow M_{\hat{X}}.$$

Dann heißt h *universelle Quasi-Hashfunktion*, wenn es eine Familie von Bijektionen $f_{\tilde{r}} : M_{\hat{X}} \to M'_{\hat{X}}$, $\tilde{r} \in M_{\tilde{R}}$ gibt, so daß

$$
\begin{array}{rccc}
\tilde{h}: & (M_R \times M_{\tilde{R}}) \times M_X & \longrightarrow & M'_{\hat{X}} \\
& (r, \tilde{r}),\ x & \longmapsto & f_{\tilde{r}}(h(r,x))
\end{array}
$$

eine universelle Hashfunktion ist. □

Lemma 3.5: Leftover Hash Lemma, 2. Fassung
Es seien X, R, U Zufallsvariablen mit Werten in M_X, M_R bzw. $M_{\hat{X}}$, sowie $k \in \mathbb{R}$. Hierbei sei R gleichverteilt auf M_R, U gleichverteilt auf $M_{\hat{X}}$, $H_{\text{Ren}}(X) \geq k$, sowie X, R, U stochastisch unabhängig. Weiter sei $h : M_R \times M_X \to M_{\hat{X}}$ eine universelle *Quasi*-Hashfunktion.

Dann ist

$$\mathrm{SD}(R, h(R,X); R, U) \leq \tfrac{1}{2} \sqrt{\# M_{\hat{X}} \cdot 2^{-k}}. \qquad \square$$

Beweis: Es seien $\tilde{h}$, $f_{\tilde{r}}$ und $M_{\tilde{R}}$ wie in Definition 3.4 (universelle Quasi-Hashfunktion). Weiter sei $\tilde{R}$ eine von R, X und U unabhängige und auf $M_{\tilde{R}}$ gleichverteilte Zufallsvariable.

Dann gilt nach Lemma 3.5

$$\mathrm{SD}(R, \tilde{R}, f_{\tilde{R}}(h(R,X));\ R, \tilde{R}, U) = \mathrm{SD}\big((R, \tilde{R}), \tilde{h}((R, \tilde{R}), X);\ (R, \tilde{R}), U\big) \overset{3.5}{\leq} \tfrac{1}{2} \sqrt{\# M_{\hat{X}} \cdot 2^{-k}} =: \varepsilon. \qquad (25)$$

Durch

$$\pi(r, \tilde{r}, x) := (r, \tilde{r}, f_{\tilde{r}}^{-1}(x))$$

ist eine Bijektion auf $M_R \times M_{\tilde{R}} \times M_{\hat{X}}$ definiert. Anwendung von π auf die linke Seite von (25) liefert nach Lemma 2.9 (1):

$$\mathrm{SD}\big(R, \tilde{R}, h(R, X);\ \pi(R, \tilde{R}, U)\big) \leq \varepsilon.$$

Da $(R, \tilde{R}, U)$ auf dem Definitionsbereich von π gleichverteilt ist, haben $\pi(R, \tilde{R}, U)$ und $(R, \tilde{R}, U)$ die gleiche Verteilung, also ergibt sich

$$\mathrm{SD}\big(R, \tilde{R}, h(R, X);\ R, \tilde{R}, U\big) \leq \varepsilon.$$

Schließlich liefert Lemma 2.9 (2) wegen der Unabhängigkeit von R:

$$\mathrm{SD}\big(R, h(R, X);\ R, U\big) \leq \varepsilon. \qquad \blacksquare$$

A.3.4 Leftover Hash Lemma

Satz 3.6: Leftover Hash Lemma
Es seien X, R, U und S Zufallsvariablen mit Werten in M_X, M_R, $M_{\hat{X}}$ bzw. M_S, sowie $k \in \mathbb{R}$. Dabei seien (X, S), R und U unabhängig. Es sei U auf $M_{\hat{X}}$ und R auf M_R gleichverteilt. Schließlich seien $H_{\mathrm{Ren}}(X) \geq k$ und $h : M_R \times M_X \to M_{\hat{X}}$ eine universelle Quasi-Hashfunktion.

Dann ist

$$\mathrm{SD}(S, R, h(R, X);\ S, R, U) \leq \tfrac{1}{2}\, \#M_S \sqrt{\#M_{\hat{X}} \cdot 2^{-k}}. \qquad \square$$

Beweis: Gegeben $S = s$ erfüllen X, R und U die Bedingungen für Lemma 3.5 mit $k := H_{\mathrm{Ren}}(X|S = s)$, also ergibt sich

$$\begin{aligned}
\mathrm{SD}(S, R, h(R, X);\ S, R, U) &\overset{2.9(4)}{=} \sum_{s \in M_S} P(S = s)\ \mathrm{SD}(R, h(R, X);\ R, U \parallel S = s) \\
&\overset{3.5}{\leq} \sum_{s \in M_S} P(S = s)\, \tfrac{1}{2} \sqrt{\#M_{\hat{X}} \cdot 2^{-H_{\mathrm{Ren}}(X|S=s)}} \\
&= \sum_{s \in M_S} P(S = s)\, \tfrac{1}{2} \sqrt{\#M_{\hat{X}} \sum_{x \in M_X} P(X = x|S = s)^2} \\
&= \sum_{s \in M_S} \tfrac{1}{2} \sqrt{\#M_{\hat{X}} \sum_{x \in M_X} P(X = x|S = s)^2\, P(S = s)^2} \\
&= \sum_{s \in M_S} \tfrac{1}{2} \sqrt{\#M_{\hat{X}} \sum_{x \in M_X} P(X = x, S = s)^2} \\
&\leq \sum_{s \in M_S} \tfrac{1}{2} \sqrt{\#M_{\hat{X}} \sum_{x \in M_X} P(X = x)^2} \\
&= \sum_{s \in M_S} \tfrac{1}{2} \sqrt{\#M_{\hat{X}} \cdot 2^{-H_{\mathrm{Ren}}(X)}} \\
&\leq \tfrac{1}{2}\, \#M_S \sqrt{\#M_{\hat{X}} \cdot 2^{-k}}.
\end{aligned} \qquad \blacksquare$$

A.3.5 Lemma 3.7

Lemma 3.7: Affine Transformationen als universelle Hashfunktion
Es sei $\mathbb{F}$ ein endlicher Körper, $M_X := \mathbb{F}^n$, $M_{\hat{X}} := \mathbb{F}^m$ mit $n \geq 1$, $m \leq n$, und $M_R := \mathbb{F}^{m \times n} \times \mathbb{F}^m \cong \mathbb{F}^{m(n+1)}$. Dann ist

$$\begin{aligned}
h:\ M_R \times M_X &\longrightarrow M_{\hat{X}} \\
(M, b),\ x &\longmapsto Mx + b
\end{aligned}$$

eine universelle Hashfunktion. □

Beweis: Ist $m = 0$, also $\#M_{\hat{X}} = 1$, so folgt die Behauptung trivial aus der Definition universeller Hashfunktionen.

Es sei R eine gleichverteilte Zufallsvariable mit Werten in M_R.

Seien $n = m = 1$. Für $x, x' \in \mathbb{F}$, $x \neq x$ und $a, a' \in \mathbb{F}$ gilt dann $Mx + b = a$ und $Mx' + b = a'$ genau für $M = \frac{a-a'}{x-x'}$ und $b = a - Mx$. Somit ist

$$P(h(R,x) = a,\ h(R,x') = a') = \#M_R^{-1} = \#\mathbb{F}^{-2} = \#M_{\hat{X}}^{-2}.$$

Also können wir $n \geq 2$ annehmen. Seien $x, x' \in \mathbb{F}^n$, $x \neq x'$ und $a, a' \in \mathbb{F}^m$. Wähle nun $b' \in \mathbb{F}^n$ so, daß $x + b'$ und $x' + b'$ linear unabhängig sind (dies ist möglich, da $\dim \mathbb{F}^n \geq 2$), und ein reguläres $S \in \mathbb{F}^{n\times n}$, so daß $S(x + b') = e_1$ und $S(x' + b') = e_2$.

Wir setzen

$$\mathcal{M} := \{r \in M_R :\ h(r,x) = a,\ h(r,x') = a'\},$$

dann ist

$$\begin{aligned}
\#\mathcal{M} &= \#\{\, (M,b) \in \mathbb{F}^{m\times n} \times \mathbb{F}^m \ :\ Mx + b = a,\ Mx' + b = a' \,\} \\
&= \#\{\, (\tilde{M},\tilde{b}) \in \mathbb{F}^{m\times n} \times \mathbb{F}^m \ :\ Mx + b = a,\ Mx' + b = a' \\
&\qquad \text{mit } M := \tilde{M}S \text{ und } b := \tilde{M}Sb' + \tilde{b} \,\} \\
&= \#\{\, (\tilde{M},\tilde{b}) \in \mathbb{F}^{m\times n} \times \mathbb{F}^m \ :\ \tilde{M}Sx + \tilde{M}Sb' + \tilde{b} = a,\ \tilde{M}Sx' + \tilde{M}Sb' + \tilde{b} = a' \,\} \\
&= \#\{\, (\tilde{M},\tilde{b}) \in \mathbb{F}^{m\times n} \times \mathbb{F}^m \ :\ \tilde{M}\,\underbrace{S(x + b')}_{=e_1} = a - \tilde{b},\ \tilde{M}\,\underbrace{S(x' + b')}_{=e_2} = a' - \tilde{b} \,\} \\
&= \#\{\, (\tilde{M},\tilde{b}) \in \mathbb{F}^{m\times n} \times \mathbb{F}^m \ :\ \tilde{M} \text{ hat } a - \tilde{b} \text{ in der ersten und } a' - \tilde{b} \text{ in der zweiten Spalte} \,\} \\
&= \#\mathbb{F}^{m(n-2)+m}.
\end{aligned}$$

Also

$$P\big(h(R,x) = a,\ h(R,x') = a'\big) = \frac{\#\mathcal{M}}{\#M_R} = \#\mathbb{F}^{m(n-2)+m-mn-m} = \#\mathbb{F}^{-2m} = \#M_{\hat{X}}^{-2}.$$ ■

A.3.6 Lemma 3.8

Lemma 3.8: Affine Toeplitz-Transformationen als universelle Hashfunktion
Es sei $\mathbb{F}$ ein endlicher Körper, $M_X := \mathbb{F}^n$, $M_{\hat{X}} := \mathbb{F}^m$ mit $n \geq 1$, $m \leq n$, und $M_R := \mathrm{Toeplitz}(\mathbb{F}^{m\times n}) \times \mathbb{F}^m \cong \mathbb{F}^{2m+n-1}$. Dann ist

$$\begin{array}{rrcl} h: & M_R \times M_X & \longrightarrow & M_{\hat{X}} \\ & (M,b),\ x & \longmapsto & Mx + b \end{array}$$

eine universelle Hashfunktion. □

Beweis: Es seien $x = (x_i), x' = (x'_i) \in \mathbb{F}^n$, $x \neq x'$, $a = (a_i), a' = (a'_i) \in \mathbb{F}^m$ und R eine auf M_R gleichverteilte Zufallsvariable.

Wir setzen

$$\mathcal{M} := \{r \in M_R :\ h(r,x) = a,\ h(r,x') = a'\},$$

Da jedes $M = (m_{ij}) \in \mathrm{Toeplitz}(\mathbb{F}^{m\times n})$ eineindeutig bestimmt ist durch $t_{-n+1}, \ldots, t_{m-1}$ mit $t_{i-j} := m_{ij}$, entsprechen die Elemente von $\mathcal{M}$ den Lösungen von

$$a_i = \sum_{k=1}^{n} t_{i-k} x_k + b_i, \qquad a'_i = \sum_{k=1}^{n} t_{i-k} x'_k + b_i \qquad\qquad (i = 1, \ldots, m).$$

Dieses Gleichungssystem können wir schreiben als

$$\begin{pmatrix} \mathbb{1}_m & T(x) \\ \mathbb{1}_m & T(x') \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b_m \\ t_{-n+1} \\ \vdots \\ t_{m-1} \end{pmatrix} = \begin{pmatrix} a_1 \\ \vdots \\ a_m \\ a'_1 \\ \vdots \\ a'_m \end{pmatrix} \tag{26}$$

mit $T(v) \in \mathbb{F}^{m\times(m+n-1)}$ und

$$(T(v))_{i\nu} := \begin{cases} v_{n+i-\nu}, & \text{falls } \nu \in \{i, \dots, i+n-1\}, \\ 0, & \text{sonst,} \end{cases}$$

denn

$$\left(T(x)\,(t_{-n+1}, \dots, t_{m-1})^T\right)_i = \sum_{\nu=1}^{n+m-1} (T(x))_{i\nu} t_{\nu-n} = \sum_{\nu=i}^{i+n-1} x_{n+i-\nu} t_{\nu-n} = \sum_{k=1}^{n} x_k t_{i-k} \qquad (i = 1, \dots, m)$$

und analog für $T(x')$ und a'.

Da $T(v)$ für $v \in \mathbb{F}^n \setminus \{0\}$ maximalen Rang m hat, gilt

$$\operatorname{Rg} \begin{pmatrix} \mathbb{1}_m & T(x) \\ \mathbb{1}_m & T(x') \end{pmatrix} = \operatorname{Rg} \begin{pmatrix} \mathbb{1}_m & T(x) \\ 0 & T(x'-x) \end{pmatrix} = \operatorname{Rg} \mathbb{1}_m + \operatorname{Rg} T(x'-x) = 2m.$$

Damit hat die Matrix in (26) maximalen Rang $2m$ und liegt in $\mathbb{F}^{2m\times(2m+n-1)}$, somit hat das Gleichungssystem $\#\mathbb{F}^{n-1}$ Lösungen, und $\#\mathcal{M} = \#\mathbb{F}^{n-1}$.

Es folgt schließlich

$$P(h(R,x) = a,\ h(R,x') = a') = \frac{\#\mathcal{M}}{\#M_R} = \#\mathbb{F}^{(n-1)-(m+n-1+m)} = \#\mathbb{F}^{-2m} = \#M_{\hat{X}}^{-2}. \qquad \blacksquare$$

A.3.7 Lemmata 3.9 und 3.10

Lemma 3.9: Lineare Abbildungen als universelle Quasi-Hashfunktion

Es sei $\mathbb{F}$ ein endlicher Körper, $M_X := \mathbb{F}^n$, $M_{\hat{X}} := \mathbb{F}^m$ mit $n \geq 1$, $m \leq n$, und $M_R := \mathbb{F}^{m\times n} \cong \mathbb{F}^{mn}$. Dann ist

$$\begin{aligned} h:\ M_R \times M_X &\longrightarrow M_{\hat{X}} \\ M,\ x &\longmapsto Mx \end{aligned}$$

eine universelle Quasi-Hashfunktion. □

Lemma 3.10: Toeplitz-Transformationen als universelle Quasi-Hashfunktion

Es sei $\mathbb{F}$ ein endlicher Körper, $M_X := \mathbb{F}^n$, $M_{\hat{X}} := \mathbb{F}^m$ mit $n \geq 1$, $m \leq n$, und $M_R := \operatorname{Toeplitz}(\mathbb{F}^{m\times n}) \cong \mathbb{F}^{m+n-1}$. Dann ist

$$\begin{aligned} h:\ M_R \times M_X &\longrightarrow M_{\hat{X}} \\ M,\ x &\longmapsto Mx \end{aligned}$$

eine universelle Quasi-Hashfunktion. □

Beweis: Da die beiden Lemmata fast identische Beweise haben, werden jene hier zu einem zusammengefaßt.

Setze $M_{\tilde{R}} := \mathbb{F}^m$ und $f_{\tilde{r}}(x) := x + \tilde{r}$. Mit der Notation aus Definition 3.4 ist

$$\begin{aligned} \tilde{h}:\ (M_R \times M_{\tilde{R}}) \times M_X &\longrightarrow M_{\hat{X}} \\ (M, b),\ x &\longmapsto f_b(h(M,x)) = Mx + b \end{aligned}$$

eine universelle Hashfunktion nach Lemma 3.7 bzw. 3.8, somit ist h eine universelle Quasi-Hashfunktion. ■

A.3.8 Lemma 3.11

Lemma 3.11: Vergrößerung des initialen Zufalls einer Hashfunktion
Ist $h : M_{f(R)} \times M_X \to M_{\hat{X}}$ eine universelle Quasi-Hashfunktion, und $f : M_R \to M_{f(R)}$ eine Abbildung mit $\#f^{-1}(r) = \#f^{-1}(r')$ für alle $r, r' \in M'_R$, dann ist auch

$$\begin{array}{rccc} h_f : & M_R \times M_X & \longrightarrow & M_{\hat{X}} \\ & r,\ x & \longmapsto & h(f(r), x), \end{array}$$

eine universelle Quasi-Hashfunktion. Ist h eine universelle Hashfunktion, so ist h_f auch eine universelle Hashfunktion. □

Beweis: Sei $c := \#f^{-1}(r)$ mit $r \in M'_R$.

Zunächst wollen wir die Aussage für den Fall beweisen, daß h eine universelle Hashfunktion ist.

Seien $x, x' \in M_X$, $x \neq x'$ und $a, a' \in M_{\hat{X}}$, sowie R eine auf M_R gleichverteilte Zufallsvariable. Weil $\#f^{-1}(r)$ unabhängig ist von $r \in M_R$, ist $f(R)$ auf $M_{f(R)}$ gleichverteilt.

$$P\big(h_f(R,x) = a \ \wedge\ h_f(R,x') = a'\big) = P\big(h(f(R),x) = a \ \wedge\ h(f(R),x') = a'\big) \overset{3.2}{=} (\#M_{\hat{X}})^{-2},$$

also ist h_f eine universelle Hashfunktion.

Sei nun h eine universelle Quasi-Hashfunktion.

Nach Definition 3.4 existiert eine Familie von Bijektionen $f_{\tilde{r}} : M_{\hat{X}} \to M'_{\hat{X}}$, $\tilde{r} \in M_{\tilde{R}}$, so daß

$$\begin{array}{rccc} \tilde{h} : & (M_{f(R)} \times M_{\tilde{R}}) \times M_X & \longrightarrow & M'_{\hat{X}} \\ & (r, \tilde{r}),\ x & \longmapsto & f_{\tilde{r}}(h(r,x)) \end{array}$$

eine universelle Hashfunktion ist. Nach dem bereits bewiesenen Teil des Lemmas ist für $\tilde{f}(r, \tilde{r}) := (f(r), \tilde{r})$ auch

$$\begin{array}{rccc} \tilde{h}_{\tilde{f}} : & (M_R \times M_{\tilde{R}}) \times M_X & \longrightarrow & M'_{\hat{X}} \\ & (r, \tilde{r}),\ x & \longmapsto & \tilde{h}(\tilde{f}(r,\tilde{r}), x) \end{array}$$

eine universelle Hashfunktion. Und wegen

$$\tilde{h}(\tilde{f}(r,\tilde{r}), x) = f_{\tilde{r}}(h(f(r), x)) = f_{\tilde{r}}(h_f(r), x)$$

ist auch

$$\begin{array}{rccc} \tilde{h}_f : & (M_R \times M_{\tilde{R}}) \times M_X & \longrightarrow & M'_{\hat{X}} \\ & (r, \tilde{r}),\ x & \longmapsto & f_{\tilde{r}}(h_f(r), x) \end{array}$$

eine solche. Damit ist nach Definition 3.4 h_f eine universelle Quasi-Hashfunktion. ■

A.4 Zu Kapitel 4

A.4.1 Lemma 4.2

Lemma 4.2: Komposition von Symbolgewichtungen
Es sei $\mathcal{X}$ eine Familie von Quellen und $\alpha, x_1, \ldots, x_n \in \Sigma^*_{\mathcal{X}}$. Dann ist

$$\eta^{\mathcal{X}}(\alpha; x_1 \ldots x_n) \geq \sum_{\nu=1}^{n} \eta^{\mathcal{X}}(\alpha x_1 \ldots x_{\nu-1}; x_\nu).$$

Ist eine Seite dieser Ungleichung definiert (d. h. nicht $\perp$), so ist es auch die andere. □

Beweis: Sind alle $\eta^{\mathcal{X}}(\alpha x_1 \dots x_{\nu-1};\, x_\nu)$ $(\nu = 1, \dots, n)$ definiert, so folgt aus folgender Rechnung die zu beweisende Gleichung und die Definiertheit von $\eta^{\mathcal{X}}(\alpha;\, x_1 \dots x_n)$:

$$\begin{aligned}
&\eta^{\mathcal{X}}(\alpha; x_1 \dots x_n)\\
&\quad = -\log \sup_{X \in \mathcal{X}} P(X_{|\alpha|+1} \dots X_{|\alpha|+|x_1 \dots x_n|} = x_1 \dots x_n \mid X_1 \dots X_{|\alpha|} = \alpha)\\
&\quad = -\log \sup_{X \in \mathcal{X}} \prod_{\nu=1}^{n} P(X_{|\alpha x_1 \dots x_{\nu-1}|+1} \dots X_{|\alpha x_1 \dots x_{\nu-1}|+|x_\nu|} = x_\nu \mid X_1 \dots X_{|\alpha x_1 \dots x_{\nu-1}|} = \alpha x_1 \dots x_{\nu-1})\\
&\quad \geq -\log \prod_{\nu=1}^{n} \sup_{X \in \mathcal{X}} P(X_{|\alpha x_1 \dots x_{\nu-1}|+1} \dots X_{|\alpha x_1 \dots x_{\nu-1}|+|x_\nu|} = x_\nu \mid X_1 \dots X_{|\alpha x_1 \dots x_{\nu-1}|} = \alpha x_1 \dots x_{\nu-1})\\
&\quad = \sum_{\nu=1}^{n} \eta^{\mathcal{X}}(\alpha x_1 \dots x_{\nu-1}; x_\nu).
\end{aligned}$$

Ist $\eta^{\mathcal{X}}(\alpha; x_1)$ undefiniert, so sind dies auch alle anderen Vorkommnisse von $\eta^{\mathcal{X}}$ in der zu beweisenden Ungleichung, somit sind beide Seiten undefiniert.

Ist $\eta^{\mathcal{X}}(\alpha; x_1)$ definiert, aber ein $\eta^{\mathcal{X}}(\alpha x_1 \dots x_{\nu-1};\, x_\nu) = \perp$, so sei $\nu_0 > 1$ der kleinste Index mit dieser Eigenschaft. Dann ist

$$P(X_1 \dots X_{|\alpha x_1 \dots x_{\nu_0-1}|} = \alpha x_1 \dots x_{\nu_0-1}) = 0, \qquad P(X_1 \dots X_{|\alpha|} = \alpha) > 0, \tag{27}$$

also

$$\eta^{\mathcal{X}}(\alpha x_1 \dots x_{\nu-1};\, x_\nu) \begin{cases} \neq \perp, & \nu < \nu_0 - 1,\\ = \infty, & \nu = \nu_0 - 1,\\ = \perp, & \nu > \nu_0 - 1, \end{cases}$$

womit sich

$$\sum_{\nu=1}^{n} \eta^{\mathcal{X}}(\alpha x_1 \dots x_{\nu-1};\, x_\nu) = \infty \tag{28}$$

ergibt.

Aus (27) ergibt sich

$$\eta^{\mathcal{X}}(\alpha;\, x_1 \dots x_n) = \infty,$$

zusammen mit (28) folgt daraus die Ungleichung und die Definiertheit beider Seiten derselben.

A.4.2 Bemerkung Seite 22

Definition 4.5: Konditioniert links-zeitinvariante Familien von Quellen
Es sei $X^{(n)}$ wie in Definition 4.3.

Eine Familie $\mathcal{X}$ von Quellen heißt *konditioniert links-zeitinvariant*, wenn für jedes $X \in \mathcal{X}$, jedes $n \in \mathbb{N}_0$ und jedes $x \in \Sigma_{\mathcal{X}}^n$ mit $P(X_1 \dots X_n = x) > 0$ auch

$$X^{(n)}|(X_1 \dots X_n = x) \in \mathcal{X}$$

gilt. □

Auf Seite 22 wurde angemerkt, daß es in den Definitionen von Links-Zeitinvarianz (Definition 4.3), Rechts-Zeitinvarianz (Definition 4.4) und konditionierter Links-Zeitinvarianz (Definition 4.5) genügt, eine Erfüllung der Bedingungen für den Fall $n = 1$ zu fordern (statt $n \in \mathbb{N}_0$). Für Links- und Rechts-Zeitinvarianz ist dies offensichtlich, für konditionierte Links-Zeitinvarianz zeigt man es wie folgt:

Beweis: Es sei $\mathcal{X}$ eine Familie von Quellen, so daß für jedes $X \in \mathcal{X}$ gilt: Ist $x \in \Sigma_{\mathcal{X}}$ mit $P(X_1 = x) > 0$, so ist auch

$$X^{(1)}|(X_1 = x) \in \mathcal{X}. \tag{29}$$

Zu zeigen ist nun, daß für $n \in \mathbb{N}_0$ und $x \in \Sigma_{\mathcal{X}}^n$ mit $P(X_1 \dots X_n = x_1 \dots x_n) > 0$ auch

$$X^{(n)}|(X_1 \dots X_n = x_1 \dots x_n) \in \mathcal{X}$$

gilt.

Hierzu bedienen wir uns einer Induktion über n. Der Induktionsanfang $n = 0$ ist klar. Sei $n > 0$. Aufgrund der Induktionsvoraussetzung gilt

$$Y := X^{(n-1)}|(X_1 \dots X_{n-1} = x_1 \dots x_{n-1}) \in \mathcal{X}$$

und somit ist

$$X^{(n)}|(X_1 \dots X_n = x_1 \dots x_n) = Y^{(1)}|(Y_1 = x_n) \overset{(29)}{\in} \mathcal{X},$$

da wegen $P(X_1 \dots X_n = x_1 \dots x_n) > 0$ auch

$$P(Y_1 = x_n) = P(X_n = x_n \mid X_1 \dots X_{n-1} = x_1 \dots x_{n-1}) > 0$$

gilt. ■

A.4.3 Bemerkung Seite 22

Auf Seite 22 wurde weiterhin angemerkt, daß für jede Teilmenge der drei Eigenschaften Links-Zeitinvarianz, Rechts-Zeitinvarianz und konditionierter Links-Zeitinvarianz eine Familie von Quellen existiert, die genau diese Teilmenge erfüllt. Dies soll hier mit Beispielen belegt werden.

Es seien $\Sigma := \{0, 1, 2, 3, 4\}$ und die Zufallsvariablen $A_i, B_i, C, D_i, E, E_a, E_b$ $(i \geq 0)$ mit Werten in $\Sigma^{\mathbb{N}}$ wie folgt:

- A_i konstant $2^i 0^\infty$,
- B_i konstant $3^i 1^\infty$,
- C gleichverteilt auf $\{0^\infty, 1^\infty\}$ und
- D_i gleichverteilt auf $\{4^i 20^\infty, 4^i 31^\infty\}$.

Dann gilt:

Familie	links-zeitinv.	rechts-zeitinv.	kond. links-zeitinv.
$\{A_1\}$	nein	nein	nein
$\{D_0, A_0, B_0\}$	nein	nein	ja
$\{A_i : i \geq 1\}$	nein	ja	nein
$\{A_0, B_0, D_i : i \geq 0\}$	nein	ja	ja
$\{C, D_0\}$	ja	nein	nein
$\{A_0, A_1\}$	ja	nein	ja
$\{C\}$	ja	ja	nein
$\{B_0\}$	ja	ja	ja

Zum Überprüfen helfen die folgenden Gleichungen (mit $i \geq 1$ und $X^{(1)}$ wie in Definition 4.3):

$$A_0^{(1)} = A_0, \quad A_i^{(1)} = A_{i-1}, \quad B_0^{(1)} = B_0, \quad B_i^{(1)} = B_{i-1},$$
$$C^{(1)} = C, \quad E^{(1)} = E, \quad D_0^{(1)} = C, \quad D_i^{(1)} = D_{i-1},$$

und

$$\begin{aligned} A_0^{(1)}|(A_{0,1}=0) &= A_0, & B_0^{(1)}|(B_{0,1}=1) &= B_0,\\ D_0^{(1)}|(D_{0,1}=2) &= A_0, & D_0^{(1)}|(D_{0,1}=3) &= B_0,\\ A_i^{(1)}|(A_{i,1}=2) &= A_{i-1}, & D_i^{(1)}|(D_{i,1}=4) &= D_{i-1},\\ C^{(1)}|(C_1=0) &= A_0, & C^{(1)}|(C_1=1) &= B_0. \end{aligned}$$ ■

A.4.4 Lemma 4.6

Lemma 4.6: Verschiebung von Symbolgewichtungen
Es sei $\mathcal{X}$ eine Familie von Quellen, $\alpha_1, \alpha_2, x \in \Sigma_{\mathcal{X}}^*$ und $n \in \mathbb{N}_0$.
Ist $\mathcal{X}$ konditioniert links-zeitinvariant, so gilt, falls $\eta^{\mathcal{X}}(\alpha_1\alpha_2; x) \neq \perp$:

$$\eta^{\mathcal{X}}(\alpha_2; x) \leq \eta^{\mathcal{X}}(\alpha_1\alpha_2; x). \tag{5}$$

Ist $\mathcal{X}$ rechts-zeitinvariant, so gilt für $\eta^{\mathcal{X}}(\alpha_2; x) \neq \perp$:

$$\eta^{\mathcal{X}}(\alpha_2; x) \geq \min_{\alpha \in \Sigma_{\mathcal{X}}^n} \eta^{\mathcal{X}}(\alpha\alpha_2; x). \tag{6}$$

Ist $\mathcal{X}$ rechts-zeitinvariant und konditioniert links-zeitinvariant, so gilt in (6) sogar Gleichheit. □

Beweis: Zunächst zeigen wir (5). Wir können o. B. d. A. $|\alpha_1| = 1$ annehmen. Sei $X \in \mathcal{X}$ beliebig mit $P(X_1 \dots X_{|\alpha_1\alpha_2|} = \alpha_1\alpha_2) > 0$ und $Y := X^{(1)}|(X_1 = \alpha_1) \in \mathcal{X}$. Dann gilt

$$\begin{aligned} &P(X_{|\alpha_1\alpha_2|+1} \dots X_{|\alpha_1\alpha_2 x|} = x \mid X_1 \dots X_{|\alpha_1\alpha_2|} = \alpha_1\alpha_2)\\ &\quad = P(X^{(1)}_{|\alpha_2|+1} \dots X^{(1)}_{|\alpha_2 x|} = x \mid X^{(1)}_1 \dots X^{(1)}_{|\alpha_2|} = \alpha_2,\ X_1 = \alpha_1)\\ &\quad = P(Y_{|\alpha_2|+1} \dots Y_{|\alpha_2 x|} = x \mid Y_1 \dots Y_{|\alpha_2|} = \alpha_2)\\ &\quad \leq \sup_{X' \in \mathcal{X}} P(X'_{|\alpha_2|+1} \dots X'_{|\alpha_2 x|} = x \mid X'_1 \dots X'_{|\alpha_2|} = \alpha_2) \end{aligned}$$

Daraus ergibt sich

$$\begin{aligned} \eta^{\mathcal{X}}(\alpha_1\alpha_2; x) &= -\log \sup_{X \in \mathcal{X}} P(X_{|\alpha_1\alpha_2|+1} \dots X_{|\alpha_1\alpha_2 x|} = x \mid X_1 \dots X_{|\alpha_1\alpha_2|} = \alpha_1\alpha_2)\\ &\overset{\text{s.o.}}{\geq} -\log \sup_{X \in \mathcal{X}} \sup_{X' \in \mathcal{X}} P(X'_{|\alpha_2|+1} \dots X'_{|\alpha_2 x|} = x \mid X'_1 \dots X'_{|\alpha_2|} = \alpha_2)\\ &= -\log \sup_{X' \in \mathcal{X}} P(X'_{|\alpha_2|+1} \dots X'_{|\alpha_2 x|} = x \mid X'_1 \dots X'_{|\alpha_2|} = \alpha_2)\\ &= \eta^{\mathcal{X}}(\alpha_2; x), \end{aligned}$$

womit (5) gezeigt wäre.

Um (6) zu zeigen, können wir uns wieder o. B. d. A. auf den Fall $n = 1$ beschränken. Sei diesmal $X \in \mathcal{X}$ beliebig mit $P(X_1 \dots X_{|\alpha_2|} = \alpha_2) > 0$. Weil $\mathcal{X}$ rechts-zeitinvariant ist, existiert ein $X^{(-1)}$ mit $(X^{(-1)})^{(1)} = X$. Wir erhalten dann:

$$\begin{aligned} &P(X_{|\alpha_2|+1} \dots X_{|\alpha_2 x|} = x \mid X_1 \dots X_{|\alpha_2|} = \alpha_2)\\ &\quad = \sum_{\alpha \in \Sigma_{\mathcal{X}}} P(X_1^{(-1)} = \alpha,\ X_{|\alpha_2|+1} \dots X_{|\alpha_2 x|} = x \mid X_1 \dots X_{|\alpha_2|} = \alpha_2)\\ &\quad = \sum_{\alpha \in \Sigma_{\mathcal{X}}} P(X_1^{(-1)} = \alpha \mid X_1 \dots X_{|\alpha_2|} = \alpha_2) \cdot P(X_{|\alpha_2|+1} \dots X_{|\alpha_2 x|} = x \mid X_1^{(-1)} = \alpha,\ X_1 \dots X_{|\alpha_2|} = \alpha_2)\\ &\quad \leq \max_{\alpha \in \Sigma_{\mathcal{X}}} P(X_{|\alpha_2|+1} \dots X_{|\alpha_2 x|} = x \mid X_1^{(-1)} = \alpha,\ X_1 \dots X_{|\alpha_2|} = \alpha_2)\\ &\quad = \max_{\alpha \in \Sigma_{\mathcal{X}}} P(X^{(-1)}_{|\alpha\alpha_2|+1} \dots X^{(-1)}_{|\alpha\alpha_2 x|} = x \mid X_1^{(-1)} \dots X^{(-1)}_{|\alpha\alpha_2|} = \alpha\alpha_2) \end{aligned}$$

Daraus ergibt sich

$$\begin{aligned}
\eta^{\mathcal{X}}(\alpha_2;x) &= -\log \sup_{X\in\mathcal{X}} P(X_{|\alpha_2|+1}\dots X_{|\alpha_2 x|} = x \mid X_1\dots X_{|\alpha_2|} = \alpha_2) \\
&\overset{\text{s.o.}}{\geq} -\log \sup_{X\in\mathcal{X}} \max_{\alpha\in\Sigma_{\mathcal{X}}} P(X^{(-1)}_{|\alpha\alpha_2|+1}\dots X^{(-1)}_{|\alpha\alpha_2 x|} = x \mid X^{(-1)}_1\dots X^{(-1)}_{|\alpha\alpha_2|} = \alpha\alpha_2) \\
&\geq -\log \sup_{X'\in\mathcal{X}} \max_{\alpha\in\Sigma_{\mathcal{X}}} P(X'_{|\alpha\alpha_2|+1}\dots X'_{|\alpha\alpha_2 x|} = x \mid X'_1\dots X'_{|\alpha\alpha_2|} = \alpha\alpha_2) \\
&= \max_{\alpha\in\Sigma_{\mathcal{X}}} \eta^{\mathcal{X}}(\alpha\alpha_2;x),
\end{aligned}$$

also ist (6) wahr.

Ist $\mathcal{X}$ rechts-zeitinvariant und konditioniert links-zeitinvariant, so gelten (5) und (6) zugleich, es folgt

$$\eta^{\mathcal{X}}(\alpha_2;x) \overset{(6)}{\geq} \min_{\alpha\in\Sigma^n_{\mathcal{X}}} \eta^{\mathcal{X}}(\alpha\alpha_2;x) \overset{(5)}{\geq} \min_{\alpha\in\Sigma^n_{\mathcal{X}}} \eta^{\mathcal{X}}(\alpha_2;x) = \eta^{\mathcal{X}}(\alpha_2;x),$$

also gilt in diesem Fall Gleichheit in (6). ■

A.4.5 Satz 4.8

Definition 4.7: Adaptiver Hash-Extraktor $\Xi^{n,m}_{\eta,h}$

Es sei η eine Symbolgewichtung, $n \in \mathbb{N}$, $m : \mathbb{R}_{\geq 0} \cup \{\infty\} \to \mathbb{N}_0$, weiter M_R, Σ und Σ_{out} endliche, nichtleere Mengen, und h eine Familie von Funktionen

$$h_{\tilde{m}} : M_R \times \Sigma^n \to \Sigma^{\tilde{m}}_{\text{out}} \qquad (\tilde{m} \in \mathcal{M} := m(\mathbb{R}_{\geq 0} \cup \{\infty\}) \setminus \{0\}).$$

Dann ist der *adaptive Hash-Extraktor*

$$\Xi^{n,m}_{\eta,h} : \Sigma^* \cup \Sigma^{\mathbb{N}} \longrightarrow \Sigma^*_{\text{out}} \cup \Sigma^{\mathbb{N}}_{\text{out}}$$

durch folgende Konstruktion definiert:

Sei $X \in \Sigma^* \cup \Sigma^{\mathbb{N}}$ und $R \in M_R$, sowie

$$B_i := \begin{cases} X_{(i-1)n+1}\dots X_{in}, & \text{falls } |X| \geq in, \\ \perp, & \text{sonst,} \end{cases}$$

$$\hat{X}_i := \begin{cases} h_{m(\eta(B_1\dots B_{i-1};\, B_i))}(R, B_i), & \text{falls } B_i \neq \perp,\ \eta(B_1\dots B_{i-1};\, B_i) \neq \perp \\ & \text{und } m(\eta(B_1\dots B_{i-1};\, B_i)) > 0, \\ \lambda, & \text{sonst,} \end{cases}$$

und schließlich

$$\Xi^{n,m}_{\eta,h}(R, X) := \hat{X}_1 \hat{X}_2 \dots$$ □

Satz 4.8: Adaptive Extraktion

Es sei $\mathcal{X}$ eine Familie von Quellen über Σ, $\eta \leq \eta^{\mathcal{X}}$ eine Symbolgewichtung über Σ, $l \in \mathbb{N}$, $n \in \mathbb{N}$, $m : \mathbb{R}_{\geq 0} \cup \{\infty\} \to \{0, \dots, n\}$, weiter M_R, M_S, Σ_{out} endliche, nichtleere Mengen und h eine Familie von universellen Quasi-Hashfunktionen

$$h_{\tilde{m}} : M_R \times \Sigma^n \to \Sigma^{\tilde{m}}_{\text{out}} \qquad (\tilde{m} \in \mathcal{M} := m(\mathbb{R}_{\geq 0} \cup \{\infty\}) \setminus \{0\}).$$

Außerdem sei R eine auf M_R gleichverteilte und von X, S unabhängige Zufallsvariable, S eine Zufallsvariable mit Werten in M_S und $X \in \mathcal{X}$.

Seien ferner

$$\hat{X} := \Xi^{n,m}_{\eta,h}(R, X_1 \dots X_l)$$

und

$$\log\varepsilon := \tfrac{1}{2}\sup_{\substack{k\in\mathbb{R}_{\geq 0}\\ m(k)\neq 0}}(m(k)\log\#\Sigma_{\mathrm{out}}-k)+\log(l+1)+\log\lfloor l/n\rfloor+\log\#M_S+\tfrac{1}{2}\log\#\mathcal{M}-1 \tag{7}$$

Dann ist $P(|\hat{X}|\leq l)=1$, und $\hat{X}$ ist ε-zufällig unter Kenntnis von $R, S, |\hat{X}|$. □

Beweis: Da wir nur $X_1\dots X_l$ verwenden, können wir o. B. d. A. annehmen, daß $|X|\leq l$, also $X_1\dots X_l = X$.

Es seien $\hat{X}_i$, B_i wie in Definition 4.7. Allerdings handelt es sich nun um Zufallsvariablen, da auch X und R welche sind.

Setze darüber hinaus $\mathcal{M} := m(\mathbb{R}_{\geq 0})\setminus\{0\}$, sowie $\tilde{k}(\tilde{m}) := \inf\{k\in\mathbb{R}_{\geq 0} : m(k)=\tilde{m}\}$ $(\tilde{m}\in\mathcal{M})$ und

$$M_i := \begin{cases} m(\eta(B_1\dots B_{i-1};\,B_i)), & \text{falls } B_i\neq\perp \text{ und } \eta(B_1\dots B_{i-1};\,B_i)\neq\perp,\\ 0, & \text{sonst.}\end{cases}$$

Da $\max m(\mathbb{R}_{\geq 0})\leq n$, ist $L := |\hat{X}| = \sum_{i\in\mathbb{N}}|\hat{X}_i| \leq \sum_{i\in\mathbb{N}}|B_i| = |X|\leq l$.

Wir setzen abkürzend:

$$\begin{aligned} C_{i,\alpha,\tilde{m}} &:= (M_i=\tilde{m},\ B_1\dots B_{i-1}=\alpha) && (i\in\mathbb{N},\alpha\in\Sigma^{n(i-1)},\tilde{m}\in\mathcal{M}\cup\{0\}),\\ \delta_{i,\alpha,\tilde{m}} &:= P(M_i=\tilde{m}\mid B_1\dots B_{i-1}=\alpha) && (i\in\mathbb{N},\alpha\in\Sigma^{n(i-1)},\tilde{m}\in\mathcal{M}\cup\{0\}).\end{aligned}$$

Mit $i\in\mathbb{N}$, $\alpha\in\Sigma^{n(i-1)}$, $\tilde{m}\in\mathcal{M}$ rechnen wir nun weiter:

Im Falle $\eta(\alpha;x)<\tilde{k}(\tilde{m})$ gilt $m(\eta(\alpha;x))\neq\tilde{m}$, also $P(B_i=x,\ B_1\dots B_{i-1}=\alpha,\ M_i=\tilde{m})=0$, und somit ist, falls definiert,

$$P(B_i=x,\ M_i=\tilde{m}\mid B_1\dots B_{i-1}=\alpha)=0, \tag{30}$$

und im Falle $\eta(\alpha;x)\geq\tilde{k}(\tilde{m})$ ist

$$\begin{aligned} &P(B_i=x,\ M_i=\tilde{m}\mid B_1\dots B_{i-1}=\alpha)\\ &\quad\leq P(B_i=x\mid B_1\dots B_{i-1}=\alpha)\leq 2^{-\eta^{\mathcal{X}}(\alpha;x)}\leq 2^{-\eta(\alpha;x)}\leq 2^{-\tilde{k}(\tilde{m})}, \end{aligned}\tag{31}$$

falls definiert.

Dann ist für $P(C_{i,\alpha,\tilde{m}})>0$, $\tilde{m}\in\mathcal{M}$:

$$\begin{aligned} H_\infty(B_i\mid C_{i,\alpha,\tilde{m}}) &= -\max_{x\in\Sigma^n}\log P(B_i=x\mid M_i=\tilde{m},\ B_1\dots B_{i-1}=\alpha)\\ &= -\max_{x\in\Sigma^n}\log\delta_{i,\alpha,\tilde{m}}^{-1}P(B_i=x,\ M_i=\tilde{m}\mid B_1\dots B_{i-1}=\alpha)\\ &= \log\delta_{i,\alpha,\tilde{m}}-\max_{x\in\Sigma^n}\log P(B_i=x,\ M_i=\tilde{m}\mid B_1\dots B_{i-1}=\alpha)\\ &\overset{(30,31)}{\geq}\tilde{k}(\tilde{m})+\log\delta_{i,\alpha,\tilde{m}}. \end{aligned}\tag{32}$$

Für $\tilde{m}\in\mathcal{M}\cup\{0\}$ seien $U_{i,\tilde{m}}$ auf $\Sigma_{\mathrm{out}}^{\tilde{m}}$ gleichverteilte Zufallsvariablen, wobei $U_{i,\tilde{m}}$ unabhängig sei von R, S, L, B_i, X_i, $\hat{X}_i$, M_i, $U_{i',\tilde{m}'}$ $((i',\tilde{m}')\neq(i,\tilde{m}))$.

Ist $P(C_{i,\alpha,\tilde{m}})>0$, $\tilde{m}\in\mathcal{M}$, so gilt mit dem nach $C_{i,\alpha,\tilde{m}}$ konditionierten Wahrscheinlichkeitsmaß:

- Es sind (B_i,S,L), R und $U_{i,\tilde{m}}$ unabhängig,
- es ist $\hat{X}_i = h_{\tilde{m}}(R,B_i)$ und somit $\hat{X}_i\in\Sigma_{\mathrm{out}}^{\tilde{m}}$,
- $U_{i,\tilde{m}}$ ist gleichverteilt auf $\Sigma_{\mathrm{out}}^{\tilde{m}}$,
- R ist gleichverteilt auf M_R,
- es gilt $H_{\mathrm{Ren}}(B_i)\overset{2.6}{\geq}H_\infty(B_i)\overset{(32)}{\geq}\tilde{k}(\tilde{m})+\log\delta_{i,\alpha,\tilde{m}}$,

- da $L \leq l$, nimmt (S, L) nur Werte aus $M_S \times \{0, \ldots, l\}$ an,
- und schließlich ist $h_{\tilde{m}} : \Sigma^n \to \Sigma_{\text{out}}^{\tilde{m}}$ eine universelle Quasi-Hashfunktion.

Somit kann hier das Leftover Hash Lemma (Satz 3.6) angewandt werden, und wir erhalten mit $b := \lfloor \frac{l}{n} \rfloor$ im Falle $\tilde{m} \in \mathcal{M}$:

$$\begin{aligned}
\mathrm{SD}(S, L, R, \hat{X}_i;\ S, L, R, U_{i,\tilde{m}}) &= \mathrm{SD}((S, L), R, h_{\tilde{m}}(R, B_i);\ (S, L), R, U_{i,\tilde{m}}) \\
&\overset{3.6}{\leq} \tfrac{1}{2} \#M_S(l+1) \sqrt{\#\Sigma_{\text{out}}^{\tilde{m}} \cdot 2^{-\tilde{k}(\tilde{m}) - \log \delta_{i,\alpha,\tilde{m}}}} \\
&\leq \varepsilon\, b^{-1}\, \#\mathcal{M}^{-1/2}\, \delta_{i,\alpha,\tilde{m}}^{-1/2} \frac{\exp_2\big(\frac{1}{2}(\tilde{m} \log \#\Sigma_{\text{out}} - \tilde{k}(\tilde{m}))\big)}{\exp_2\big(\frac{1}{2} \sup_{\substack{k \in \mathbb{R}_{\geq 0} \\ m(k) \neq 0}} (m(k) \log \#\Sigma_{\text{out}} - k)\big)}
\end{aligned}$$

Da

$$\sup_{\substack{k \in \mathbb{R}_{\geq 0} \\ m(k) \neq 0}} (m(k) \log \#\Sigma_{\text{out}} - k) \ \geq \lim_{\substack{k \to \tilde{k}(\tilde{m}) \\ k \in m^{-1}(\{\tilde{m}\}) \\ k \geq \tilde{k}(\tilde{m})}} (m(k) \log \#\Sigma_{\text{out}} - k) \ = \ \tilde{m} \log \#\Sigma_{\text{out}} - \tilde{k}(\tilde{m}),$$

folgt

$$\mathrm{SD}(S, L, R, \hat{X}_i;\ S, L, R, U_{i,\tilde{m}}) \leq \varepsilon\, b^{-1}\, \#\mathcal{M}^{-1/2}\, \delta_{i,\alpha,\tilde{m}}^{-1/2}. \tag{33}$$

Im ursprünglichen Wahrscheinlichkeitsraum (ohne Konditionierung nach $C_{i,\alpha,\tilde{m}}$) lautet (33):

$$\mathrm{SD}(S, L, R, \hat{X}_i;\ S, L, R, U_{i,\tilde{m}} \parallel C_{i,\alpha,\tilde{m}}) \leq \varepsilon\, b^{-1}\, \#\mathcal{M}^{-1/2}\, \delta_{i,\alpha,\tilde{m}}^{-1/2}. \tag{34}$$

Für $P(B_1 \ldots B_{i-1} = \alpha) > 0$ gilt dann weiter

$$\begin{aligned}
&\mathrm{SD}(S, L, R, \hat{X}_i;\ S, L, R, U_{i,M_i} \parallel B_1 \ldots B_{i-1} = \alpha) \\
&\quad \overset{2.9(1)}{\leq} \mathrm{SD}(S, L, R, \hat{X}_i, M_i;\ S, L, R, U_{i,M_i}, M_i \parallel B_1 \ldots B_{i-1} = \alpha) \\
&\quad \overset{2.9(4)}{=} \sum_{\tilde{m} \in \mathcal{M}} P(M_i = \tilde{m} \mid B_1 \ldots B_{i-1} = \alpha)\ \mathrm{SD}(S, L, R, \hat{X}_i;\ S, L, R, U_{i,\tilde{m}} \parallel C_{i,\alpha,\tilde{m}}) \\
&\qquad\qquad + P(M_i = 0 \mid B_1 \ldots B_{i-1} = \alpha)\ \mathrm{SD}(S, L, R, \lambda;\ S, L, R, \lambda \parallel C_{i,\alpha,0}) \\
&\quad = \sum_{\tilde{m} \in \mathcal{M}} \delta_{i,\alpha,\tilde{m}}\ \mathrm{SD}(S, L, R, \hat{X}_i;\ S, L, R, U_{i,\tilde{m}} \parallel C_{i,\alpha,\tilde{m}}) + P(M_i = 0 | B_1 \ldots B_{i-1} = \alpha) \cdot 0 \\
&\quad \overset{(34)}{\leq} \sum_{\tilde{m} \in \mathcal{M}} \delta_{i,\alpha,\tilde{m}}\, \varepsilon\, b^{-1}\, \#\mathcal{M}^{-1/2}\, \delta_{i,\alpha,\tilde{m}}^{-1/2} \\
&\quad = \varepsilon\, b^{-1}\, \#\mathcal{M}^{-1/2} \sum_{\tilde{m} \in \mathcal{M}} \big(\delta_{i,\alpha,\tilde{m}}^{1/2} \cdot 1\big) \\
&\quad \overset{\text{CSU}}{\leq} \varepsilon\, b^{-1}\, \#\mathcal{M}^{-1/2} \sqrt{\sum_{\tilde{m} \in \mathcal{M}} \delta_{i,\alpha,\tilde{m}}} \sqrt{\sum_{\tilde{m} \in \mathcal{M}} 1} \\
&\quad \leq \varepsilon\, b^{-1}\, \#\mathcal{M}^{-1/2} \cdot 1 \cdot \sqrt{\#\mathcal{M}} \\
&\quad = \varepsilon\, b^{-1}.
\end{aligned} \tag{35}$$

Hierbei bezeichnet CSU die Cauchy-Schwarzsche Ungleichung.

Wie setzen nun $U := U_{1,M_1} U_{2,M_2} \ldots$, dann ist U nach Lemma 2.12 perfekt zufällig unter Kenntnis von $R, S, |\hat{X}|$. Können wir noch

$$\mathrm{SD}(R, S, L, \hat{X}_i \ldots \hat{X}_b;\ R, S, L, U_{i,M_i} \ldots U_{b,M_b} \parallel B_1 \ldots B_{i-1} = \alpha) \leq (b - i + 1)\, \varepsilon\, b^{-1} \tag{36}$$

für $i = 1, \ldots, b$ und alle $\alpha \in \Sigma^{(i-1)n}$ mit $P(B_1 \ldots B_{i-1} = \alpha) > 0$ zeigen, so folgt

$$\begin{aligned}
\mathrm{SD}(R, S, |\hat{X}|, \hat{X};\ R, S, |\hat{X}|, U) &= \mathrm{SD}(R, S, L, \hat{X}_1 \hat{X}_2 \ldots;\ R, S, L, U_{1,M_1} U_{2,M_2} \ldots) \\
&= \mathrm{SD}(R, S, L, \hat{X}_1 \ldots \hat{X}_b;\ R, S, L, U_{1,M_1} \ldots U_{b,M_b}) \\
&\overset{(36)}{\leq} (b - 1 + 1)\, \varepsilon\, b^{-1} = \varepsilon,
\end{aligned}$$

und der Satz ist bewiesen.

Also wollen wir zuletzt (36) mit vollständiger Induktion über absteigendes i zeigen.

Für den Induktionsanfang $i = b$ lautet (36):

$$\mathrm{SD}(R, S, L, \hat{X}_b;\ R, S, L, U_{b,M_b} \parallel B_1 \dots B_{b-1} = \alpha) \leq \varepsilon\, b^{-1},$$

was genau die Aussage von (35) mit $i := b$ ist.

Es sei nun $1 \leq i < b$ und wir nehmen an, (36) sei wahr für $i + 1$.

Wir setzen abkürzend $\tilde{U} := U_{i+1,M_{i+1}} \dots U_{b,M_b}$, dann ist $\tilde{U}$ nach Lemma 2.12 perfekt zufällig unter Kenntnis von R, S, L, $\hat{X}_\nu$, M_ν, B_i, $U_{\nu,\tilde{m}}$ $(\nu \leq i)$.

Weiter definieren wir neue Zufallsvariablen $U_{(\mu)}$, gleichverteilt auf $\Sigma^\mu_{\mathrm{out}}$ und unabhängig von R, S, L, $\hat{X}_\nu$, M_ν, $\tilde{U}$, $U_{\nu,\tilde{m}}$, B_ν, und erhalten

$$\begin{aligned}
&\mathrm{SD}(R, S, L, \hat{X}_i \hat{X}_{i+1} \dots \hat{X}_b;\ R, S, L, \hat{X}_i \tilde{U} \parallel B_1 \dots B_{i-1} = \alpha) \\
&\quad \overset{2.9(1)}{\leq} \mathrm{SD}(R, S, L, B_i, \hat{X}_{i+1} \dots \hat{X}_b;\ R, S, L, B_i, \tilde{U} \parallel B_1 \dots B_{i-1} = \alpha) \\
&\quad \overset{2.9(4)}{=} \sum_{x \in \Sigma^n} \mathrm{SD}(R, S, L, \hat{X}_{i+1} \dots \hat{X}_b;\ R, S, L, \tilde{U} \parallel B_1 \dots B_{i-1} B_i = \alpha x) \\
&\quad \overset{(36)}{\leq} (b - i)\, \varepsilon\, b^{-1},
\end{aligned} \tag{37}$$

sowie

$$\begin{aligned}
&\mathrm{SD}(R, S, L, \hat{X}_i \tilde{U};\ R, S, L, U_{i,M_i} \tilde{U} \parallel B_1 \dots B_{i-1} = \alpha) \\
&\quad \overset{2.9(1)}{\leq} \mathrm{SD}(R, S, L, \hat{X}_i, \tilde{U}, |\tilde{U}|;\ R, S, L, U_{i,M_i}, \tilde{U}, |\tilde{U}| \parallel B_1 \dots B_{i-1} = \alpha) \\
&\quad \overset{2.9(4)}{=} \sum_{\mu \in \mathbb{N}_0} P(|\tilde{U}| = \mu \mid B_1 \dots B_{i-1} = \alpha) \\
&\qquad\qquad \mathrm{SD}(R, S, L, \hat{X}_i, \tilde{U};\ R, S, L, U_{i,M_i}, \tilde{U} \parallel |\tilde{U}| = \mu,\ B_1 \dots B_{i-1} = \alpha) \\
&\quad = \sum_{\mu \in \mathbb{N}_0} P(|\tilde{U}| = \mu \mid B_1 \dots B_{i-1} = \alpha) \\
&\qquad\qquad \mathrm{SD}(R, S, L, \hat{X}_i, U_{(\mu)};\ R, S, L, U_{i,M_i}, U_{(\mu)} \parallel |\tilde{U}| = \mu,\ B_1 \dots B_{i-1} = \alpha) \\
&\quad \overset{2.9(2)}{=} \sum_{\mu \in \mathbb{N}_0} P(|\tilde{U}| = \mu \mid B_1 \dots B_{i-1} = \alpha) \\
&\qquad\qquad \mathrm{SD}(R, S, L, \hat{X}_i;\ R, S, L, U_{i,M_i} \parallel |\tilde{U}| = \mu,\ B_1 \dots B_{i-1} = \alpha) \\
&\quad \overset{2.9(4)}{=} \mathrm{SD}(R, S, L, \hat{X}_i, |\tilde{U}|;\ R, S, L, U_{i,M_i}, |\tilde{U}| \parallel B_1 \dots B_{i-1} = \alpha) \\
&\quad = \mathrm{SD}(R, S, L, \hat{X}_i, (L - |\hat{X}_i| - c_\alpha); \\
&\qquad\qquad R, S, L, U_{i,M_i}, (L - |U_{i,M_i}| - c_\alpha) \parallel B_1 \dots B_{i-1} = \alpha) \qquad \text{für geeignete } c_\alpha \in \mathbb{N}_0 \\
&\quad \overset{2.9(1)}{\leq} \mathrm{SD}(R, S, L, \hat{X}_i;\ R, S, L, U_{i,M_i} \parallel B_1 \dots B_{i-1} = \alpha) \\
&\quad \overset{(35)}{\leq} \varepsilon\, b^{-1},
\end{aligned} \tag{38}$$

und schließlich

$$\begin{aligned}
&\mathrm{SD}(R, S, L, \hat{X}_i \dots \hat{X}_b;\ R, S, L, U_{i,M_i} \dots U_{b,M_b} \parallel B_1 \dots B_{i-1} = \alpha) \\
&\quad \overset{2.9(3)}{\leq} \mathrm{SD}(R, S, L, \hat{X}_i \hat{X}_{i+1} \dots \hat{X}_b;\ R, S, L, \hat{X}_i \tilde{U} \parallel B_1 \dots B_{i-1} = \alpha) \\
&\qquad + \mathrm{SD}(R, S, L, \hat{X}_i \tilde{U};\ R, S, L, U_{i,M_i} \tilde{U} \parallel B_1 \dots B_{i-1} = \alpha) \\
&\quad \overset{(37,38)}{\leq} (b - i)\, \varepsilon\, b^{-1} + \varepsilon\, b^{-1} \ = \ (b - i + 1)\, \varepsilon\, b^{-1}.
\end{aligned}$$

■

A.4.6 Korollar 4.9

Korollar 4.9: Adaptive Extraktion
Es sei $\mathcal{X}$ eine Familie von Quellen über Σ, $\eta \leq \eta^{\mathcal{X}}$ eine Symbolgewichtung über Σ, $l \in \mathbb{N}$, $n \in \mathbb{N}$, $\varepsilon > 0$, weiter M_R, M_S endliche, nichtleere Mengen und h eine Familie von universellen Quasi-Hashfunktionen $h_{\tilde{m}} : M_R \times \Sigma^n \to \{0,1\}^{\tilde{m}}$ ($\tilde{m} = 1, \ldots, n$). Weiter sei R eine auf M_R gleichverteilte Zufallsvariable, S eine Zufallsvariable mit Werten in M_S und $X \in \mathcal{X}$.

Wir setzen

$$\begin{aligned} c &:= -2\log\varepsilon + 4\log l - \log n + 2\log \#M_S, \\ m(k) &:= \begin{cases} 0, & (k - c \leq 0), \\ \lfloor k - c \rfloor, & (0 \leq k - c \leq n), \\ n, & (k - c \geq n), \end{cases} \\ \hat{X} &:= \Xi^{n,m}_{\eta,h}(R, X_1 \ldots X_l), \end{aligned}$$

dann ist $\hat{X}$ ε-zufällig unter Kenntnis von $R, S, |\hat{X}|$. □

Beweis: Nach Satz 4.8 ist $\hat{X}$ $\tilde{\varepsilon}$-zufällig unter Kenntnis von R, S, $|\hat{X}|$ mit

$$\log\tilde{\varepsilon} = \tfrac{1}{2} \sup_{\substack{k\in\mathbb{R}_{\geq 0} \\ m(k)\neq 0}} (m(k)\log\#\Sigma_{\text{out}} - k) + \log(l+1) + \log\lfloor l/n \rfloor + \log\#M_S + \tfrac{1}{2}\log\#\big(m(\mathbb{R}_{\geq 0}) \setminus \{0\}\big) - 1. \quad (39)$$

Wir müssen nur zeigen, daß $\varepsilon \geq \tilde{\varepsilon}$.

Anhand der Definition von m überprüft man, daß $m(k) - k \leq -c$ für $m(k) \neq 0$, und wegen $\#\Sigma_{\text{out}} = 2$ erhalten wir

$$\tfrac{1}{2} \sup_{\substack{k\in\mathbb{R}_{\geq 0} \\ m(k)\neq 0}} (m(k)\log\#\Sigma_{\text{out}} - k) \leq -\frac{c}{2} = \log\varepsilon - 2\log l + \tfrac{1}{2}\log n - \log\#M_S. \quad (40)$$

Für $l \geq 1$ gilt $l + 1 \leq 2l$, daher ist auch

$$\log(l+1) \leq \log l + 1. \quad (41)$$

Weiterhin ist

$$\log\lfloor \tfrac{l}{n} \rfloor \leq \log\tfrac{l}{n} = \log l - \log n. \quad (42)$$

Und schließlich wegen $m(\mathbb{R}_{\geq 0}) = \{0, \ldots, n\}$

$$\tfrac{1}{2}\log\#(m(\mathbb{R}_{\geq 0}) \setminus \{0\}) = \tfrac{1}{2}\log n. \quad (43)$$

Wenden wir (40–43) auf (39) an, so erhalten wir

$$\begin{aligned} \log\tilde{\varepsilon} &\leq \log\varepsilon - 2\log l + \tfrac{1}{2}\log n - \log\#M_S \\ &\quad + \log l + 1 + \log l - \log n + \log\#M_S + \tfrac{1}{2}\log n - 1 \\ &= \log\varepsilon, \end{aligned}$$

also $\varepsilon \geq \tilde{\varepsilon}$. ■

A.4.7 Lemma 4.11

Lemma 4.11: Rate einelementiger Quellen
Sei X eine Quelle über Σ und $\mathcal{X} := \{X\}$. Dann ist

$$\eta^{\mathcal{X}}(\alpha;\, x) = \sum_{\nu=1}^{|x|} \eta^{\mathcal{X}}(\alpha x_1 \ldots x_{\nu-1};\, x_\nu) \qquad (\alpha, x \in \Sigma^*) \quad (8)$$

und, falls $H(X)$ existiert,

$$R(\mathcal{X}) = R(X, \mathcal{X}) = H(X). \qquad \square$$

Beweis: Wir schreiben kurz $\eta := \eta^{\mathcal{X}}$. Da $\#\mathcal{X} = 1$, ist

$$\begin{aligned}
\eta(\alpha; x) &\overset{4.1}{=} -\log P(X_{|\alpha|+1} \dots X_{|\alpha x|} = x \mid X_1 \dots X_{|\alpha|} = \alpha) \\
&= -\log \prod_{\nu=1}^{|x|} P(X_{|\alpha|+\nu} = x_\nu \mid X_1 \dots X_{|\alpha|+\nu-1} = \alpha x_1 \dots x_{\nu-1}) \\
&= -\sum_{\nu=1}^{|x|} \log P(X_{|\alpha|+\nu} = x_\nu \mid X_1 \dots X_{|\alpha|+\nu-1} = \alpha x_1 \dots x_{\nu-1}) \\
&\overset{4.1}{=} \sum_{\nu=1}^{|x|} \eta(\alpha x_1 \dots x_{\nu-1}; x_\nu),
\end{aligned}$$

womit (8) gezeigt ist.

Weiter ist

$$\begin{aligned}
R(X, \mathcal{X}) &\overset{4.10}{=} \lim_{n\to\infty} \lim_{l\to\infty} \tfrac{1}{l} \sum_{i=1}^{\lfloor l/n \rfloor} \mathrm{E}\, \eta(X_1 \dots X_{(i-1)n};\ X_{(i-1)n+1} \dots X_{in}) \\
&\overset{(8)}{=} \lim_{l\to\infty} \tfrac{1}{l} \sum_{i=1}^{l} \mathrm{E}\, \eta(X_1 \dots X_{i-1};\ X_i) \\
&\overset{(8)}{=} \lim_{l\to\infty} \tfrac{1}{l}\, \mathrm{E}\, \eta(\lambda; X_1 \dots X_l) \\
&= \lim_{l\to\infty} \tfrac{1}{l} \sum_{x\in\Sigma^l} P(X_1 \dots X_l = x)\eta(\lambda; x) \\
&\overset{4.1}{=} -\lim_{l\to\infty} \tfrac{1}{l} \sum_{x\in\Sigma^l} P(X_1 \dots X_l = x) \log P(X_1 \dots X_l = x) \\
&\overset{2.3}{=} \lim_{l\to\infty} \tfrac{1}{l} H(X_1 \dots X_l) \overset{2.3}{=} H(X),
\end{aligned}$$

und direkt nach Definition 4.10

$$R(\mathcal{X}) = \inf_{X'\in\{X\}} R(X', \mathcal{X}) = R(X, \mathcal{X}). \qquad \blacksquare$$

A.4.8 Gleichung (∗) aus Abschnitt 4.3.3

In Abschnitt 4.3.3 haben wir behauptet, daß für $x \in \Sigma^* \setminus \{\lambda\}$

$$\sup_{p\in\mathbb{R}_1^\Sigma} \sum_{\sigma\in\Sigma} \omega_\sigma(x) \log p_\sigma = |x| \sum_{\sigma\in\Sigma} \frac{\omega_\sigma(x)}{|x|} \log \frac{\omega_\sigma(x)}{|x|}.$$

Beweis: Dividieren wir auf beiden Seiten durch $|x|$, so erhalten wir für geeignetes $a \in \mathbb{R}_1^\Sigma$ die äquivalente Gleichung

$$\sup_{p\in\mathbb{R}_1^\Sigma} \sum_{\sigma\in\Sigma} a_\sigma \log p_\sigma = \sum_{\sigma\in\Sigma} a_\sigma \log a_\sigma. \tag{44}$$

Ist diese gezeigt, so folgt die zu beweisende Aussage.

Für $a_i = 0$ ist

$$\sup_{p\in\mathbb{R}_1^\Sigma} \sum_{\sigma\in\Sigma} a_\sigma \log p_\sigma = \sup_{\substack{p\in\mathbb{R}_1^\Sigma \\ p_i=0}} \sum_{\sigma\in\Sigma} a_\sigma \log p_\sigma = \sup_{p\in\mathbb{R}_1^{\Sigma\setminus\{i\}}} \sum_{\sigma\in\Sigma\setminus\{i\}} a_\sigma \log p_\sigma,$$

da für $p_i > 0$ der Term $\sum_\sigma a_\sigma \log p_\sigma$ vergrößert werden kann, indem wir $p_i := 0$ setzen und p_j mit $a_j \neq 0$ entsprechend erhöhen. Wir können also einfach o. B. d. A. $i \notin \Sigma$ und $a_\sigma \neq 0$ für alle $\sigma \in \Sigma$ annehmen.

Da $\lim_{p_i \to 0} a_i \log p_i = -\infty$, liegt existiert eine globale Maximalstelle und liegt im Inneren von $\mathbb{R}_1^\Sigma$, wir können dann die Multiplikatorenregel von Lagrange anwenden. Nach dieser gilt, da die Ableitung der Nebenbedingung konstant $\mathbb{1}$ ist, und somit nirgends verschwindet, für die Maximalstelle $p \in \mathbb{R}_1^\Sigma$ und geeignetes $\lambda \in \mathbb{R}$:

$$\begin{aligned}
& & 0 &= \frac{\partial \sum a_\sigma \log p_\sigma}{\partial p_i} + \lambda \frac{\partial \sum p_\sigma - 1}{\partial p_i} & & (i \in \Sigma) \\
&\Longrightarrow & 0 &= \frac{a_i}{p_i} + \lambda & & (i \in \Sigma) \\
&\Longrightarrow & a_i &= -\lambda p_i & & (i \in \Sigma)
\end{aligned}$$

Und wegen der Normierung von p und a folgt daraus $p = a$, damit ist (44) bewiesen. ■

A.4.9 Schätzer für die Entropie (Abschnitt 4.3.3)

In Abschnitt 4.3.3 haben wir behauptet, daß

$$\hat{h}(x) := -\sum_{\sigma \in \Sigma} \frac{\omega_\sigma(x)}{|x|} \log \frac{\omega_\sigma(x)}{|x|}$$

ein asymptotisch erwartungstreuer Schätzer der Entropie unabhängig identisch verteilter Zufallsfolgen sei, also für eine solche Folge X gilt

$$\lim_{n\to\infty} \mathrm{E}\, \hat{h}(X_1 \dots X_n) = H(X).$$

Beweis: Es sei $\varepsilon > 0$.

Da $-p_\sigma \log p_\sigma$ stetig in p_σ ist, existiert ein δ, so daß

$$|p_\sigma - f_\sigma(x)| \leq \delta \qquad \Longrightarrow \qquad |p_\sigma \log p_\sigma - f_\sigma(x) \log f_\sigma(x)| \leq \frac{\varepsilon}{2\#\Sigma}.$$

mit $p_\sigma := P(X_0 = \sigma)$, $f_\sigma(x) := \omega_\sigma(x)/|x|$.

Weiter existiert nach dem schwachen Gesetz der großen Zahlen ein $n_0 \in \mathbb{N}$, so daß für $n \geq n_0$ gilt:

$$P(|p_\sigma - \underbrace{f_\sigma(X_1 \dots X_n)}_{=:F_\sigma^{(n)}}| \leq \delta) \leq \frac{\varepsilon}{2\#\Sigma}.$$

Weil $f_\sigma(x) \log f_\sigma(x) \in [0,1]$, hat $-\mathrm{E}\, F_\sigma^{(n)} \log F_\sigma^{(n)}$ für $n \geq n_0$ damit die Form

$$\begin{aligned}
\mathrm{E}\, F_\sigma^{(n)} \log F_\sigma^{(n)} &= P\big(|p_0 - F_\sigma^{(n)}| > \delta\big)\, \mathrm{E}\big(F_\sigma^{(n)} \log F_\sigma^{(n)} \;\big|\; |p_0 - F_\sigma^{(n)}| > \delta\big) \\
&\quad + P\big(|p_0 - F_\sigma^{(n)}| \leq \delta\big)\, \mathrm{E}\big(F_\sigma^{(n)} \log F_\sigma^{(n)} \;\big|\; |p_0 - F_\sigma^{(n)}| \leq \delta\big) \\
&= \frac{\varepsilon}{2\#\Sigma} \hat{h}_1^\sigma + \hat{h}_2^\sigma
\end{aligned} \tag{45}$$

mit geeigneten

$$\hat{h}_1^\sigma \in [0,1] \qquad \text{und} \qquad |\hat{h}_2^\sigma - p_\sigma \log p_\sigma| \leq \frac{\varepsilon}{2\#\Sigma}. \tag{46}$$

Dies fügen wir für $n \geq n_0$ zusammen zu

$$\begin{aligned}
\big|\mathrm{E}\, \hat{h}(X_1 \dots X_n) - H(X_0)\big| &\overset{(45)}{=} \Big|\sum_{\sigma \in \Sigma} \big(\tfrac{\varepsilon}{2\#\Sigma} \hat{h}_1^\sigma + \hat{h}_2^\sigma\big) - \sum_{\sigma \in \Sigma} p_\sigma \log p_\sigma\Big| \\
&\leq \sum_{\sigma \in \Sigma} \Big|\tfrac{\varepsilon}{2\#\Sigma} \hat{h}_1^\sigma\Big| + \sum_{\sigma \in \Sigma} \big|\hat{h}_2^\sigma - p_\sigma \log p_\sigma\big| \\
&\overset{(46)}{\leq} \varepsilon.
\end{aligned}$$

Also ist

$$\lim_{n\to\infty} \mathrm{E}\, \hat{h}(X_1 \dots X_n) = H(X_0) = H(X).$$

■

A.5 Zu Kapitel 5

A.5.1 Lemma 5.4

Lemma 5.4: Zeitinvarianz von CHMM-Familien
Sei $\mathcal{C}$ ein CHMM. Dann ist $\mathcal{X}^{\mathcal{C}}$ links-zeitinvariant und konditioniert links-zeitinvariant. □

Hilfsatz A.25: Einschränkung von Gleichverteilungen
Es sei U auf $[0,1)^{\mathbb{N}_0}$ gleichverteilt. Weiter sei $M \subseteq [0,1)^{\mathbb{N}_0}$ eine meßbare Menge mit $P(U \in M) > 0$, und Z eine auf M gleichverteilte Zufallsvariable. Dann existiert eine meßbare Abbildung

$$f : [0,1)^{\mathbb{N}_0} \longrightarrow [0,1)^{\mathbb{N}_0},$$

so daß $f(U)$ die gleiche Verteilung wie Z hat. □

Der Beweis zu diesem Hilfsatz findet sich auf Seite 75.

Beweis (zu Lemma 5.4): Es sei $X^A \in \mathcal{X}^{\mathcal{C}}$. Wir beweisen zunächst die Links-Zeitinvarianz. Hierfür ist zu zeigen, daß für jedes $n \in \mathbb{N}$ ein Adversary $\tilde{A} \in \mathrm{Adv}_{\mathcal{C}}$ existiert, so daß $X^{\tilde{A}} = (X^A)^{(n)}$ (wir verwenden die Notation aus Definition 4.3). Es genügt hierbei, den Fall $n = 1$ zu untersuchen, siehe die entsprechende Bemerkung auf Seite 22.

Es seien T_i^A, T_*^A, X_i^A, Q_i^A, R, R' wie in Definition 5.3, wobei wir allerdings verschiedene Wahrscheinlichkeitsmaße betrachten werden, und nicht in allen ist R' gleichverteilt.

Der Kürze halber schreiben wir R'_+ für $(R'_{i+1})_{i=0}^{\infty}$ und r_+ für $(r_{i+1})_{i=0}^{\infty}$, d. h. der Index + entfernt das jeweils erste Glied dieser Folgen.

Wir werden die folgenden Wahrscheinlichkeitsmaße betrachten: In P sind R und R' unabhängig und gleichverteilt (wie in Definition 5.3). In $P_{R'=t}$ ist R gleichverteilt und R' konstant t. In $P_{R'_+=t}$ sind R und R'_0 unabhängig und gleichverteilt, und R'_+ konstant t. Man kann sich diese Wahrscheinlichkeitsmaße als nach $R' = t$ bzw. $R'_+ = t$ konditionierte Wahrscheinlichkeitsmaße vorstellen, jedoch ist dies formal nicht korrekt, da $P(R' = t) = P(R'_+ = t) = 0$, und somit $P(\,\cdot\,|R' = t)$ und $P(\,\cdot\,|R'_+ = t)$ undefiniert sind.

Es existieren nun meßbare Funktionen $\hat{q}_0, \hat{q}_1 : [0,1) \times [0,1)^{\mathbb{N}_0} \to Q_{\mathcal{C}}$ und $\hat{x}_1 : [0,1) \times [0,1)^{\mathbb{N}_0} \to \Sigma_{\mathcal{C}}$ mit

$$\int_0^1 \delta\big(\hat{q}_0(r_0,r_+) = q_0,\ \hat{q}_1(r_0,r_+) = q_1,\ \hat{x}_1(r_0,r_+) = x_1\big)\ dr_0 = P_{R'=r_+}\big(Q_0^A = q_0,\ Q_1^A = q_1,\ X_1^A = x_1\big) \tag{47}$$

für alle $q_0, q_1 \in Q_{\mathcal{C}}$, $x_1 \in \Sigma_{\mathcal{C}}$ und $r_+ \in [0,1)^{\mathbb{N}_0}$. Diese Funktionen sind so konstruiert, daß sie die Verteilung von Q_0^A, Q_1^A, X_1^A besitzen, wenn r_0 auf $[0,1)$ gleichverteilt ist.

Wir können mit diesen Hilfsmitteln nun den Adversary $\tilde{A}$ definieren: Es seien für $r \in [0,1)^{\mathbb{N}_0}$, $i \in \mathbb{N}$, $q, q_i \in Q_{\mathcal{C}}$ und $x_i \in \Sigma_{\mathcal{C}}$

$$\tilde{A}^*(r) := \big(\delta(q = \hat{q}_1(r_0,r_+))\big)_{q \in Q_{\mathcal{C}}}, \tag{48}$$

und

$$\begin{aligned}&\tilde{A}\big(i, r, q, (q_0,\ldots,q_{i-1}), (x_0,\ldots,x_{i-1})\big)\\ &\qquad := A\big(i+1, r_+, q, (\hat{q}_0(r_0,r_+), \hat{q}_1(r_0,r_+), q_1,\ldots,q_{i-1}), (\hat{x}_1(r_0,r_+), x_1,\ldots,x_{i-1})\big).\end{aligned} \tag{49}$$

Es ist offenbar auch $\tilde{A} \in \mathrm{Adv}_{\mathcal{C}}$. Dieser Adversary ist so konstruiert, daß er zu Beginn das Anfangsverhalten von A und X^A mittels $\hat{q}_0$, $\hat{q}_1$, $\hat{x}_1$ simuliert und dann jeden Aufruf an A weiterleitet, wobei er die Folgen der bereits erreichten Zustände und ausgegebenen Symbole um $\hat{q}_0$ und $\hat{x}_1$ erweitert, so daß es A immer scheint, wir seien schon um eine Stelle weiter in der Ausgabe der Quelle fortgeschritten. Der Effekt ist, daß der Ausgabe von $X^{\tilde{A}}$ das erste Glied fehlt. Dies wollen wir nun zeigen:

Es ist für $i \in \mathbb{N}$, $q_\mu \in Q_{\mathcal{C}}$, $x_\mu \in \Sigma_{\mathcal{C}}$:

$$
\begin{aligned}
&P_{R'=r}\big(Q_0^{\tilde{A}} \dots Q_i^{\tilde{A}} = q_0 \dots q_i,\ X_1^{\tilde{A}} \dots X_i^{\tilde{A}} = x_1 \dots x_i\big)\\
&\quad = P_{R'=r}(Q_0^{\tilde{A}} = q_0) \prod_{\nu=1}^{i} P_{R'=r}\big(Q_\nu^{\tilde{A}} = q_\nu,\ X_\nu^{\tilde{A}} = x_\nu \mid Q_0^{\tilde{A}} \dots Q_{\nu-1}^{\tilde{A}} = q_0 \dots q_{\nu-1},\ X_1^{\tilde{A}} \dots X_{\nu-1}^{\tilde{A}} = x_1 \dots x_{\nu-1}\big)\\
&\quad \overset{5.3}{=} (\tilde{A}^*(r))_{q_0} \prod_{\nu=1}^{i} \big(\tilde{A}(\nu, r, q_{\nu-1}, (q_0, \dots, q_{\nu-1}), (x_1, \dots, x_{\nu-1}))\big)_{x_\nu, q_\nu}\\
&\quad \overset{(49)}{=} (\tilde{A}^*(r))_{q_0} \prod_{\nu=1}^{i} \big(A(\nu+1, r_+, q_{\nu-1}, (\hat{q}_0(r_0, r_+), \hat{q}_1(r_0, r_+), q_1, \dots, q_{\nu-1}), (\hat{x}_1(r_0, r_+), x_1, \dots, x_{\nu-1}))\big)_{x_\nu, q_\nu}\\
&\quad \overset{5.3}{=} (\tilde{A}^*(r))_{q_0} \prod_{\nu=1}^{i} P_{R'=r_+}\big(Q_{\nu+1}^{A} = q_\nu,\ X_{\nu+1}^{A} = x_\nu \mid Q_0^{A} \dots Q_\nu^{A} = \hat{q}_0(r_0, r_+), \hat{q}_1(r_0, r_+), q_1, \dots, q_{\nu-1},\\
&\qquad\qquad\qquad X_1^{A} \dots X_\nu^{A} = \hat{x}_1(r_0, r_+), x_1, \dots, x_{\nu-1}\big)\\
&\quad = (\tilde{A}^*(r))_{q_0}\ P_{R'=r_+}\big(Q_2^{A} \dots Q_{i+1}^{A} = q_1 \dots q_i,\ X_2^{A} \dots X_{i+1}^{A} = x_1 \dots x_i\\
&\qquad\qquad\qquad \mid Q_0^{A} = \hat{q}_0(r_0, r_+),\ Q_1^{A} = \hat{q}_1(r_0, r_+),\ X_1^{A} = \hat{x}_1(r_0, r_+)\big)\\
&\quad \overset{(48)}{=} \delta(q_0 = \hat{q}_1(r_0, r_+)) \sum_{\substack{q_0', q_1' \in Q_{\mathcal{C}}\\ x_1' \in \Sigma_{\mathcal{C}}}} P_{R'=r_+}\big(Q_2^{A} \dots Q_{i+1}^{A} = q_1 \dots q_i,\ X_2^{A} \dots X_{i+1}^{A} = x_1 \dots x_i\\
&\qquad\qquad\qquad \mid Q_0^{A} = q_0',\ Q_1^{A} = q_1',\ X_1^{A} = x_1'\big)\cdot\\
&\qquad\qquad\qquad \delta\big(q_0' = \hat{q}_0(r_0, r_+),\ q_1' = \hat{q}_1(r_0, r_+),\ x_1' = \hat{x}_1(r_0, r_+)\big).
\end{aligned}
\tag{50}
$$

Weiter ist

$$
\begin{aligned}
&P_{R'_+=r_+}(Q_0^{\tilde{A}} \dots Q_i^{\tilde{A}} = q_0 \dots q_i,\ X_1^{\tilde{A}} \dots X_i^{\tilde{A}} = x_1 \dots x_i)\\
&\quad = \int_0^1 P_{R'=r}(Q_0^{\tilde{A}} \dots Q_i^{\tilde{A}} = q_0 \dots q_i,\ X_1^{\tilde{A}} \dots X_i^{\tilde{A}} = x_1 \dots x_i)\, dr_0\\
&\quad \overset{(50)}{=} \sum_{\substack{q_0', q_1' \in Q_{\mathcal{C}}\\ x_1' \in \Sigma_{\mathcal{C}}}} P_{R'=r_+}\big(Q_2^{A} \dots Q_{i+1}^{A} = q_1 \dots q_i,\ X_2^{A} \dots X_{i+1}^{A} = x_1 \dots x_i\\
&\qquad\qquad \mid Q_0^{A} = q_0',\ Q_1^{A} = q_1',\ X_1^{A} = x_1'\big)\cdot\\
&\qquad\qquad \int_0^1 \delta\big(q_0' = \hat{q}_0(r_0, r_+),\ q_0 = q_1' = \hat{q}_1(r_0, r_+),\ x_1' = \hat{x}_1(r_0, r_+)\big)\, dr_0\\
&\quad \overset{(47)}{=} \sum_{\substack{q_0', q_1' \in Q_{\mathcal{C}}\\ x_1' \in \Sigma_{\mathcal{C}}}} P_{R'=r_+}\big(Q_2^{A} \dots Q_{i+1}^{A} = q_1 \dots q_i,\ X_2^{A} \dots X_{i+1}^{A} = x_1 \dots x_i\\
&\qquad\qquad \mid Q_0^{A} = q_0',\ Q_1^{A} = q_1',\ X_1^{A} = x_1'\big)\cdot\\
&\qquad\qquad P_{R'=r_+}\big(Q_0^{A} = q_0',\ Q_1^{A} = q_1' = q_0,\ X_1^{A} = x_1'\big)\\
&\quad = P_{R'=r_+}\big(Q_1^{A} \dots Q_{i+1}^{A} = q_0 \dots q_i,\ X_2^{A} \dots X_{i+1}^{A} = x_1 \dots x_i\big).
\end{aligned}
\tag{51}
$$

Damit ist auch

$$P(Q_0^{\tilde{A}} \dots Q_i^{\tilde{A}} = q_0 \dots q_i,\ X_1^{\tilde{A}} \dots X_i^{\tilde{A}} = x_1 \dots x_i) = P\big(Q_1^{A} \dots Q_{i+1}^{A} = q_0 \dots q_i,\ X_2^{A} \dots X_{i+1}^{A} = x_1 \dots x_i\big).$$

und insbesondere

$$P(X_1^{\tilde{A}} \dots X_i^{\tilde{A}} = x_1 \dots x_i) = P\big(X_2^{A} \dots X_{i+1}^{A} = x_1 \dots x_i\big) = P\big((X^A)_1^{(1)} \dots (X^A)_i^{(1)} = x_1 \dots x_i\big). \tag{52}$$

Damit ist $\mathcal{X}^{\mathcal{C}}$ als links-zeitinvariant bewiesen.

Nun wollen zeigen, daß $\mathcal{X}^{\mathcal{C}}$ auch konditioniert links-zeitinvariant ist, d. h. daß es ein $\bar{A} \in \mathrm{Adv}_{\mathcal{C}}$ gibt, so daß $\mathcal{X}^{\bar{A}} = (X^A)^{(n)} | (X_1^A \dots X_n^A = \tilde{x}_1 \dots \tilde{x}_n)$, sofern $P(X_1^A \dots X_n^A = \tilde{x}_1 \dots \tilde{x}_n) > 0$. Auch diesmal kann man sich auf den Fall $n = 1$ beschränken, vergleiche die entsprechende Bemerkung auf Seite 22.

Es sei wieder $X^A \in \mathcal{X}^{\mathcal{C}}$.

Im durch die Bedingung $X_1^A = \tilde{x}_1$ gegebenen Wahrscheinlichkeitsmaß ist (R_0, R_1, R') auf einer Teilmenge von $[0,1)^{\mathbb{N}_0} \times [0,1)^{\mathbb{N}_0}$ gleichverteilt, da X_1^A deterministisch von (R_0, R_1, R') abhängt und (R_0, R_1, R') gleichverteilt war. Somit existiert nach Hilfsatz A.25 eine meßbare Abbildung $f : [0,1)^{\mathbb{N}_0} \to [0,1)^{\mathbb{N}_0}$, so daß $f(U)$ die gleiche Verteilung wie $(R_0, R_1, R')|(X_1^A = \tilde{x}_1)$ hat, wobei U unabhängig von R, R' und auf $[0,1)^{\mathbb{N}_0}$ gleichverteilt sei. Der Kürze halber schreiben wir noch $f'(r) := (f(r)_i)_{i=2}^{\infty}$.

Dann seien $\bar{q}_0$, $\bar{q}_1$ und $\bar{x}_1$ derart, daß für $P_{R'=r_+}(X_1^A = \tilde{x}_1) > 0$ die folgende Gleichung erfüllt ist:

$$\begin{aligned}\int_0^1 &\delta(\bar{q}_0(r_0, r_+) = q_0,\ \bar{q}_1(r_0, r_+) = q_1,\ \bar{x}_1(r_0, r_+) = x_1)\ dr_0 \\ &= P_{(R_0,R_1,R')=f(r_+)}(Q_0^A = q_0,\ Q_1^A = q_1,\ X_1^A = x_1)\end{aligned} \tag{53}$$

für alle $q_0, q_1 \in Q_{\mathcal{C}}$, $x_1 \in \Sigma_{\mathcal{C}}$ und $r_+ \in [0,1)^{\mathbb{N}_0}$.

Wir definieren nun den Adversary $\bar{A}$: Es seien für $r \in [0,1)^{\mathbb{N}_0}$, $i \in \mathbb{N}$, $q, q_i \in Q_{\mathcal{C}}$ und $x_i \in \Sigma_{\mathcal{C}}$

$$\bar{A}^*(r) := \big(\delta(q = \bar{q}_1(r_0, f(r_+)))\big)_{q \in Q_{\mathcal{C}}},$$

und

$$\begin{aligned}&\bar{A}(i, r, q, (q_0, \dots, q_{i-1}), (x_0, \dots, x_{i-1})) \\ &\quad := A(i+1, f'(r_+), q, (\bar{q}_0(r_0, r_+), \bar{q}_1(r_0, r_+), q_1, \dots, q_{i-1}), (\bar{x}_1(r_0, r_+), x_1, \dots, x_{i-1})).\end{aligned}$$

Dieser Adversary $\bar{A}$ unterscheidet sich vom oben definierten $\tilde{A}$ dadurch, daß er zusätzlich zu dem, was $\tilde{A}$ tut, R' und R_0, R_1 in der Simulation nicht als gleichverteilt annimmt, sondern als entsprechend der Bedingung $X_1^A = \tilde{x}_1$ verteilt. Dies ist möglich, da $\bar{A}$ die Zufallsvariable R' beliebig abbilden kann, bevor sie an den simulierten A weitergegeben wird, und weil R_0 und R_1 nur in der Simulation vorkommen. $R_2, R_3, \dots$ müssen nicht modifiziert werden, da sie von X_1^A unabhängig sind. Formal sieht dies wie folgt aus:

Ganz analog zur Herleitung von (50) erhalten wir

$$\begin{aligned}&P_{R'=r}\big(Q_0^{\bar{A}} \dots Q_i^{\bar{A}} = q_0 \dots q_i,\ X_1^{\bar{A}} \dots X_i^{\bar{A}} = x_1 \dots x_i\big) \\ &= \delta(q_0 = \bar{q}_1(r_0, r_+)) \sum_{\substack{q_0', q_1' \in Q_{\mathcal{C}} \\ x_1' \in \Sigma_{\mathcal{C}}}} P_{R'=f'(r_+)}\big(Q_2^A \dots Q_{i+1}^A = q_1 \dots q_i,\ X_2^A \dots X_{i+1}^A = x_1 \dots x_i \\ &\qquad\qquad \mid Q_0^A = q_0',\ Q_1^A = q_1',\ X_1^A = x_1'\big) \cdot \\ &\qquad\qquad \delta\big(q_0' = \bar{q}_0(r_0, r_+),\ q_1' = \bar{q}_1(r_0, r_+),\ x_1' = \bar{x}_1(r_0, r_+)\big).\end{aligned} \tag{54}$$

Analog zu (51) folgern wir

$$\begin{aligned}&P_{R'_+=r_+}(Q_0^{\bar{A}} \dots Q_i^{\bar{A}} = q_0 \dots q_i,\ X_1^{\bar{A}} \dots X_i^{\bar{A}} = x_1 \dots x_i) \\ &\overset{(53,54)}{=} \sum_{\substack{q_0', q_1' \in Q_{\mathcal{C}} \\ x_1' \in \Sigma_{\mathcal{C}}}} P_{R'=f'(r_+)}\big(Q_2^A \dots Q_{i+1}^A = q_1 \dots q_i,\ X_2^A \dots X_{i+1}^A = x_1 \dots x_i \\ &\qquad\qquad \mid Q_0^A = q_0',\ Q_1^A = q_1',\ X_1^A = x_1'\big) \cdot \\ &\qquad\qquad P_{(R_0,R_1,R')=f(r_+)}\big(Q_0^A = q_0',\ Q_1^A = q_1' = q_0,\ X_1^A = x_1'\big) \\ &\overset{(*)}{=} \sum_{\substack{q_0', q_1' \in Q_{\mathcal{C}} \\ x_1' \in \Sigma_{\mathcal{C}}}} P_{(R_0,R_1,R')=f(r_+)}\big(Q_2^A \dots Q_{i+1}^A = q_1 \dots q_i,\ X_2^A \dots X_{i+1}^A = x_1 \dots x_i \\ &\qquad\qquad \mid Q_0^A = q_0',\ Q_1^A = q_1',\ X_1^A = x_1'\big) \cdot \\ &\qquad\qquad P_{(R_0,R_1,R')=f(r_+)}\big(Q_0^A = q_0',\ Q_1^A = q_1' = q_0,\ X_1^A = x_1'\big) \\ &= P_{(R_0,R_1,R')=f(r_+)}\big(Q_1^A \dots Q_{i+1}^A = q_0 \dots q_i,\ X_2^A \dots X_{i+1}^A = x_1 \dots x_i\big).\end{aligned} \tag{55}$$

Hierbei gilt die mit $(*)$ gekennzeichnete Gleichheit, weil gegeben den Fall $Q_0^A = q_0'$, $Q_1^A = q_1'$, $X_1^A = x_1'$ das Eintreten des Ereignisses $Q_2^A \dots Q_{i+1}^A = q_1 \dots q_i$, $X_2^A \dots X_{i+1}^A = x_1 \dots x_i$ gar nicht von R_0 oder R_1 abhängt.

Nun haben (R, R') gegeben $X_1^A = \tilde{x}_1$ die Verteilung von $\big((f(U)_0, f(U)_1, R_2, R_3, \dots), (f(R')_i)_{i \geq 2}\big)$, da X_1^A nicht von $R_2, R_3, \dots$ abhängt. Damit erhält man durch Integrieren von (55) nach r_+ über $[0,1)^{\mathbb{N}_0}$:

$$\begin{aligned}&P(Q_0^{\bar{A}} \dots Q_i^{\bar{A}} = q_0 \dots q_i,\ X_1^{\bar{A}} \dots X_i^{\bar{A}} = x_1 \dots x_i) \\ &\qquad = P\big(Q_1^A \dots Q_{i+1}^A = q_0 \dots q_i,\ X_2^A \dots X_{i+1}^A = x_1 \dots x_i \mid X_1^A = \tilde{x}_1\big)\end{aligned}$$

und insbesondere

$$P(X_1^{\bar{A}} \dots X_i^{\bar{A}} = x_1 \dots x_i) = P\big(X_2^A \dots X_{i+1}^A = x_1 \dots x_i \mid X_1^A = \tilde{x}_1\big)$$
$$= P\big((X^A)_1^{(1)} \dots (X^A)_i^{(1)} = x_1 \dots x_i \mid X_1^A = \tilde{x}_1\big).$$

Damit hat X^A die gleiche Verteilung wie $(X^A)^{(1)}|(X_1^A = \tilde{x}_1)$, und $\mathcal{X}^{\mathcal{C}}$ ist konditioniert links-zeitinvariant. ■

Beweis (zu Hilfsatz A.25): Es sei

$$g_i(x_1, \dots, x_i) := P\big(U \in \{(t_{i+1}, t_{i+2}, \dots) : (x_1, \dots, x_i, t_{i+1}, t_{i+2}, \dots) \in M\}\big)/P(U \in M).$$

Dann ist g_i die Dichte von $(Z_1, \dots, Z_i)$. Man beachte außerdem, daß

$$\int_0^1 g_i(x_1, \dots, x_{i-1}, t)\, dt = g_{i-1}(x_1, \dots, x_{i-1}) \qquad \text{und} \qquad \int_0^1 g_1(t)\, dt = 1. \tag{56}$$

Wir konstruieren nun

$$h_i(p_1, \dots, p_i) := \sup \left\{ x \in [0,1) : \int_0^x g_i(f_{i-1}(p_1, \dots, p_{i-1}), t)\, dt \le p_i \int_0^1 g_i(f_{i-1}(p_1, \dots, p_{i-1}), t)\, dt \right\}$$

und

$$f_i(p_1, \dots, p_i) := \big(h_\nu(p_1, \dots, p_\nu)\big)_{\nu=1}^i, \qquad f(p_1, \dots, p_i) := \big(h_\nu(p_1, \dots, p_\nu)\big)_{\nu=1}^\infty.$$

Wir wollen nun zeigen, daß g_i auch die Dichte von $f_i(U)$ ist. Haben wir dies gezeigt, so sind $f_i(U_1, \dots, U_i)$ und $Z_1, \dots, Z_i$ von gleicher Verteilung und damit auch $f(U)$ und Z.

Wir verwenden vollständige Induktion und zeigen zunächst, daß g_1 die Dichte von $f_1(U_1)$ ist. Es ist für $x_1 \in [0,1)$:

$$f_1(p_1) \le x_1 \qquad \Longleftrightarrow \qquad h_1(p_1) \le x_1 \qquad \Longleftrightarrow \qquad \int_0^{x_1} g_1(t)\, dt \le p_1 \int_0^1 g_1(t)\, dt \stackrel{(56)}{=} p_1 \tag{57}$$

und somit

$$P(f_1(U_1) \le x_1) = \int_0^1 \delta(f_1(p_1) \le x_1)\, dp_1 = \max\{p_1 \in [0,1] : f_1(p_1) \le x_1\} \stackrel{(57)}{=} \int_0^{x_1} g_1(t)\, dt.$$

Damit ist g_1 Dichte von $f_1(U_1)$.

Sei nun $i \ge 2$ und g_{i-1} die Dichte von $f_{i-1}(U_1, \dots, U_{i-1})$. Dann gilt es zu zeigen, daß g_i die Dichte von $f_i(U_1, \dots, U_i)$ ist.

Hierzu sei zunächst

$$h_i'(a_1, \dots, a_{i-1}, p_i) := \sup \left\{ x \in [0,1) : \int_0^x g(a_1, \dots, a_{i-1}, t)\, dt \le p_i \int_0^1 g(a_1, \dots, a_{i-1}, t)\, dt \right\}.$$

Damit ergibt sich für $a_1, \dots, a_{i-1} \in [0,1)$ mit $g_{i-1}(a_1, \dots, a_{i-1}) > 0$ und $x \in [0,1)$:

$$P(h'(a_1, \dots, a_{i-1}, U_i) \le x_i) = P\left(\int_0^x g_i(a_1, \dots, a_{i-1}, t)\, dt \ge U_i \int_0^1 g_i(a_1, \dots, a_{i-1}, t)\, dt \right)$$
$$\stackrel{(56)}{=} \frac{\int_0^x g_i(a_1, \dots, a_{i-1}, t)\, dt}{g_{i-1}(a_1, \dots, a_{i-1})}. \tag{58}$$

Dann haben wir

$$\begin{aligned}
&\int_0^{x_1}\cdots\int_0^{x_i} g_i(t_1,\dots,t_i)\,dt_i\dots dt_1\\
&\quad=\int_0^{x_1}\cdots\int_0^{x_{i-1}} g_{i-1}(t_1,\dots,t_{i-1})\int_0^{x_i}\frac{g_i(t_1,\dots,t_i)}{g_{i-1}(t_1,\dots,t_{i-1})}\,dt_i\,dt_{i-1}\dots dt_1\\
&\quad\overset{(58)}{=}\int_0^{x_1}\cdots\int_0^{x_{i-1}} g_{i-1}(t_1,\dots,t_{i-1})\;P(h_i'(t_1,\dots,t_{i-1},U_i)\le x_i)\,dt_{i-1}\dots dt_1\\
&\quad\overset{\text{I.V.}}{=}P\big(h_i'(f_{i-1}(U_1,\dots,U_{i-1}),U_i)\le x,\ (f_{i-1}(U_1,\dots,U_\nu))_\nu\le x_\nu\ (\nu=1,\dots,i-1)\big)\\
&\quad=P\big((f_i(U_1,\dots,U_i))_\nu\le x_\nu\ (\nu=1,\dots,i)\big).
\end{aligned}$$

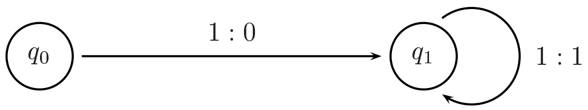
Damit ist g_i Dichte von $f_i(U_1,\dots,U_i)$ und der Induktionsbeweis abgeschlossen. ■

A.5.2 Bemerkung Seite 31

> **Definition 4.4: Rechts-zeitinvariante Familien von Quellen**
> Es sei $Y^{(n)}$ analog zu $X^{(n)}$ in der vorangehenden Definition.
> Eine Familie $\mathcal{X}$ von Quellen heißt *rechts-zeitinvariant*, wenn für jedes $X\in\mathcal{X}$ und jedes $n\in\mathbb{N}_0$ ein $Y\in\mathcal{X}$ existiert mit $Y^{(n)}=X$. □

Auf Seite 31 wurde angemerkt, daß eine durch ein CHMM definierte Familie von Quellen i. a. nicht rechts-zeitinvariant ist.

Beweis: Betrachte das CHMM $\mathcal{C}$, definiert durch[30]



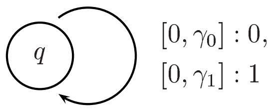
Dann hängt X^A nur von A^* ab (mit $A\in\mathrm{Adv}_{\mathcal{C}}$), und es ist $\mathcal{X}^{\mathcal{C}}$ die Menge der Quellen X mit

$$P(X=01^\infty \text{ oder } X=1^\infty)=1.$$

Sei $X_0\in\mathcal{X}^{\mathcal{C}}$ konstant 01^∞. Dann müßte ein Y mit $Y^{(1)}=X$ die Eigenschaft $P(Y_2=0)=1$ haben, aber kein $Y\in\mathcal{X}^{\mathcal{C}}$ hat diese. Also ist $\mathcal{X}^{\mathcal{C}}$ nicht rechts-zeitinvariant. ■

A.5.3 Symbolgewichtung aus Abschnitten 5.2.4 und 5.2.5

Die CHMM aus Abschnitten 5.2.4 und 5.2.5 sind Spezialfälle des folgenden CHMM $\mathcal{C}$ mit festem $\gamma_0,\gamma_1\in[0,1]$, $\gamma_0+\gamma_1\ge 1$:



Es hat $\mathcal{C}$ die folgende Symbolgewichtung:

$$\eta^{\mathcal{C}}(\alpha;x)\;=\;-\omega_1(x)\log\gamma_1-\omega_0(x)\log\gamma_0.$$

Beweis: Wir bedienen uns des erst in Abschnitt 5.3 eingeführten Satzes 5.5 und übernehmen dessen Notation. Da $\#Q_{\mathcal{C}}=1$, ist

$$\mathcal{N}(M)=\mathbb{R}_1=\{1\}. \tag{59}$$

[30]Dateiname `notrti.chmm` (siehe Abschnitt B.4).

Weiterhin entartet $\max \mathcal{T}_x^{\mathcal{C}}(M)$ zu

$$\max \mathcal{T}_x^{\mathcal{C}}(M) = \{\, t_x p \ : \ t_0 \in [0, \gamma_0],\ t_1 \in [0, \gamma_1],\ t_0 + t_1 = 1,\ p \in M \,\} = \gamma_x \max M, \tag{60}$$

sofern $\max M$ existiert.

Damit ergibt sich nach Satz 5.5

$$\eta^{\mathcal{C}}(\alpha; x) = -\log \max_{p \in \mathcal{P}} \|p\|_1$$

mit

$$\mathcal{P} := \mathcal{T}^{\mathcal{C}}_{x_{|x|}} \circ \cdots \circ \mathcal{T}^{\mathcal{C}}_{x_1} \circ \mathcal{N} \circ \mathcal{T}^{\mathcal{C}}_{\alpha_{|\alpha|}} \circ \cdots \circ \mathcal{T}^{\mathcal{C}}_{\alpha_1}(\{1\}).$$

Nach (59) und (60) erhalten wir daraus induktiv

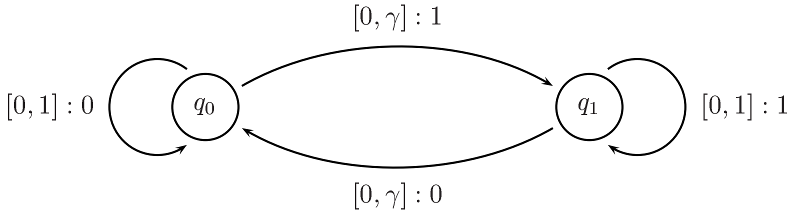
$$\max \mathcal{P} = \prod_{i=1}^{|x|} \gamma_{x_i} = \gamma_1^{\omega_1(x)} \gamma_0^{\omega_0(x)},$$

also

$$\eta^{\mathcal{C}}(\alpha; x) = -\log \gamma_1^{\omega_1(x)} \gamma_0^{\omega_0(x)} = -\omega_1(x) \log \gamma_1 - \omega_0(x) \log \gamma_0. \qquad \blacksquare$$

A.5.4 Symbolgewichtung aus Abschnitt 5.2.6

In Abschnitt 5.2.6 haben wir behauptet, daß das folgende CHMM $\mathcal{C}$



die Symbolgewichtung

$$\eta^{\mathcal{C}}(\alpha; x) = \begin{cases} -\kappa(x) \log \gamma, & \text{falls } \alpha = \lambda, \\ -\kappa(\alpha_{|\alpha|} x) \log \gamma, & \text{sonst}, \end{cases} \qquad (\alpha, x \in \{0,1\}^*)$$

hat, wobei $\kappa(x)$ für $x = x_1 \dots x_n \in \{0,1\}^*$ die Anzahl der $i \in \{1, \dots, n-1\}$ mit $x_i \neq x_{i+1}$ bezeichne.

Beweis: Wir bedienen uns wieder des erst in Abschnitt 5.3 eingeführten Satzes 5.5 und übernehmen dessen Notation. Damit sie besser zu dieser Notation paßt, formulieren wir die zu beweisende Aussage wie folgt um:

$$\eta^{\mathcal{C}}(x_1 \dots x_{i-1};\ x_i \dots x_j) = \begin{cases} -\kappa(x_i \dots x_j) \log \gamma, & \text{falls } i = 1, \\ -\kappa(x_{i-1} \dots x_j) \log \gamma, & \text{falls } i \geq 2, \end{cases} \qquad (1 \leq i \leq j,\ x \in \{0,1\}^j).$$

In Satz 5.5 entartet nun $\mathcal{T}_x^{\mathcal{C}}$ zu

$$\mathcal{T}_x^{\mathcal{C}}(M) = \{\, p' \in \mathbb{R}_{\geq 0}^{Q_{\mathcal{C}}} \ : \ p'_{1-x} = 0,\ p'_x \leq p_x + \gamma p_{1-x},\ p \in M \,\}.$$

Man kann nun mit Satz 5.5 die folgenden Gleichungen nacheinander induktiv nachrechnen:

$$\mathcal{P}_{1,\nu} = \left\{ p \in \mathbb{R}_{\geq 0}^{Q_{\mathcal{C}}} : p_{x_\nu} \leq \prod_{\mu=1}^{\nu-1} (\delta(x_\mu = x_{\mu+1}) \cdot 1 + \delta(x_\mu \neq x_{\mu+1}) \cdot \gamma),\ p_{1-x_\nu} = 0 \right\} \qquad (\nu = 1, \dots, j),$$

und für $i \geq 2$

$$\mathcal{N}(\mathcal{P}_{1,i-1}) = \{e_{x_{i-1}}\}, \qquad (i \geq 2),$$

$$\mathcal{P}_{i,\nu} = \left\{ p \in \mathbb{R}_{\geq 0}^{Q_{\mathcal{C}}} : p_{x_\nu} \leq \prod_{\mu=i-1}^{\nu-1} (\delta(x_\mu = x_{\mu+1}) \cdot 1 + \delta(x_\mu \neq x_{\mu+1}) \cdot \gamma),\ p_{1-x_\nu} = 0 \right\} \qquad (\nu = i, \dots, j),$$

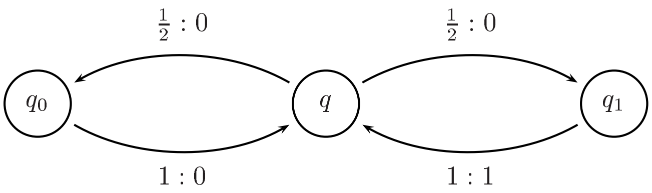
woraus wegen $\kappa(x_\nu \dots x_\mu) = \sum_{\pi=\nu}^{\mu-1} \delta(x_\pi \neq x_{\pi+1})$ direkt

$$\begin{aligned}
\eta^{\mathcal{C}}(x_1 \dots x_{i-1};\ x_i \dots x_j) &= -\log \max_{p \in \mathcal{P}_{i,j}} \|p\|_1 \\
&= \begin{cases} -\log \gamma^{\kappa(x_1 \dots x_j)}, & \text{falls } i = 1, \\ -\log \gamma^{\kappa(x_{i-1} \dots x_j)}, & \text{falls } i \geq 2, \end{cases} \\
&= \begin{cases} -\kappa(x_i \dots x_j) \log \gamma, & \text{falls } i = 1, \\ -\kappa(x_{i-1} \dots x_j) \log \gamma, & \text{falls } i \geq 2 \end{cases}
\end{aligned}$$

folgt. ■

A.5.5 Symbolgewichtung aus Abschnitt 5.2.7

In Abschnitt 5.2.7 haben wir behauptet, daß das folgende CHMM $\mathcal{C}$



die Symbolgewichtung $\eta^{\mathcal{C}}$ mit

$$\eta^{\mathcal{C}}(\lambda; 00) = 0, \quad \eta^{\mathcal{C}}(00; 0) = 0, \quad \eta^{\mathcal{C}}(\lambda; 000) = 1$$

hat.

Beweis: Diese Aussage kann mit Satz 5.5 direkt nachgerechnet werden, oder wie folgt gezeigt:

Da alle Transitionsbereiche einelementig sind, kann der Adversary nur bei Wahl des ersten Anfangszustands auf die Quelle Einfluß nehmen. Wir unterscheiden zunächst zwei Adversaries A_q und A_0, welche deterministisch den Anfangszustand q bzw. q_0 wählen.

Es ist dann

$$P(X_1^{A_0} X_2^{A_0} = 00) = 1, \qquad P(X_1^{A_q} X_2^{A_q} = 00) > 0, \qquad \text{und} \qquad P(X_3^{A_q} = 0) = 1,$$

also $\eta^{\mathcal{C}}(\lambda; 00) = 0$ und $\eta^{\mathcal{C}}(00; 0) = 0$.

Zuletzt betrachten wir den allgemeinen Adversary A, der die Anfangszustände q, q_0, q_1 mit den Wahrscheinlichkeiten p_q, p_0 bzw. p_1 wählt. Dann ist

$$P(X_1^A X_2^A X_3^A = 000) = p_q \cdot \tfrac{1}{2} + p_0 \cdot \tfrac{1}{2} + p_1 \cdot 0 \leq \tfrac{1}{2},$$

also $\eta^{\mathcal{C}}(\lambda; 000) \geq 1$, und, wie man am Fall $p_q = 1$ erkennt, $\eta^{\mathcal{C}}(\lambda; 000) = 1$. ■

A.5.6 Satz 5.5

> **Satz 5.5: Berechnung der Symbolgewichtung von CHMM**
> Es sei $\mathcal{C}$ ein CHMM und $x \in \Sigma^{\mathbb{N}}$. Weiterhin seien die folgenden Abbildungen auf $2^{\mathbb{R}_{\geq 0}^{Q_{\mathcal{C}}}}$ definiert:
>
> $$\begin{aligned} \mathcal{N}(M) &:= \left\{ \tfrac{p}{\|p\|_1} : p \in M \setminus \{0\} \right\}, \\ \mathcal{T}_x^{\mathcal{C}}(M) &:= \left\{ \Big(\sum_{q' \in Q_{\mathcal{C}}} t_{x,q}^{(q')} p_{q'} \Big)_q :\ t^{(q')} \in \overline{\overline{\mathcal{C}}}_{q'},\ p \in M \right\}, \end{aligned}$$
>
> wobei $\overline{\overline{\mathcal{C}}}_{q'}$ den topologischen Abschluß (im folgenden immer kurz Abschluß genannt) der konvexen Hülle von $\mathcal{C}_{q'}$ bezeichne.

Betrachte folgende Rekursion:

$$\begin{aligned} \mathcal{P}_{1,1} &:= \mathcal{T}^{\mathcal{C}}_{x_1}(\mathbb{R}^{Q_\mathcal{C}}_1), && \\ \mathcal{P}_{j,j} &:= \mathcal{T}^{\mathcal{C}}_{x_j} \circ \mathcal{N}(\mathcal{P}_{i,j-1}) && (1 \le i < j), \\ \mathcal{P}_{i,j} &:= \mathcal{T}^{\mathcal{C}}_{x_j}(\mathcal{P}_{i,j-1}) && (1 \le i < j). \end{aligned} \tag{11}$$

Hierbei ist (11) wohldefiniert.

Dann ist

$$\eta^{\mathcal{C}}(x_1 \dots x_{i-1};\ x_i \dots x_j) = -\log \max_{p \in \mathcal{P}_{i,j}} \|p\|_1 \qquad (1 \le i \le j). \qquad \square$$

Hilfsatz A.28: Transitionswahrscheinlichkeiten im CHMM
Es sei $\mathcal{C}$ ein CHMM und $A \in \mathrm{Adv}_\mathcal{C}$. Wir verwenden die Notation aus Definition 5.3. Dann gilt für beliebige $x_i \in \Sigma_\mathcal{C}$ und $q' \in Q_\mathcal{C}$:

$$\big(P(Q^A_j = q,\ X^A_j = x \mid Q^A_{j-1} = q',\ X^A_1 \dots X^A_{j-1} = x_1 \dots x_{j-1})\big)_{x \in \Sigma_\mathcal{C}, q \in Q_\mathcal{C}} \in \overline{\overline{\mathcal{C}}}_{q'},$$

sofern $P(Q^A_{j-1} = q',\ X^A_1 \dots X^A_{j-1} = x_1 \dots x_{j-1}) > 0$, wobei $\overline{\overline{\mathcal{C}}}_{q'}$ den Abschluß der konvexen Hülle von $\mathcal{C}_{q'}$ meine. $\square$

Der Beweis zu diesem Hilfsatz findet sich auf Seite 81.

Beweis (zu Satz 5.5): Nach Lemma 5.7 können wir o. B. d. A. alle $\mathcal{C}_q$ $(q \in Q_\mathcal{C})$ als konvex annehmen.

Es seien für $A \in \mathrm{Adv}_\mathcal{C}$, $1 \le i \le j$, $q, q' \in Q_\mathcal{C}$

$$\begin{aligned} p^{(A,i,j)}_q &:= P(Q^A_j = q,\ X^A_i \dots X^A_j = x_i \dots x_j \mid X^A_1 \dots X^A_{i-1} = x_1 \dots x_{i-1}), \\ p^{(A,0,0)}_q &:= P(Q^A_j = q), \\ t^{(A,q',j)}_{x,q} &:= P(Q^A_j = q,\ X^A_j = x \mid Q^A_{j-1} = q',\ X^A_1 \dots X^A_{j-1} = x_1 \dots x_{j-1}), \end{aligned}$$

und für $1 \le i \le j$ und $i = 0$, $j = 0$ sei

$$\mathcal{P}^0_{i,j} := \{p^{(A,i,j)} : A \in \mathrm{Adv}_\mathcal{C},\ p^{(A,i,j)} \ne \perp\},$$

und $\mathcal{P}^*_{i,j}$ der Abschluß von $\mathcal{P}^0_{i,j}$.

Wir wollen nun zeigen, daß

$$\mathcal{P}^*_{0,0} = \mathbb{R}^{Q_\mathcal{C}}_1, \tag{61}$$
$$\mathcal{P}^*_{j,j} = \mathcal{T}^{\mathcal{C}}_{x_j} \circ \mathcal{N}(\mathcal{P}^*_{i,j-1}) \qquad (1 \le i < j \text{ oder } i = 0,\ j = 1), \tag{62}$$
$$\mathcal{P}^*_{i,j} = \mathcal{T}^{\mathcal{C}}_{x_j}(\mathcal{P}^*_{i,j-1}) \qquad (1 \le i < j). \tag{63}$$

Haben wir dies gezeigt, so wissen wir, daß $\mathcal{P}_{i,j} = \mathcal{P}^*_{i,j}$ für $1 \le i \le j$, und daß (11) wohldefiniert ist.

Nach Definitionen 5.2 und 5.3 ist (61) klar.

Wir wollen nun $\mathcal{P}^*_{j,j} \subseteq \mathcal{T}^{\mathcal{C}}_{x_j} \circ \mathcal{N}(\mathcal{P}^*_{i,j-1})$ für $1 \le i < j$ und für $i = 0$, $j = 1$ zeigen. Sei $\tilde{p} \in \mathcal{P}^0_{j,j}$. Dann existiert ein $A \in \mathrm{Adv}_{\mathcal{C}}$, so daß $P(X^A_1 \dots X^A_{j-1} = x_1 \dots x_{j-1}) > 0$ und $\tilde{p} = p^{(A,j,j)}$. Damit gilt für $q \in Q_{\mathcal{C}}$

$$\begin{aligned}
p^{(A,j,j)}_q &= P(Q^A_j = q,\ X^A_j = x_j \mid X^A_1 \dots X^A_{j-1} = x_1 \dots x_{j-1}) \\
&= \sum_{q' \in Q_{\mathcal{C}}} P(Q^A_j = q,\ Q^A_{j-1} = q',\ X^A_j = x_j \mid X^A_1 \dots X^A_{j-1} = x_1 \dots x_{j-1}) \\
&= \sum_{q' \in Q_{\mathcal{C}}} P(Q^A_j = q,\ X^A_j = x_j \mid Q^A_{j-1} = q',\ X^A_1 \dots X^A_{j-1} = x_1 \dots x_{j-1}) \cdot \\
&\qquad\qquad P(Q^A_{j-1} = q' \mid X^A_1 \dots X^A_{j-1} = x_1 \dots x_{j-1}) \\
&= \sum_{q' \in Q_{\mathcal{C}}} t^{(A,q',j)}_{x_j,q}\, P(Q^A_{j-1} = q' \mid X^A_1 \dots X^A_{j-1} = x_1 \dots x_{j-1}) \\
&= \sum_{q' \in Q_{\mathcal{C}}} t^{(A,q',j)}_{x_j,q} \cdot \begin{cases} P(Q^A_0 = q'), & (j = 1), \\ \frac{P(Q^A_{j-1} = q',\, X^A_i \dots X^A_{j-1} = x_i \dots x_{j-1} \mid X^A_1 \dots X^A_{i-1} = x_1 \dots x_{i-1})}{P(X^A_i \dots X^A_{j-1} = x_i \dots x_{j-1} \mid X^A_1 \dots X^A_{i-1} = x_1 \dots x_{i-1})}, & (1 \le i < j) \end{cases} \\
&= \sum_{q' \in Q_{\mathcal{C}}} t^{(A,q',j)}_{x_j,q} \frac{p^{(A,i,j-1)}_{q'}}{\|p^{(A,i,j-1)}\|_1}.
\end{aligned} \tag{64}$$

Es gilt $p^{(A,i,j-1)} \in \mathcal{P}^*_{i,j-1}$, und mit Hilfsatz A.28 ist $t^{(A,q',j)} \in \overline{\overline{\mathcal{C}}}_{q'}$, also nach vorstehender Rechnung $\tilde{p} = p^{(A,j,j)}_q \in \mathcal{T}^{\mathcal{C}}_{x_j} \circ \mathcal{N}(\mathcal{P}^*_{i,j-1})$, womit $\mathcal{P}^0_{j,j} \subseteq \mathcal{T}^{\mathcal{C}}_{x_j} \circ \mathcal{N}(\mathcal{P}^*_{i,j-1})$ folgt. Aufgrund der stetigen Natur der in der Definition von $\mathcal{T}^{\mathcal{C}}_{x_j}$ und $\mathcal{N}$ verwendeten Abbildungen und der Abgeschlossenheit von $\mathcal{P}^*_{i,j-1}$ und $\overline{\overline{\mathcal{C}}}_{q'}$ ist auch $\mathcal{T}^{\mathcal{C}}_{x_j} \circ \mathcal{N}(\mathcal{P}^*_{i,j-1})$ abgeschlossen, also haben wir sogar $\mathcal{P}^*_{j,j} \subseteq \mathcal{T}^{\mathcal{C}}_{x_j} \circ \mathcal{N}(\mathcal{P}^*_{i,j-1})$.

Nun wollen wir $\mathcal{T}^{\mathcal{C}}_{x_j} \circ \mathcal{N}(\mathcal{P}^*_{i,j-1}) \subseteq \mathcal{P}^*_{j,j}$ für $1 \le i < j$ und für $i = 0$, $j = 1$ beweisen.

Es sei dazu $\tilde{p} \in \mathcal{T}^{\mathcal{C},0}_{x_j} \circ \mathcal{N}(\mathcal{P}^0_{i,j-1})$ mit

$$\mathcal{T}^{\mathcal{C},0}_{x_j}(M) := \Big\{ \Big(\sum_{q' \in Q_{\mathcal{C}}} t^{(q')}_{x_j,q} p_{q'} \Big)_q \ :\ t^{(q')} \in \mathcal{C}_{q'},\ p \in M \Big\}.$$

Dann existieren ein $A \in \mathrm{Adv}_{\mathcal{C}}$ und $t^{(q')} \in \mathcal{C}_{q'}$ $(q' \in Q_{\mathcal{C}})$, so daß für alle $q \in Q_{\mathcal{C}}$ gilt:

$$\tilde{p}_q = \sum_{q' \in Q_{\mathcal{C}}} t^{(q')}_{x_j,q} \frac{p^{(A,i,j-1)}_{q'}}{\|p^{(A,i,j-1)}\|_1}.$$

Wir konstruieren dann ein $\tilde{A} \in \mathrm{Adv}_{\mathcal{C}}$ mit $\tilde{A}^* := A^*$ und

$$\tilde{A}(\nu, r', q', (q_\mu), (x_\mu)) := \begin{cases} A(\nu, r', q', (q_\mu), (x_\mu)), & \text{für } \nu < j, \\ t^{(q')}, & \text{für } \nu = j, \\ \text{beliebig}, & \text{für } \nu > j. \end{cases} \tag{65}$$

Es ist dann $p^{(\tilde{A},i,j-1)} = p^{(A,i,j-1)}$, da diese nur von A bzw. $\tilde{A}$ mit erstem Argument $\nu < j$ abhängen. Außerdem ist $t^{(\tilde{A},q',j)} = t^{(q')}$ nach Konstruktion von $\tilde{A}$. Dies erlaubt, wie folgt zu rechnen:

$$\tilde{p}_q \ = \sum_{q' \in Q_{\mathcal{C}}} t^{(q')}_{x_j,q} \frac{p^{(A,i,j-1)}_{q'}}{\|p^{(A,i,j-1)}\|_1} \ = \sum_{q' \in Q_{\mathcal{C}}} t^{(\tilde{A},q',j)}_{x_j,q} \frac{p^{(\tilde{A},i,j-1)}_{q'}}{\|\tilde{p}^{(A,i,j-1)}\|_1} \overset{(64)}{=} p^{(\tilde{A},j,j)}_q, \tag{66}$$

also ist $\tilde{p} = p^{(\tilde{A},j,j)} \in \mathcal{P}^*_{j,j}$. Damit wissen wir $\mathcal{T}^{\mathcal{C},0}_{x_j} \circ \mathcal{N}(\mathcal{P}^0_{i,j-1}) \subseteq \mathcal{P}^*_{j,j}$. Mit dem gleichen Stetigkeits- und Abgeschlossenheitsargument wie oben erkennen wir, daß $\mathcal{T}^{\mathcal{C}}_{x_j} \circ \mathcal{N}(\mathcal{P}^*_{i,j-1})$ der Abschluß von $\mathcal{T}^{\mathcal{C},0}_{x_j} \circ \mathcal{N}(\mathcal{P}^0_{i,j-1})$ ist – man beachte hierbei, daß die $\mathcal{C}_{q'}$ konvex sind –, somit ist auch $\mathcal{T}^{\mathcal{C}}_{x_j} \circ \mathcal{N}(\mathcal{P}^*_{i,j-1}) \subseteq \mathcal{P}^*_{j,j}$, es folgt (62).

Nun wenden wir uns (63) zu und zeigen $\mathcal{P}^*_{i,j} \subseteq \mathcal{T}^{\mathcal{C}}_{x_j}(\mathcal{P}^*_{i,j-1})$ für $1 \le i < j$. Sei wieder $\tilde{p} \in \mathcal{P}^0_{i,j}$, dann gibt es wieder einen Adversary $A \in \mathrm{Adv}_{\mathcal{C}}$, so daß $P(X^A_1 \dots X^A_{i-1} = x_1 \dots x_{i-1}) > 0$ und $\tilde{p} = p^{(A,i,j)}$. Damit gilt für $q \in Q_{\mathcal{C}}$:

$$\begin{aligned}
p_q^{(A,i,j)} &= P(Q_j^A = q,\ X_i^A \dots X_j^A = x_i \dots x_j \mid X_1^A \dots X_{i-1}^A = x_1 \dots x_{i-1}) \\
&= \sum_{q' \in Q_{\mathcal{C}}} P(Q_j^A = q,\ Q_{j-1}^A = q',\ X_i^A \dots X_j^A = x_i \dots x_j \mid X_1^A \dots X_{i-1}^A = x_1 \dots x_{i-1}) \\
&= \sum_{q' \in Q_{\mathcal{C}}} P(Q_j^A = q,\ X_j = x_j \mid Q_{j-1}^A = q',\ X_1^A \dots X_{j-1}^A = x_1 \dots x_{j-1}) \\
&\qquad\qquad P(Q_{j-1} = q',\ X_i \dots X_{j-1} = x_i \dots x_{j-1} \mid X_1 \dots X_{i-1} = x_1 \dots x_{i-1}) \\
&= \sum_{q' \in Q_{\mathcal{C}}} t_{x_j,q}^{(A,q',j)}\, p_{q'}^{(A,i,j-1)}. \qquad (67)
\end{aligned}$$

Da $p^{(A,i,j-1)} \in \mathcal{P}_{i,j-1}^*$, und $t^{(A,q',j)} \in \overline{\overline{\mathcal{C}}}_{q'}$ wegen Hilfsatz A.28, haben wir $\tilde{p} = p_q^{(A,i,j)} \in \mathcal{T}_{x_j}^{\mathcal{C}}(\mathcal{P}_{i,j-1}^*)$, woraus mit obigen Stetigkeits- und Abgeschlossenheitsüberlegungen $\mathcal{P}_{i,j}^* \subseteq \mathcal{T}_{x_j}^{\mathcal{C}}(\mathcal{P}_{i,j-1}^*)$ folgt.

Um $\mathcal{T}_{x_j}^{\mathcal{C}}(\mathcal{P}_{i,j-1}^*) \subseteq \mathcal{P}_{i,j}^*$ für $1 \le i < j$ zu zeigen, wählen wir wieder ein beliebiges $\tilde{p} \in \mathcal{T}_{x_j}^{\mathcal{C},0}(\mathcal{P}_{i,j-1}^0)$. Dann gibt es $A \in \mathrm{Adv}_{\mathcal{C}}$ und $t^{(q')} \in \mathcal{C}_{q'}$ $(q' \in Q_{\mathcal{C}})$, so daß für alle $q \in Q_{\mathcal{C}}$ gilt:

$$\tilde{p}_q = \sum_{q' \in Q_{\mathcal{C}}} t_{x_j,q}^{(q')}\, p_{q'}^{(A,i,j-1)}.$$

Wieder sei $\tilde{A} \in \mathrm{Adv}_{\mathcal{C}}$ durch (65) definiert, und wir folgern analog zu (66):

$$\tilde{p}_q \;=\; \sum_{q' \in Q_{\mathcal{C}}} t_{x_j,q}^{(q')}\, p_{q'}^{(A,i,j-1)} \;=\; \sum_{q' \in Q_{\mathcal{C}}} t_{x_j,q}^{(\tilde{A},q',j)}\, p_{q'}^{(\tilde{A},i,j-1)} \overset{(67)}{=} p_q^{(\tilde{A},i,j)}. \qquad (68)$$

Folglich ist $\tilde{p} = p^{(\tilde{A},i,j)} \in \mathcal{P}_{i,j}^*$ und mit den gleichen Stetigkeits- und Abgeschlossenheitsüberlegungen wie oben folgt $\mathcal{T}_{x_j}^{\mathcal{C}}(\mathcal{P}_{i,j-1}^*) \subseteq \mathcal{P}_{i,j}^*$, und (63) ist bewiesen.

Zuletzt erkennen wir noch, daß für $1 \le i \le j$:

$$\begin{aligned}
&\eta^{\mathcal{C}}(x_1 \dots x_{i-1};\ x_i \dots x_j) \\
&\quad = -\log \sup_{A \in \mathrm{Adv}_{\mathcal{C}}} P(X_i^A \dots X_j^A = x_i \dots x_j \mid X_1^A \dots X_{i-1}^A = x_1 \dots x_{i-1}) \\
&\quad = -\log \sup_{A \in \mathrm{Adv}_{\mathcal{C}}} \sum_{q \in Q_{\mathcal{C}}} P(Q_i^A = q,\ X_i^A \dots X_j^A = x_i \dots x_j \mid X_1^A \dots X_{i-1}^A = x_1 \dots x_{i-1}) \\
&\quad = -\log \sup_{A \in \mathrm{Adv}_{\mathcal{C}}} \|p^{(A,i,j)}\|_1 = -\log \sup_{p \in \mathcal{P}_{i,j}^0} \|p\|_1 = -\log \max_{p \in \mathcal{P}_{i,j}^*} \|p\|_1 \\
&\quad = -\log \max_{p \in \mathcal{P}_{i,j}} \|p\|_1. \qquad \blacksquare
\end{aligned}$$

Beweis (zu Hilfsatz A.28): Wir interpretieren $\mathbb{R}_1^{\Sigma_{\mathcal{C}} \times Q_{\mathcal{C}}}$ als Teilmenge des $\mathbb{R}^m$ mit $m := \#\Sigma_{\mathcal{C}} \cdot \#Q_{\mathcal{C}}$. Es sei $C := \overline{\overline{\mathcal{C}}}_{q'}$. Nach Konstruktion ist $C \subseteq \mathbb{R}^m$ konvex und abgeschlossen.

Unter einem offenen Halbraum $H_{n,d}$ (im folgenden kurz Halbraum) verstehen wir eine Menge $\{x \in \mathbb{R}^m : n^T x < d\}$ mit $n \in \mathbb{R}^m \setminus \{0\}$, $d \in \mathbb{R}$. Es sei $\mathcal{H}$ die Menge aller Halbräume H mit $C \cap H = \emptyset$. Wir behaupten, daß

$$\mathbb{R}^m \setminus \bigcup_{H \in \mathcal{H}} H = C. \qquad (69)$$

Um (69) zu zeigen, sei $x \in \mathbb{R}^m \setminus C$. Dann müssen wir zeigen, daß $x \in H$ für ein $H \in \mathcal{H}$. Wähle $y \in C$ so, daß $\|x - y\|_2$ minimal (das Minimum existiert, da C geschnitten mit einer hinreichend großen Kugel um x nichtleer und kompakt ist). Es sei dann H ein Halbraum, dessen Rand orthogonal auf xy steht, und für den $x \in H$, $y \notin H$ gilt. Wir müssen nun zeigen, daß $H \cap C = \emptyset$ (damit $H \in \mathcal{H}$). Nehmen wir hierzu an, es gebe ein $z \in H \cap C$. Da $z \in H$, gibt es ein $w \in \overline{yz}$ (der Strecke zwischen y und z), so daß $\|x - w\|_2 < \|x - y\|_2$. Da $y, z \in C$ liegen und C kompakt ist, ist auch $w \in C$, damit ist aber $\|x - y\|_2$ nicht minimal, es liegt ein Widerspruch vor. Somit gilt (69).

Sei nun $H_{n,d} \in \mathcal{H}$. Wir wollen zeigen, daß

$$\big(P(Q_j^A = q,\ X_j^A = x \mid Q_{j-1}^A = q',\ X_1^A \dots X_{j-1}^A = x_1 \dots x_{j-1})\big)_{x \in \Sigma_\mathcal{C}, q \in Q_\mathcal{C}} \notin H_{n,d}. \tag{70}$$

Es gelte die Notation aus Definition 5.3. Sei

$$\begin{aligned} M := \Big\{ &(r_0, \dots, r_{j-1}, r_0', r_1', \dots) \in [0,1)^{\mathbb{N}_0} : \\ &P_{(R_0,\dots,R_{j-1},R_0',R_1',\dots)=(r_0,\dots,r_{j-1},r_0',r_1',\dots)}\big(Q_{j-1}^A = q',\ X_1^A \dots X_{j-1}^A = x_1 \dots x_{j-1}\big) = 1 \Big\}. \end{aligned}$$

Man beachte, daß die Wahrscheinlichkeit in dieser Gleichung für beliebige Werte für $r_0, \dots, r_{j-1}$ und (r_i') nur 0 oder 1 sein kann, da Q_{j-1}^A und $X_1^A \dots X_{j-1}^A$ vollständig durch $R_0, \dots, R_{j-1}$ und R' determiniert sind.

Es ist nun

$$\begin{aligned} 1 &= P\big(T_j^A \in C \supseteq \mathcal{C}_{q'} \mid Q_{j-1}^A = q',\ X_1^A \dots X_{j-1}^A = x_1 \dots x_{j-1}\big) \\ &= P\big(T_j^A \in C \mid (R_0, \dots, R_{j-1}, R_0', R_1', \dots) \in M\big) \\ &\le P\big(T_j^A \notin H_{n,d} \mid (R_0, \dots, R_{j-1}, R_0', R_1', \dots) \in M\big) \\ &= P\big(n^T T_j^A \ge d \mid (R_0, \dots, R_{j-1}, R_0', R_1', \dots) \in M\big). \end{aligned}$$

Mit

$$E := \mathrm{E}\big(T_j^A | (R_0, \dots, R_{j-1}, R_0', R_1', \dots)\big)$$

folgt daraus

$$n^T E = \mathrm{E}\big(n^T T_j^A | (R_0, \dots, R_{j-1}, R_0', R_1', \dots) \in M\big) \ge d,$$

also $E \notin H_{n,d}$. Da dies für alle $H_{n,d} \in \mathcal{H}$ gilt, erhalten wir nach (69) sofort $E \in C$.

Schließlich es sei $t = t(r_0, \dots, r_{j-1}, r_0', \dots) \in \mathbb{R}^m$ der Wert mit

$$P\big(T_j^A = t \mid (R_0, \dots, R_{j-1}, R_0', \dots) = (r_0, \dots, r_{j-1}, r_0', \dots)\big) = 1$$

(dieser existiert, da T_j^A nur von $R_0, \dots, R_{j-1}, R_0', \dots$ abhängt).

Wir erhalten dann für $x \in \Sigma_\mathcal{C}$, $q \in Q_\mathcal{C}$

$$\begin{aligned} &P(Q_j^A = q,\ X_j^A = x \mid Q_{j-1}^A = q',\ X_1^A \dots X_{j-1}^A = x_1 \dots x_{j-1}) \\ &\quad = \frac{\int_M P_{(R_0,\dots,R_{j-1},R_0',R_1',\dots)=(r_0,\dots,r_{j-1},r_0',r_1',\dots)}(Q_j^A = q,\ X_j^A = x)\, d(r_0, \dots, r_{j-1}, r_0', r_1', \dots)}{P\big((R_0, \dots, R_{j-1}, R_0', R_1', \dots) \in M\big)} \\ &\quad \overset{5.3}{=} \frac{\int_M t(r_0, \dots, r_{j-1}, r_0', r_1', \dots)_{x,q}\ d(r_0, \dots, r_{j-1}, r_0', r_1', \dots)}{P\big((R_0, \dots, R_{j-1}, R_0', R_1', \dots) \in M\big)} \\ &\quad = \frac{\mathrm{E}\big((T_j^A)_{x,q} \cdot (\delta(R_0, \dots, R_{j-1}, R_0', R_1', \dots) \in M)\big)}{P\big((R_0, \dots, R_{j-1}, R_0', R_1', \dots) \in M\big)} \\ &\quad = E_{x,q}. \end{aligned}$$

Damit ist

$$\big(P(Q_j^A = q,\ X_j^A = x \mid Q_{j-1}^A = q',\ X_1^A \dots X_{j-1}^A = x_1 \dots x_{j-1})\big)_{x \in \Sigma_\mathcal{C}, q \in Q_\mathcal{C}} = E \in C = \overline{\overline{\mathcal{C}}}_{q'}. \qquad \blacksquare$$

A.5.7 Lemma 5.7

Lemma 5.7: Konvex-äquivalente CHMM
Sind $\mathcal{C}$ und $\mathcal{C}'$ konvex-äquivalente CHMM, so ist $\mathcal{X}^\mathcal{C} = \mathcal{X}^{\mathcal{C}'}$. $\square$

Beweis: O. B. d. A. seien alle $\mathcal{C}_q'$ konvex. Dann ist zu zeigen, daß $\mathcal{X}^{\mathcal{C}'} \subseteq \mathcal{X}^\mathcal{C}$.

Sei $A \in \mathrm{Adv}_{\mathcal{C}'}$. Wir konstruieren dann $\tilde{A} \in \mathrm{Adv}_{\mathcal{C}}$ wie folgt:

$$\begin{aligned} \tilde{A}^*(r') &:= A^*\big((r'_{2\nu})_\nu\big), \\ \tilde{A}(i, r', q, (q_\nu), (\sigma_\nu)) &:= g_q\big(r'_{2i+1}, A(i, (r'_{2\nu})_\nu, q, (q_\nu), (\sigma_\nu))\big) \end{aligned} \tag{71}$$

für $r' \in [0,1)^{\mathbb{N}_0}$, $i \in \mathbb{N}$, $q \in Q_{\mathcal{C}}$, $(q_\nu) \in Q^*_{\mathcal{C}}$, $(\sigma_\nu) \in \Sigma^*_{\mathcal{C}}$.

Es bleibt die Funktion $g_q : [0,1) \times \mathcal{C}'_q \to \mathcal{C}_q$ zu definieren. Seien $q \in Q_{\mathcal{C}}$, $r \in [0,1)$, $p \in \mathcal{C}'_q$. Dann wähle $n^{(q,p)} \in \mathbb{N}$, $a_\nu^{(q,p)} \in [0,1]$, $p_\nu^{(q,p)} \in \mathcal{C}_q$ $(\nu = 1, \ldots, n^{(q,p)})$ so, daß

$$\sum_{\nu=1}^{n^{(q,p)}} a_\nu^{(q,p)} p_\nu^{(q,p)} = p \qquad \text{und} \qquad \sum_{\nu=1}^{n^{(q,p)}} a_\nu^{(q,p)} = 1. \tag{72}$$

Dies ist möglich, da $\mathcal{C}'_q$ in der konvexen Hülle von $\mathcal{C}_q$ liegt. Es sei dann

$$M_\nu^{(q,p)} := \left[\sum_{\mu=1}^{\nu-1} a_\mu^{(q,p)},\ \sum_{\mu=1}^{\nu} a_\mu^{(q,p)}\right) \qquad (\nu = 1, \ldots, n^{(q,p)}) \tag{73}$$

und

$$g_q(r,p) := p_\nu^{(q,p)} \qquad \Longleftrightarrow \qquad r \in M_\nu^{(q,p)} \qquad (\nu = 1, \ldots, n^{(q,p)}). \tag{74}$$

Wir behaupten nun, daß X^A und $X^{\tilde{A}}$ von gleicher Verteilung sind.

Seien $q_\nu \in Q_{\mathcal{C}}$, $x_\nu \in \Sigma_{\mathcal{C}}$ und $r' \in [0,1)^{\mathbb{N}_0}$. Zunächst wollen wir induktiv für alle $i \in \mathbb{N}_0$ zeigen, daß

$$\begin{aligned} &P_{(R'_{2\nu})_\nu = r'}(Q_0^{\tilde{A}} \ldots Q_i^{\tilde{A}} = q_0 \ldots q_i,\ X_1^{\tilde{A}} \ldots X_i^{\tilde{A}} = x_1 \ldots x_i) \\ &\qquad = P_{R'=r'}(Q_0^A \ldots Q_i^A = q_0 \ldots q_i,\ X_1^A \ldots X_i^A = x_1 \ldots x_i). \end{aligned} \tag{75}$$

Für $i = 0$ folgt die Aussage direkt aus der Definition von $\tilde{A}^*$. Sie gelte nun für $i-1$, wir wollen sie für i zeigen. Es sei $p := A(i, r', q, (q_\nu), (x_\nu))$, dann erhalten wir

$$\begin{aligned} &P_{(R'_{2\nu})_\nu = r'}(Q_0^{\tilde{A}} \ldots Q_i^{\tilde{A}} = q_0 \ldots q_i,\ X_1^{\tilde{A}} \ldots X_i^{\tilde{A}} = x_1 \ldots x_i) \\ &\overset{5.3}{=} \sum_{\nu=1}^{n^{(q_{i-1},p)}} P_{(R'_{2\nu})_\nu = r'}\big(R'_{2i+1} \in M_\nu^{(q_{i-1},p)}\big) P_{(R'_{2\nu})_\nu = r'}\big(Q_0^{\tilde{A}} \ldots Q_{i-1}^{\tilde{A}} = q_0 \ldots q_{i-1},\ X_1^{\tilde{A}} \ldots X_{i-1}^{\tilde{A}} = x_1 \ldots x_{i-1}\big) \\ &\qquad\qquad P_{(R'_{2\nu})_\nu = r'}\big(Q_i^{\tilde{A}} = q_i,\ X_i^{\tilde{A}} = x_i \mid R'_{2i+1} \in M_\nu^{(q_{i-1},p)}, \\ &\qquad\qquad\qquad Q_0^{\tilde{A}} \ldots Q_{i-1}^{\tilde{A}} = q_0 \ldots q_{i-1},\ X_1^{\tilde{A}} \ldots X_{i-1}^{\tilde{A}} = x_1 \ldots x_{i-1}\big) \\ &\overset{(73),\mathrm{IV}}{=} \sum_{\nu=1}^{n^{(q_{i-1},p)}} a_\nu^{(q_{i-1},p)}\ P_{R'=r'}\big(Q_0^A \ldots Q_{i-1}^A = q_0 \ldots q_{i-1},\ X_1^A \ldots X_{i-1}^A = x_1 \ldots x_{i-1}\big) \\ &\qquad\qquad P_{(R'_{2\nu})_\nu = r'}\big(Q_i^{\tilde{A}} = q_i,\ X_i^{\tilde{A}} = x_i \mid R'_{2i+1} \in M_\nu^{(q_{i-1},p)}, \\ &\qquad\qquad\qquad Q_0^{\tilde{A}} \ldots Q_{i-1}^{\tilde{A}} = q_0 \ldots q_{i-1},\ X_1^{\tilde{A}} \ldots X_{i-1}^{\tilde{A}} = x_1 \ldots x_{i-1}\big) \\ &\overset{(71),(74)}{=} \sum_{\nu=1}^{n^{(q_{i-1},p)}} a_\nu^{(q_{i-1},p)}\ P_{R'=r'}\big(Q_0^A \ldots Q_{i-1}^A = q_0 \ldots q_{i-1},\ X_1^A \ldots X_{i-1}^A = x_1 \ldots x_{i-1}\big)\ (p_\nu^{(q_{i-1},p)})_{x_i,q_i} \\ &\overset{(72)}{=} P_{R'=r'}\big(Q_0^A \ldots Q_{i-1}^A = q_0 \ldots q_{i-1},\ X_1^A \ldots X_{i-1}^A = x_1 \ldots x_{i-1}\big)\ \big(A(i, r', q, (q_\nu), (x_\nu))\big)_{x_i,q_i} \\ &= P_{R'=r'}\big(Q_0^A \ldots Q_{i-1}^A = q_0 \ldots q_{i-1},\ X_1^A \ldots X_{i-1}^A = x_1 \ldots x_{i-1}\big) \\ &\qquad P_{R'=r'}\big(Q_i^A = q_i,\ X_i^A = x_i \mid Q_0^A \ldots Q_{i-1}^A = q_0 \ldots q_{i-1},\ X_1^A \ldots X_{i-1}^A = x_1 \ldots x_{i-1}\big) \\ &= P_{R'=r'}\big(Q_0^A \ldots Q_i^A = q_0 \ldots q_i,\ X_1^A \ldots X_i^A = x_1 \ldots x_i\big). \end{aligned}$$

Damit ist (75) bewiesen.

Es folgt aus (75) durch Integration

$$P(Q_0^{\tilde{A}} \ldots Q_i^{\tilde{A}} = q_0 \ldots q_i,\ X_1^{\tilde{A}} \ldots X_i^{\tilde{A}} = x_1 \ldots x_i) = P(Q_0^A \ldots Q_i^A = q_0 \ldots q_i,\ X_1^A \ldots X_i^A = x_1 \ldots x_i),$$

insbesondere

$$P(X_1^{\tilde{A}} \dots X_i^{\tilde{A}} = x_1 \dots x_i) = P(X_1^A \dots X_i^A = x_1 \dots x_i),$$

es haben X^A und $X^{\tilde{A}}$ die gleiche Verteilung, also $X^A \in \mathcal{X}^{\mathcal{C}}$. ■

A.5.8 Lemma 5.9

Definition 5.8: Endlich repräsentierbare CHMM

Ein CHMM $\mathcal{C}$ heißt *endlich,* wenn alle $\mathcal{C}_q$ $(q \in Q_{\mathcal{C}})$ endlich sind.

Ein CHMM $\mathcal{C}$ heißt *endlich repräsentierbar,* wenn ein endliches CHMM $\mathcal{C}'$ existiert, welches konvex-äquivalent zu $\mathcal{C}$ ist.

Ein CHMM $\mathcal{C}$ heißt *fast endlich repräsentierbar,* wenn ein endliches CHMM $\mathcal{C}'$ existiert, welches fast konvex-äquivalent zu $\mathcal{C}$ ist. □

Lemma 5.9: Repräsentierbarkeit von durch Diagramme definierten CHMM

Läßt sich ein CHMM durch ein Diagramm mit beigefügten Gleichungen und Ungleichungen darstellen (wie vor Definition 5.1 erläutert), und sind diese Gleichungen und Ungleichungen linear, sowie alle an den Pfeilen notierten Wahrscheinlichkeitsmengen Intervalle, so ist das CHMM fast endlich repräsentierbar.

Sind zusätzlich alle an den Pfeilen angegebenen Wahrscheinlichkeitsmengen abgeschlossen, und kommen in den Gleichungen und Ungleichungen nur die Relationen $\leq$, $\geq$ und $=$ vor (nicht $<$ oder $>$), so ist das CHMM sogar endlich repräsentierbar. □

Beweis: Sei das CHMM $\mathcal{C}$ wie im Lemma beschrieben darstellbar. Es bezeichne $M_{q,x,q'}$ $(q, q' \in Q_{\mathcal{C}},\ x \in \Sigma_{\mathcal{C}})$ die Menge der für den Pfeil von q nach q' mit Symbol x zugelassenen Wahrscheinlichkeiten. Falls der Pfeil mit einer freien Variable (die in den Gleichungen/Ungleichungen vorkommt) beschriftet ist, sei $M_{q,x,q'} = [0, 1]$. Alle $M_{q,x,q'}$ sind nach Voraussetzung Intervalle, somit konvex.

Sei $G_i^q \subseteq \mathbb{R}^{\Sigma_{\mathcal{C}} \times Q_{\mathcal{C}}}$ der Lösungsraum der Gleichung g_i, projiziert auf die an von $q \in Q_{\mathcal{C}}$ ausgehenden Pfeilen stehenden Variablen. Dann ist G_i ein Hyperebene, also konvex und abgeschlossen.

Sei $U_i^q \subseteq \mathbb{R}^{\Sigma_{\mathcal{C}} \times Q_{\mathcal{C}}}$ der Lösungsraum der Ungleichung u_i, projiziert auf die an von $q \in Q_{\mathcal{C}}$ ausgehenden Pfeilen stehenden Variablen. Dann ist U_i ein Halbraum, also konvex und – falls nur die Relationen $\leq$, $\geq$ und $=$ vorkommen – abgeschlossen.

Wir erhalten

$$\mathcal{C}_q = \mathbb{R}_1^{\Sigma_{\mathcal{C}} \times Q_{\mathcal{C}}} \cap \prod_{\substack{x \in \Sigma_{\mathcal{C}} \\ q' \in Q_{\mathcal{C}}}} M_{q,x,q'} \cap \bigcap_i G_i^q \cup \bigcap_i U_i^q. \tag{76}$$

Es ist $\mathbb{R}_1^{\Sigma_{\mathcal{C}} \times Q_{\mathcal{C}}}$ eine konvexe Menge, deren Abschluß endlich erzeugt ist (er wird von den Einheitsvektoren aufgespannt), diese Eigenschaft bleibt erhalten bei Schnitt mit einem Quader $(\prod_{\substack{x \in \Sigma_{\mathcal{C}} \\ q' \in Q_{\mathcal{C}}}} M_{q,x,q'})$, mit einer Hyperebene (G_i^q) oder einem Halbraum (U_i^q) erhalten. Also ist $\mathcal{C}_q$ konvex und der Abschluß von $\mathcal{C}_q$ ebenfalls endlich erzeugt. Damit ist $\mathcal{C}$ fast endlich repräsentierbar.

Treffen noch die zusätzlichen Voraussetzungen aus der zweiten Hälfte des Lemmas zu, so sind alle Mengen auf der rechten Seite von (76) abgeschlossen, damit ist es auch $\mathcal{C}_q$. Damit nicht nur der Abschluß von $\mathcal{C}_q$ endlich erzeugt, sondern auch $\mathcal{C}_q$ selbst. ■

A.5.9 Lemma 5.10

Lemma 5.10: Konvexität der Rekursion in Satz 5.5

Sind $\mathcal{C}$ und $\mathcal{C}'$ zwei fast konvex-äquivalente CHMM, $x \in \Sigma_{\mathcal{C}}$, $\mathcal{T}_x$ und $\mathcal{N}$ wie in Satz 5.5, sowie $\mathcal{P}, \mathcal{P}' \subseteq \mathbb{R}_1^{Q_{\mathcal{C}}}$ konvex-äquivalent, dann sind

$$\mathcal{N}(\mathcal{P}) \approx \mathcal{N}(\mathcal{P}'), \qquad \mathcal{T}_x^{\mathcal{C}}(\mathcal{P}) \approx \mathcal{T}_x^{\mathcal{C}'}(\mathcal{P}') \qquad \text{und} \qquad -\log \sup_{p \in \mathcal{P}} \|p\|_1 = -\log \sup_{p \in \mathcal{P}'} \|p\|_1,$$

wobei $\approx$ konvexe Äquivalenz meine. □

Hilfsatz A.33: Konvexität einiger Operationen
Es sei V ein endlichdimensionaler Vektorraum, $A_i, B_i \subseteq V$ mit $A_i \approx B_i$ $(i = 1, \ldots, n)$, und $L : V \to V$ linear. Dann sind

$$\{La : a \in A_1\} \approx \{Lb : b \in B_1\}, \tag{77}$$

$$A_1 \times \{1\} \approx B_1 \times \{1\}, \tag{78}$$

$$\left\{\bigotimes_{i=1}^{n} a_i : a_i \in A_i\right\} \approx \left\{\bigotimes_{i=1}^{n} b_i : b_i \in B_i\right\}. \tag{79}$$

Hierbei bezeichne $\approx$ konvexe Äquivalenz und $\otimes$ das Kroneckerprodukt. □

Den Beweis dieses Hilfsatzes findet man auf Seite 85.

Beweis (zu Lemma 5.10): Es bezeichne $\bar{M}$ die konvexe Hülle von M.

O. B. d. A. seien $\mathcal{P}$ und $\mathcal{P}'$ nichtleer.

Sei $\tilde{p} \in \mathcal{N}(\mathcal{P})$. Dann existiert ein $p \in \mathcal{P}$ mit $\tilde{p} = c^{-1}p$, $c := \|p\|_1 \neq 0$. Da $\mathcal{P}$ und $\mathcal{P}'$ konvex-äquivalent sind, existieren endlich viele $p'_i \in \mathcal{P}' \setminus \{0\}$, $r_i \in [0, 1]$ mit $p = \sum r_i p'_i$, $\sum r_i \leq 1$. Wir setzen $c_i := \|p'_i\|_1$ und erhalten damit

$$\tilde{p} = \sum_i c^{-1} r_i c_i \tilde{p}'_i \tag{80}$$

mit $\tilde{p}'_i := p'_i / \|p'_i\|_1$. Da c, r_i, c_i und alle Koeffizienten von $\tilde{p}'_i$ nichtnegativ sind, gilt

$$1 = \|\tilde{p}\|_1 = \sum_i c^{-1} r_i c_i \|\tilde{p}'_i\|_1 = \sum_i c^{-1} r_i c_i,$$

damit ist (80) eine Konvexkombination über $\mathcal{N}(\mathcal{P}')$ (denn $\tilde{p}'_i \in \mathcal{N}(\mathcal{P}')$), also $\mathcal{N}(\mathcal{P}) \subseteq \overline{\mathcal{N}(\mathcal{P}')}$, woraus sich – da analog $\mathcal{N}(\mathcal{P}') \subseteq \overline{\mathcal{N}(\mathcal{P})}$ folgt – $\mathcal{N}(\mathcal{P}) \approx \mathcal{N}(\mathcal{P}')$ ergibt.

Sei $Q_\mathcal{C} = \{q_1, \ldots, q_n\}$. Für $p \in M$, $t^{(q')} \in \bar{\mathcal{C}}_q$ enthält der Vektor

$$(p, 1) \otimes \bigotimes_{q' \in \mathcal{C}} (t^{(q')}, 1)$$

alle Summanden von $\sum_{q' \in Q_\mathcal{C}} t^{(q')}_{x,q} p_{q'}$ als Komponenten (mit $q \in Q_\mathcal{C}$, $x \in \Sigma_\mathcal{C}$). Also existiert ein Endomorphismus L auf $(\mathbb{R}_1^{Q_\mathcal{C}} \times \{1\})^{\otimes \# Q_\mathcal{C} + 1}$, so daß

$$\mathcal{T}_x^\mathcal{C}(\mathcal{P}) = \left\{L\big((p, 1) \otimes \bigotimes_{q' \in \mathcal{C}} (t^{(q')}, 1)\big) : p \in \mathcal{P},\ t^{(q')} \in \bar{\mathcal{C}}_{q'}\right\}$$

Nach Hilfsatz A.33 ist damit $\mathcal{T}_x^\mathcal{C}(\mathcal{P}) \approx \mathcal{T}_x^\mathcal{C}(\mathcal{P}')$. Da $\mathcal{T}_x^\mathcal{C}$ nur vom Abschluß der konvexen Hülle der $\mathcal{C}_{q'}$ abhängt, ist außerdem $\mathcal{T}_x^\mathcal{C} = \mathcal{T}_x^{\mathcal{C}'}$, damit ist $\mathcal{T}_x^\mathcal{C}(\mathcal{P}) \approx \mathcal{T}_x^{\mathcal{C}'}(\mathcal{P}')$.

Da in $\mathbb{R}_1^{Q_\mathcal{C}}$ alle Komponenten nichtnegativ sind, ist $\|\cdot\|_1 : \mathbb{R}_1^{Q_\mathcal{C}} \to \mathbb{R}_{\geq 0}$ linear. Nach Hilfsatz A.33 sind somit $\mathcal{P}_N := \{\|p\|_1 : p \in \mathcal{P}\} \approx \{\|p\|_1 : p \in \mathcal{P}'\} =: \mathcal{P}'_N$. Damit ist

$$\sup \mathcal{P}_N = \max \bar{\mathcal{P}}_N = \max \bar{\mathcal{P}}'_N = \sup \mathcal{P}'_N,$$

und schließlich

$$-\log \sup_{p \in \mathcal{P}} \|p\|_1 = -\log \sup \mathcal{P}_N = -\log \sup \mathcal{P}'_N = -\log \sup_{p \in \mathcal{P}'} \|p\|_1. \qquad \blacksquare$$

Beweis (zu Hilfsatz A.33): Sei $\tilde{a} \in \{La : a \in A_1\}$. Für geeignete $b_i \in B_1$ und $r_i \in [0, 1]$ mit $\sum r_i = 1$ ist dann $\tilde{a} = L \sum r_i b_i = \sum r_i L b_i$ eine Konvexkombination über $\{Lb : b \in B_1\}$, also $\{La : a \in A_1\} \subseteq \overline{\{Lb : b \in B_1\}}$. Damit gilt (77).

Sei $(a,1) \in A_1 \times \{1\}$. Für geeignete $b_i \in B_1$ und $r_i \in [0,1]$ mit $\sum r_i = 1$ ist dann $a = \sum r_i b_i$, also ist $(a,1) = \left(\sum r_i b_i, \sum r_i \cdot 1\right) = \sum r_i (b_i, 1)$ eine Konvexkombination über $B_1 \times \{1\}$. Damit folgt (78).

Seien $a_i \in A_i$ für $i = 1, \ldots, n$. Dann existieren $b_{ij} \in B_i$ und $r_{ij} \in [0,1]$ mit $\sum_j r_{ij} = 1$, so daß $a_i = \sum_j r_{ij} b_{ij}$. Es folgt

$$a_1 \otimes \cdots \otimes a_n = \Big(\sum_j r_{1j} b_{1j}\Big) \otimes \cdots \otimes \Big(\sum_j r_{nj} b_{nj}\Big) = \sum_{j_1,\ldots,j_n} r_{1j_1} \cdots r_{nj_n}\ b_{1j_1} \otimes \cdots \otimes b_{nj_n}$$

und

$$\sum_{j_1,\ldots,j_n} r_{1j_1} \cdots r_{nj_n} = \Big(\sum_j r_{1j}\Big) \cdots \Big(\sum_j r_{nj}\Big) = 1,$$

also ist $a_1 \otimes \cdots \otimes a_n$ eine Konvexkombination über $\left\{\bigotimes_{i=1}^n b_i : b_i \in B_i\right\}$, womit sich (79) ergibt. ■

A.6 Zu Kapitel 6

A.6.1 Bemerkung auf Seite 40

In Abschnitt 6.2, Seite 40 haben wir behauptet, daß selbst für exponentiell zufällige Familien von Quellen i. a. nicht gilt, daß $\mathcal{F}_{\mathrm{ARnd},\Sigma_\mathcal{X}}$ von $\mathcal{F}_\mathcal{X}$ sicher realisiert wird.

> **Hilfsatz A.34: Probabilistische Unentscheidbarkeit**
> Es existiert eine Funktion $h : \mathbb{N} \to \{0,1\}$, so daß keine probabilistische Turingmaschine existiert, die für fast alle $i \in \mathbb{N}$ bei Eingabe i mit einer (möglicherweise von i abhängigen) Wahrscheinlichkeit von mehr als $\frac{1}{2}$ die Ausgabe $h(i)$ hat. □

Der Beweis zu diesem Hilfsatz findet sich auf Seite 87.

Beweis (zur Bemerkung): Es sei h eine Funktion wie in Hilfsatz A.34 und $\mathcal{X}$ eine parametrische Familie von Quellen mit $I_\mathcal{X} = \mathbb{N}$, und $\mathcal{X}(k,i)$ sei gleichverteilt auf $\{0,1\}^{h(x)}$ für alle $i \in I_\mathcal{X}$ und $k \in \mathbb{N}$. Da $\mathcal{X}(k,i)$ perfekt zufällig, ist $\mathcal{X}$ exponentiell zufällig.

Wir nehmen an, $\mathcal{F}_\mathcal{X}$ würde von $\mathcal{F}_{\mathrm{ARnd},\Sigma_\mathcal{X}}$ sicher realisiert.

Man betrachte den Adversary $\mathcal{A}$ mit folgendem Verhalten: Bei einer Nachricht (i) von der Umgebung wird eine Nachricht $(\mathit{source}, 1, i)$ an die ideale Funktionalität $\mathcal{F}_\mathcal{X}$ gesandt.

Dann existiert ein Adversary $\mathcal{S}$, so daß für jede Umgebung die Ausgabe des idealen und des realen Modells ununterscheidbar sind.

Die Partei P_1 leite alle Nachrichten zwischen Funktionalität und Umgebung direkt weiter.

Es bezeichne k den Sicherheitsparameter.

Betrachte nun die Umgebung $\mathcal{Z}$, welche dem Adversary die Nachricht (k) und danach der Partei P_1 die Nachricht (random) schickt. Erhält die Umgebung die Nachricht $(\mathit{nodata}, 1)$ von P_1, so terminiert sie und gibt 0 aus, bei einer anderen Nachricht terminiert sie ebenfalls und gibt 1 aus.

Im idealen Modell ergibt sich die Ausgabe nun wie folgt: Die Funktionalität $\mathcal{F}_\mathcal{X}$ erhält vom Adversary die Nachricht $(\mathit{source}, 1, k)$, also wird sie auf die Nachricht (random) von P_1 genau dann mit $(\mathit{nodata}, 1)$ antworten, wenn $\big(\mathcal{X}(k,k)\big)_1 = \perp$, was wiederum genau für $h(k) = 0$ der Fall ist. Andernfalls $(h(k) = 1)$ wird eine Nachricht der Form $(\mathit{data}, 1, \cdot)$ an die Umgebung versandt.

Also ist die Ausgabe der Umgebung gerade $h(k)$.

Da die Ausgabe $R(k)$ im realen Modell von dieser ununterscheidbar sein soll, gilt für hinreichend großes k

$$\tfrac{1}{2} > \mathrm{SD}(h(k), R(k)) \overset{2.8}{=} \tfrac{1}{2}\big|1 - P(R(k) = h(k))\big| + \tfrac{1}{2}\big|P(R(k) \neq h(k))\big| = P\big(R(k) \neq h(k)\big). \tag{81}$$

Da alle Komponenten des realen Modells Turing-Maschinen sind, läßt sich $R(k)$ durch eine Turingmaschine simulieren, damit ist (81) ein Widerspruch zu Hilfsatz A.34. ■

Man beachte, daß die einzige Eigenschaft, die wir von $\mathcal{F}_{\mathrm{ARnd},\Sigma_\mathcal{X}}$ benutzt haben, die ist, daß sich $\mathcal{F}_{\mathrm{ARnd},\Sigma_\mathcal{X}}$ durch eine Turingmaschine simulieren läßt.

Beweis (zu Hilfsatz A.34): Es sei M eine (nicht notwendig injektive) Aufzählung aller probabilistischen Turingmaschinen.[31] Dann sei

$$h(i) := \begin{cases} 1, & P\big(M(i) \text{ hält bei Eingabe } i\big) > \frac{1}{2}, \\ 0, & \text{sonst.} \end{cases} \tag{82}$$

Wir nehmen nun an, es gebe eine probabilistische Turingmaschine H, so daß für alle $i \in \mathbb{N}$ gilt:

$$P\big(H(i) = h(i)\big) > \tfrac{1}{2}. \tag{83}$$

Nun konstruieren wir die Turingmaschine $\bar{H}$, welche bei Eingabe i zunächst $H(i)$ simuliert, bei $H(i) = 0$ hält, und bei $H(i) = 1$ in eine Endlosschleife eintritt (d. h. nicht hält). Sei $\eta \in \mathbb{N}$ mit $M(\eta) = \bar{H}$.

Wir werden nun feststellen, daß dann weder $h(\eta) = 1$ noch $h(\eta) = 0$, womit die Existenz von H zum Widerspruch geführt und der Beweis vollendet wäre.

Wir nehmen hierzu zunächst $h(\eta) = 0$ an. Dann ist mit (82)

$$P\big(M(\eta) = \bar{H} \text{ hält bei Eingabe } \eta\big) = P\big(H(\eta) = 0\big) = P\big(H(\eta) = h(\eta)\big) \overset{(83)}{>} \tfrac{1}{2},$$

und damit folgt mit (82) der Widerspruch $h(\eta) = 1$.

Nun nehmen wir $h(\eta) = 1$ an. Dann ist mit (82)

$$P\big(M(\eta) = \bar{H} \text{ hält bei Eingabe } \eta\big) = P\big(H(\eta) = 0\big) = P\big(H(\eta) = 1 - h(\eta)\big) \overset{(83)}{<} \tfrac{1}{2},$$

und es folgt mit (82) der Widerspruch $h(\eta) = 0$.

Da keine Turingmaschine H existiert, die $h(i)$ für alle $i \in \mathbb{N}$ mit einer Wahrscheinlichkeit von mehr als $\frac{1}{2}$ berechnet, existiert auch keine, die dies für fast alle $i \in \mathbb{N}$ tut, da man aus letzterer erstere konstruieren könnte. ■

A.6.2 Satz 6.7

> **Satz 6.7: Sicherheit von $\mathcal{F}_\mathcal{X}$**
> Ist $\mathcal{X}$ eine simulierbare und superpolynomiell zufällige Familie von Quellen, so wird $\mathcal{F}_{\mathrm{ARnd},\Sigma_\mathcal{X}}$ von $\mathcal{F}_\mathcal{X}$ sicher realisiert (im Canetti-Modell). □

Beweis: Zunächst wollen wir die Aussage für den Fall beweisen, daß $\mathcal{X}$ eine perfekt zufällige und exakt simulierbare Familie von Quellen ist. Es sei k der Sicherheitsparameter.

Wir nehmen o. B. d. A. an, der Real-Life-Adversary nehme einfach nur Anweisungen der Umgebung entgegen und führe diese aus, und leite empfangene Nachrichten an die Umgebung weiter (ein sogenannter Dummy-Adversary). Diese Annahme ist nach [Can00] gerechtfertigt (siehe dort die Erläuterungen zum *dummy adversary*).

Wir konstruieren dann den idealen Adversary $\mathcal{S}$ wie folgt:

- Zunächst werden die Variablen $s_j := o$ für alle j initialisiert, wobei o wie in Definition 6.5 sei.
- Ungültige Nachrichten von der Umgebung werden ignoriert (d. h. solche, die auch der Dummy-Adversary ignoriert hätte).
- Anweisungen der Umgebung, die einen der folgenden Punkte betreffen, befolgt $\mathcal{S}$:
  - Korruption von Parteien.

[31] Das heißt i ist eine Gödelnummer der probabilistischen Turingmaschine $M(i)$

 - Zurückliefern des Inhalts der Ausgangsbänder der Parteien.
 - Zurückliefern der Adressaten der Nachrichten auf dem Ausgangsband der Funktionalität. (Hierbei werden aber weiter unten verworfene Nachrichten natürlich nicht mit zurückgeliefert.)
 - Ausliefern einer Nachricht von der Funktionalität an eine Partei.

- Verlangt die Umgebung, eine Nachricht an die Funktionalität zu liefern, so liefert $\mathcal{S}$ diese nicht aus. Hat die Nachricht die Form (source, j, i), wobei P_j eine existierende Partei ist, die noch keine Nachricht der Form (random) erhalten hat,[32] und $i \in I_{\mathcal{X}}$, und noch keine Nachricht der Form (source, j, i) von der Umgebung erhalten wurde, so wird die Variable $s_j := i$ in $\mathcal{S}$ gesetzt.

- Sendet die Umgebung eine Nachricht an Partei P_j, so geht $\mathcal{S}$ wie folgt vor:

  Es sei p_j die Anzahl der von der Funktionalität bereits an P_j adressierten Nachrichten (einschließlich der aktuellen). Bestimme den Wert von $\sigma := (U_{s_j}^{(j)})_{p_j}$. Ist $\sigma = \perp$, so verwerfe die von der Funktionalität an P_j adressierte Nachricht, und fordere die Funktionalität mittels der Nachricht (stop) auf, eine Nachricht der Form $(\mathit{nodata}, \cdot)$ zu senden. Diese verwerfe dann nicht.

  Hierbei sei $U_{s_j}^{(j)}$ eine Kopie der Quelle $\mathcal{X}^{(j)}(s_j, k)$, simuliert vom Adversary unter Benutzung der nach Definition 6.6 existierenden Turingmaschine M. Da $\mathcal{X}$ exakt simulierbar ist, hat $U_{s_j}^{(j)}$ genau die gleiche Verteilung wie $\mathcal{X}^{(j)}(s_j, k)$. Es existiert dann ein Polynom p, so daß bei fortlaufender Berechnung der Komponenten von $U_{s_j}^{(j)}$ für jede Komponenten nur maximal $p(k)$ Schritte benötigt werden,[33] also kann $\mathcal{S}$ diese Simulation auch durchführen und dabei polynomiell bleiben.

Es seien (jeweils zum Zeitpunkt des Beginns der i-ten Aktivierung der Umgebung):

Z_i^R	der Zustand der Umgebung $\mathcal{Z}$ im Real-Life-Modell.
Z_i^I	der Zustand der Umgebung $\mathcal{Z}$ im idealen Modell.
P_i^R	der Verlauf (*history*) aller Parteien[34] im Real-Life-Modell.
P_i^I	der Verlauf (*history*) aller Parteien im idealen Modell.
F_i^R	das ausgehende Kommunikationsband der Funktionalität im Real-Life-Modell.
F_i^I	das ausgehende Kommunikationsband der Funktionalität im idealen Modell, ohne die von $\mathcal{S}$ verworfenen Nachrichten.
S_i^R	die Variablen $(s_j)_j$ der Funktionalität im Real-Life-Modell.
S_i^I	die Variablen $(s_j)_j$ des Adversaries $\mathcal{S}$ im idealen Modell.

Wir behaupten nun, daß

$$\mathrm{SD}(R_i; I_i) = 0 \qquad (i \in \mathbb{N}). \tag{84}$$

mit

$$R_i := (Z_i^R, P_i^R, F_i^R, S_i^R) \qquad \text{und} \qquad I_i := (Z_i^I, P_i^I, F_i^I, S_i^I).$$

Wir beweisen (84) induktiv, der Fall $i = 1$ ist klar.

Es gelte (84) nun für $i-1$, wir zeigen es dann für i. Wir unterscheiden anhand der verschiedenen Aktionen, die $\mathcal{Z}$ am Ende seiner Aktivierung durchführen kann. Führen all diese Aktionen zu den gleichen Veränderungen auf R_i und I_i, so ist die Aussage bewiesen.

1. Fall: $\mathcal{Z}$ sendet eine ungültige Nachricht an eine Partei.[35]

Diese Nachricht wird in beiden Modellen von einer Partei an die Funktionalität weitergeleitet und von dieser ignoriert. Somit hängt R_i in gleicher Weise von R_{i-1} ab wie I_i von I_{i-1}.

[32] Dies weiß $\mathcal{S}$, da es von der Funktionalität in diesem Fall mit der Auslieferung der Antwort beauftragt worden wäre.

[33] Da der Dummy-Adversary polynomiell beschränkt ist, gilt für ein von der Umgebung mitgeteiltes (und vom Dummy-Adversary nicht ignoriertes) $i \in I_{\mathcal{X}}$: $|i| \le p_1(k)$ für geeignetes Polynom p_1. Weiterhin ist aus dem gleichen Grund die Anzahl der zurückgelieferten Symbol aus der Quelle durch $p_2(k)$ für ein geeignetes Polynom p_2 beschränkt. Damit ist die Laufzeit pro Komponente von $U_i^{(j)}$ durch $p(k) := p_3(k+p_1(k)+p_2(k))$ beschränkt, wobei p_3 das in Definition 6.6 angegebene, die Laufzeit vom M beschränkende Polynom sei. Analoges gilt für die Laufzeit, die zum Entscheiden des Problems $i \in I_{\mathcal{X}}$ notwendig ist.

[34] Bestehend aus allen von diesen bislang gesandten und empfangenen Nachrichten.

[35] Das heißt eine Nachricht an P_j, die weder die Form (random) noch (init) hat, oder die die Form (init) hat und an Partei j geht, obwohl P_j bereits eine Nachricht dieser Form erhalten hat, oder die Form (random), obwohl P_j noch keine Nachricht der Form (init) erhalten hat.

2. Fall: $\mathcal{Z}$ fordert den Adversary auf, eine Partei zu P_j zu korrumpieren. In beiden Fällen führt der Adversary dies aus, und in beiden Fällen hängt die an $\mathcal{Z}$ zurückgelieferte Information nur von P^R_{i-1} bzw. P^I_{i-1} ab, somit hängt R_i in gleicher Weise von R_{i-1} ab wie I_i von I_{i-1}.

3. Fall: $\mathcal{Z}$ fordert den Adversary auf, den Inhalt der Nachrichten auf den Ausgangsbändern der Parteien und die Empfänger der Nachrichten auf dem Ausgangsband der Funktionalität zu übermitteln.

Sowohl im Real-Life- als auch im idealen Modell kommt der Adversary dieser Aufforderung nach.

Die Nachrichten auf den Ausgangsbändern der Parteien hängen – analog dem 2. Fall – nur von P^R_{i-1} bzw. P^I_{i-1} ab.

Die Empfänger der Nachrichten auf dem Ausgangsband der Funktionalität hängen nur von F^R_{i-1} bzw. P^I_{i-1} ab.

Insgesamt hängt R_i in gleicher Weise von R_{i-1} ab wie I_i von I_{i-1}.

4. Fall: $\mathcal{Z}$ fordert den Adversary auf, eine Nachricht von der Funktionalität an eine Partei auszuliefern.

In beiden Modellen kommt der Adversary dem nach. Die auszuliefernden Nachrichten sind in F^R_{i-1} bzw. F^I_{i-1} enthälten, also folgt – wie in den vorangehenden Schritten – daß R_i in gleicher Weise von R_{i-1} abhängt wie I_i von I_{i-1}.

5. Fall: $\mathcal{Z}$ schickt dem Adversary eine an die Funktionalität weiterzuleitende Nachricht.

Ist die Nachricht nicht von der Form (source, j, i) mit P_j einer existierenden Partei, die noch keine Nachricht der Form (random) von der Umgebung erhalten hat, und $i \in I_{\mathcal{X}}$, so wird sie im Real-Life-Modell von der Funktionalität, im idealen Modell vom Adversary ignoriert, und die Argumentation von Fall 1 gilt auch hier.

Ist die Nachricht der Form (source, j, i), $i \in I_{\mathcal{X}}$, und ist noch keine Nachricht der Form (random) an P_j geschickt worden, so geschieht folgendes:

Die Variable s_j in der Funktionalität (im Falle des Real-Life-Modells) bzw. im Adversary (im Falle des idealen Modells) wird auf den Wert i gesetzt. Das hat die gleichen Auswirkungen auf S^R_i bzw. S^I_i, somit hängt R_i in gleicher Weise von R_{i-1} ab wie I_i von I_{i-1}.

6. Fall: $\mathcal{Z}$ schickt eine Nachricht der Form (init) an die Partei P_j, und es wurde zuvor keine Nachricht dieser Form an P_j geschickt.

Die Funktionalität schickt dann (in beiden Modellen) eine Nachricht der Form (init, j) an den Adversary, und in beiden Modellen leitet dieser diese wiederum einfach an die Umgebung weiter.

Dies führt also zu den gleichen Veränderungen auf R_i und I_i.

7. Fall: $\mathcal{Z}$ sendet eine Nachricht der Form (random) an die Partei P_j, und es wurde bereits eine Nachricht der Form (init) an P_j gesandt.

Im Real-Life-Modell verhält sich die Funktionalität wie folgt:

- Ist $\sigma := \left(\mathcal{X}^{(j)}(s_j, k)\right)_{p_j} = \perp$, so wird (nodata, p_j) auf das Ausgangsband geschrieben.
- Sonst wird $(\mathit{data}, p_j, \sigma)$ auf das Ausgangsband geschrieben. Hierbei haben $\sigma = 1$ und $\sigma = 0$ die gleiche Wahrscheinlichkeit.

Im idealen Modell verhält sich die Funktionalität wie folgt:

- Es sei p_j die Anzahl der Nachrichten (random) die über P_j an die Funktionalität geschickt wurden (einschließlich der aktuellen). Ist $(U^{(j)}_{s_j})_{p_j} = \perp$, so wird (nodata, p_j) zurückgeliefert. (Denn die Funktionalität wird von Adversary aufgefordert, diese Nachricht zu senden.) Die von der Funktionalität gesandte, vom Adversary aber verworfene Nachricht verändert F^I_i nicht (siehe die Definition von F^I_i).
- Ansonsten wird $(\mathit{data}, p_j, \sigma)$ zurückgeliefert, wobei σ mit gleicher Wahrscheinlichkeit die Wert 0 und 1 annimmt.

Da nach Konstruktion $U_{s_j}^{(j)}$ und $\mathcal{X}^{(j)}(s_j,k)$ die gleiche Verteilung haben, sind die verschiedenen Antworten der Funktionalität in den beiden Modellen je gleichwahrscheinlich.

Dies führt wieder zu den gleichen Veränderungen auf R_i und I_i.

Es trifft also (84) zu, woraus sich direkt folgern läßt, daß auch die Ausgaben der Umgebungen in beiden Modellen den Abstand 0 haben, also $\mathcal{F}_{\mathrm{ARnd},\Sigma_\mathcal{X}}$ von $\mathcal{F}_\mathcal{X}$ sicher realisiert wird.

Wir betrachten nun den allgemeinen Fall, daß $\mathcal{X}$ nur superpolynomiell zufällig und simulierbar ist. Es sei dann $\mathcal{Y}$ eine parametrische Familie von Quellen mit

$$\mathcal{Y}(k,i) := Y^{(k,i)},$$

wobei $Y^{(k,i)}$ wie in Definition 6.6 sei. Dann ist $\mathcal{Y}$ nach Konstruktion exakt simulierbar, und es existieren ein Polynom p und eine superpolynomielle Funktion f_1, so daß für hinreichend großes $k \in \mathbb{N}$ folgt:

$$\mathrm{SD}\Big(\underbrace{\big(\mathcal{Y}(k,i)\big)_1 \dots \big(\mathcal{Y}(k,i)\big)_l}_{=:Y^{k,i,l}};\ \underbrace{\big(\mathcal{X}(k,i)\big)_1 \dots \big(\mathcal{X}(k,i)\big)_l}_{=:X^{k,i,l}}\Big) \leq \frac{p(l)}{f_1(k)} \qquad (l \in \mathbb{N}, i \in I_\mathcal{X}). \tag{85}$$

Da $\mathcal{X}$ superpolynomiell zufällig ist, existieren weiterhin eine superpolynomielle Funktion f_2 und eine perfekt zufällige parametrische Familie von Quellen $\mathcal{U}$, so daß für jedes hinreichend große $k \in \mathbb{N}$ gilt:

$$\mathrm{SD}\Big(\underbrace{\big(\mathcal{U}(k,i)\big)_1 \dots \big(\mathcal{U}(k,i)\big)_l}_{=:U^{k,i,l}};\ X^{k,i,l}\Big) \leq \frac{1}{f_2(k)} \qquad (l \in \mathbb{N}, i \in I_\mathcal{X}). \tag{86}$$

Es sei weiterhin $\mathcal{V}$ die perfekt zufällige parametrische Familie von Quellen über $\Sigma_\mathcal{X}$, bei der $|\mathcal{V}(k,i)|$ die gleiche Verteilung wie $|\mathcal{Y}(k,i)|$ hat. $\mathcal{V}$ ist exakt simulierbar, denn das Programm M, welches nach Definition 6.6 $\mathcal{Y}$ exakt simuliert, kann wie folgt in eines zur Simulation von $\mathcal{V}$ umgewandelt werden:

- Bei Aufruf mit Eingabe (k,i,n,d) berechne $(y,d') := M(k,i,n,d)$.
- Wähle zufällig $\sigma \in \Sigma_\mathcal{X}$.
- Liefere (σ, d') zurück.

Es folgt direkt aus der Konstruktion von $\mathcal{V}$, daß

$$\mathrm{SD}\big(|U^{k,i,l}|;\ |V^{k,i,l}|\big) = \mathrm{SD}\big(|U^{k,i,l}|;\ |Y^{k,i,l}|\big) \qquad (k,l \in \mathbb{N}, i \in I_\mathcal{X})$$

mit

$$V^{k,i,l} := \big(\mathcal{V}(k,i)\big)_1 \dots \big(\mathcal{V}(k,i)\big)_l,$$

woraus wegen der perfekten Zufälligkeit von $\mathcal{U}$ und $\mathcal{V}$ folgt:

$$\mathrm{SD}\big(U^{k,i,l}; V^{k,i,l}\big) \overset{2.9}{=} \mathrm{SD}\big(|U^{k,i,l}|; |V^{k,i,l}|\big) = \mathrm{SD}\big(|U^{k,i,l}|; |Y^{k,i,l}|\big) \overset{2.9(1)}{\leq} \mathrm{SD}\big(U^{k,i,l}; Y^{k,i,l}\big). \tag{87}$$

Insgesamt ergibt sich also für eine geeignete superpolynomielle Funktion f_3, geeignetes Polynom p_1 und hinreichend großes $k \in \mathbb{N}$:

$$\begin{aligned}
\mathrm{SD}(X^{k,i,l};\ V^{k,i,l}) &\overset{2.9(3)}{\leq} \mathrm{SD}(X^{k,i,l};\ U^{k,i,l}) + \mathrm{SD}(U^{k,i,l};\ V^{k,i,l}) \\
&\overset{(87)}{\leq} \mathrm{SD}(X^{k,i,l};\ U^{k,i,l}) + \mathrm{SD}(U^{k,i,l};\ Y^{k,i,l}) \\
&\overset{2.9(3)}{\leq} \mathrm{SD}(X^{k,i,l};\ U^{k,i,l}) + \mathrm{SD}(U^{k,i,l};\ X^{k,i,l}) + \mathrm{SD}(X^{k,i,l}, Y^{k,i,l}) \\
&\overset{(86),(85)}{\leq} \frac{2}{f_2(k)} + \frac{p(l)}{f_1(k)} \ \leq \frac{p_1(l)}{f_3(k)}.
\end{aligned} \tag{88}$$

Wir wollen daraus nun folgern, daß $\mathcal{F}_\mathcal{X}$ die Funktionalität $\mathcal{F}_\mathcal{V}$ sicher implementiert. Da wir bereits wissen, daß $\mathcal{F}_\mathcal{V}$ die Funktionalität $\mathcal{F}_{\mathrm{ARnd},\Sigma_\mathcal{X}}$ sicher implementiert (denn $\mathcal{V}$ ist perfekt zufällig und exakt simulierbar), folgt damit dann die Behauptung.[36]

Zu jedem Real-Life-Adversary $\mathcal{A}$ sei der ideale Adversary $\mathcal{S} := \mathcal{A}$. Es liege eine Umgebung $\mathcal{Z}$ fest. Dann sei p' ein Polynom, so daß $p'(k)$ eine obere Schranke für die Anzahl der Aktivierungen der Umgebung und des Adversaries darstellt.

Es können dann höchstens $p'(k)$ verschiedene Parteien aktiviert werden, und für jede Partei kann höchstens eine Quelle aus $\mathcal{X}$ bzw. $\mathcal{V}$ angesprochen werden (nach Konstruktion von $\mathcal{F}_\mathcal{X}$ bzw. $\mathcal{F}_\mathcal{V}$, siehe Definition 6.5). Es sei weiterhin R die Gesamtheit aller im realen bzw. idealen Modell benutzten Zufallsbänder (außer innerhalb der Funktionalität).

Wir definieren noch einige weitere Zufallsvariablen:

- Man betrachte die im Real-Life-Modell vom Adversary an die Funktionalität gesandten Nachrichten der Form $(\mathit{source}, \cdot, i)$, $i \in I_\mathcal{X}$. Dann definieren wir den Wert der Zufallsvariablen $S_R^{(j)}$ als das dritte Feld (also i) der j-ten solchen Nachricht. Analog definieren wir für das ideale Modell die Zufallsvariablen $S_I^{(j)}$.

- Es seien
$$X^{(j)} := X^{k, S_R^{(j)}, p'(k)} \qquad \text{und} \qquad V^{(j)} := V^{k, S_I^{(j)}, p'(k)}.$$
Hierbei seien für die Definition der verschiedenen $X^{(j)}$ und $V^{(j)}$ verschiedene Kopien der Zufallsvariablen $X^{k,i,l}$ und $V^{k,i,l}$ angenommen.

  Ist die j-te Nachricht $(\mathit{source}, \cdot, \cdot)$ von gültiger Form (d. h. wird sie von der Funktionalität berücksichtigt), so sind $X^{(j)}$ bzw. $V^{(j)}$ die dabei den Parteien verfügbar werdenden Quellen.

- Und zuletzt fassen wir zusammen:
$$\mathrm{Real}^{(j)} := (R, X^{(1)}, \dots, X^{(j)}) \qquad \text{und} \qquad \mathrm{Ideal}^{(j)} := (R, V^{(1)}, \dots, V^{(j)})$$
Es beschreiben nun $\mathrm{Real}^{(j)}$ und $\mathrm{Ideal}^{(j)}$ bis zum Auftreten der $(j+1)$-ten Nachricht $(\mathit{source}, \cdot, \cdot)$ sämtlichen den Parteien, der Funktionalität und der Umgebung zur Verfügung stehenden Zufall, also hängen insbesondere alle deren Aktionen bis zu dieser $(j+1)$-ten Nachricht deterministisch von $\mathrm{Real}^{(j)}$ bzw. $\mathrm{Ideal}^{(j)}$ ab. Insbesondere ist auch $S_R^{(j+1)} = s_{j+1}(\mathrm{Real}^{(j)})$ und $S_I^{(j+1)} = s_{j+1}(\mathrm{Ideal}^{(j)})$ für geeignete Funktionen s_{j+1}.

Wir behaupten nun, daß für $j \in \mathbb{N}_0$
$$\mathrm{SD}(\mathrm{Real}^{(j)}; \mathrm{Ideal}^{(j)}) \le j\,\frac{p_1 \circ p'(k)}{f_3(k)}. \tag{89}$$
Der Fall $j = 0$ ist trivial, denn es ist $\mathrm{Real}^{(0)} = R = \mathrm{Ideal}^{(0)}$. Wir wollen die Aussage nun induktiv beweisen und nehmen die Aussage für ein j als gegeben an.

Zunächst stellen wir fest, daß für beliebiges a aus dem gemeinsamen Wertebereich A von $\mathrm{Real}^{(j)}$ bzw. $\mathrm{Ideal}^{(j)}$ gilt
$$\mathrm{SD}(X^{(j+1)} \,|\, \mathrm{Real}^{(j)} = a;\; V^{(j+1)} \,|\, \mathrm{Ideal}^{(j)} = a) \;=\; \mathrm{SD}(X^{k, s_{j+1}(a), p'(k)};\; V^{k, s_{j+1}(a), p'(k)}) \overset{(88)}{\le} \frac{p_1 \circ p'(k)}{f_3(k)}. \tag{90}$$

[36] Die Transitivität des sicheren Implementierens ergibt sich direkt aus der Komponierbarkeit.

Wir setzten $M := \{m \in \Sigma_{\mathcal{X}}^* : |m| \leq p'(k)\}$ und rechnen

$$\begin{aligned}
&\mathrm{SD}(\mathrm{Real}^{(j+1)};\ \mathrm{Ideal}^{(j+1)}) \\
&\quad = \mathrm{SD}(\mathrm{Real}^{(j)}, X^{(j+1)};\ \mathrm{Ideal}^{(j)}, V^{(j+1)}) \\
&\quad \overset{2.7}{=} \tfrac{1}{2} \sum_{\substack{x \in M \\ a \in A}} \Big| P\big(\mathrm{Real}^{(j)} = a,\ X^{(j+1)} = x\big) - P\big(\mathrm{Ideal}^{(j)} = a,\ V^{(j+1)} = x\big) \Big| \\
&\quad = \tfrac{1}{2} \sum_{\substack{x \in M \\ a \in A}} \Big| P\big(X^{(j+1)} = x \mid \mathrm{Real}^{(j)} = a\big)\ P(\mathrm{Real}^{(j)} = a) \\
&\qquad\qquad - P\big(V^{(j+1)} = x \mid \mathrm{Ideal}^{(j)} = a\big)\ P(\mathrm{Ideal}^{(j)} = a) \Big| \\
&\quad = \tfrac{1}{2} \sum_{\substack{x \in M \\ a \in A}} \Big| P\big(X^{(j+1)} = x \mid \mathrm{Real}^{(j)} = a\big)\ P(\mathrm{Real}^{(j)} = a) \\
&\qquad\qquad - P\big(V^{(j+1)} = x \mid \mathrm{Ideal}^{(j)} = a\big)\ P(\mathrm{Real}^{(j)} = a) \\
&\qquad\qquad + P\big(V^{(j+1)} = x \mid \mathrm{Ideal}^{(j)} = a\big)\ \big(P(\mathrm{Real}^{(j)} = a) - P(\mathrm{Ideal}^{(j)} = a)\big) \Big| \\
&\quad \leq \sum_{a \in A} P(\mathrm{Real}^{(j)} = a)\ \tfrac{1}{2} \sum_{x \in M} \Big| P\big(X^{(j+1)} = x \mid \mathrm{Real}^{(j)} = a\big) - P\big(V^{(j+1)} = x \mid \mathrm{Ideal}^{(j)} = a\big) \Big| \\
&\qquad\qquad + \tfrac{1}{2} \sum_{a \in A} \underbrace{\sum_{x \in M} P(V^{(j+1)} = x \mid \mathrm{Ideal}^{(j)} = a)}_{=1}\ \big| P(\mathrm{Real}^{(j)} = a) - P(\mathrm{Ideal}^{(j)} = a) \big| \\
&\quad \overset{2.7}{=} \sum_{a \in A} P(\mathrm{Real}^{(j)} = a)\ \mathrm{SD}(X^{(j+1)} \mid \mathrm{Real}^{(j)} = a;\ V^{(j+1)} \mid \mathrm{Ideal}^{(j)} = a) + \mathrm{SD}(\mathrm{Real}^{(j)};\ \mathrm{Ideal}^{(j)}) \\
&\quad \overset{(90,89)}{\leq} (j+1)\, \frac{p_1 \circ p'(k)}{f_3(k)}.
\end{aligned}$$

Damit ist (89) bewiesen.

Da höchstens $p'(k)$ Nachrichten der Form $(\mathit{source}, \cdot, \cdot)$ vom Adversary an die Funktionalität geschickt werden, stellen $\mathrm{Real}^{(p'(k))}$ bzw. $\mathrm{Ideal}^{(p'(k))}$ sämtlichen das Verhalten der Umgebung bestimmenden Zufall dar, also ist die Ausgabe der Umgebung eine Funktion in einer dieser Zufallsvariablen (je nach dem, ob das Real-Life- oder das ideale Modell vorliegt). Somit ist der statistische Abstand zwischen den Ausgaben nach oben beschränkt durch

$$\mathrm{SD}\big(\mathrm{Real}^{(p'(k))};\ \mathrm{Ideal}^{(p'(k))}\big) \overset{(89)}{\leq} \frac{p'(k) \cdot p_1 \circ p'(k)}{f_3(k)}.$$

Da der Zähler ein Polynom, der Nenner aber superpolynomiell ist, ist der statistische Abstand durch das Inverse einer superpolynomiellen Funktion nach oben beschränkt, damit realisiert $\mathcal{F}_{\mathcal{X}}$ die Funktionalität $\mathcal{F}_{\mathcal{V}}$ sicher, es folgt der zu beweisende Satz. ∎

A.7 Zu Kapitel 7

A.7.1 Heuristik 7.2

> **Definition 7.1: Gewichtungstest**
> Es sei Σ nichtleer und endlich, $F > 0$, $\pi, \varrho \in \Sigma^*$, $|\varrho| > 0$, $\varepsilon \in \mathbb{R}_{>0}$, $N \in \mathbb{N}$, $M \in \mathbb{N}_0$, $L \in \mathbb{N}$, $L \geq |\pi\varrho|$. Dann sei $b_i(x)$ für $x \in \Sigma^N$ der i-te Block der Länge L in x, also
>
> $$b_i(x) := x_{(i-1)L+1} \cdots x_{iL}$$
>
> und $n_\omega(x)$ für $\omega \in \Sigma^L$ die Anzahl der $b_i(x)$ mit $b_i(x) = \omega$, also
>
> $$n_\omega(x) := \sum_{i=1}^{\lfloor N/L \rfloor} \delta(b_i(x) = \omega).$$

Weiter seien für $\varphi \in \Sigma^{L^*}$, $L^* := L - |\pi\varrho|$

$$\hat{n}_\varphi(x) := \sum_{\tilde{\varrho} \in \Sigma^{|\varrho|}} n_{\varphi\pi\tilde{\varrho}}(x),$$
$$\hat{n}(x) := \sum_{\tilde{\varphi} \in \Sigma^{L^*}} n_{\tilde{\varphi}}(x),$$
$$f_\varphi(x) := \frac{n_{\varphi\pi\varrho}(x) - 2^{-\varepsilon}\hat{n}_\varphi(x)}{\sqrt{(2^{-\varepsilon} - 2^{-2\varepsilon})\hat{n}_\varphi(x)}}$$

mit $\frac{0}{0} := 0$.

Dann ist die Testfunktion definiert durch

$$f_T(x) := \sum_{\varphi \in \Sigma^{L^*}} \max\{0, f_\varphi(x)\}^2$$

und der kritische Bereich $\mathcal{K}$ des *Gewichtungstests für $\eta(\ldots\pi;\varrho) \geq \varepsilon$ mit Schranke F, Stichprobengröße M und Blocklänge L* durch

$$\mathcal{K} = \{x \in \Sigma^L : f_T(x) \geq F \text{ oder } \hat{n}(x) < M\}.$$

Weiterhin ist der kritische Bereich des *Gewichtungstests für $\eta(\ldots\pi;\varrho) = \infty$ mit Stichprobengröße M und Blocklänge L*

$$\{x \in \Sigma^L : \exists\varphi \in \Sigma^{L^*}\colon n_{\varphi\pi\varrho}(x) \neq 0 \text{ oder } \hat{n}(x) < M\}.$$ □

Heuristik 7.2: Niveau des Gewichtungstests

Sei X eine Quelle über Σ, $\alpha \in [0,1]$, $\pi, \varrho \in \Sigma^*$, $|\varrho| > 0$, $\varepsilon \in \mathbb{R}_{>0} \cup \{\infty\}$, $N \in \mathbb{N}$, $M \in \mathbb{N}_0$, $L \in \mathbb{N}$, $L \geq |\pi\varrho|$, $L^* := L - |\pi\varrho|$.

Es sei $F \in \mathbb{R}_{>0}$ mit

$$2^{-\#\Sigma^{L^*}} \sum_{i=1}^{\#\Sigma^{L^*}} \binom{\#\Sigma^{L^*}}{i} (1 - \chi_i^2(F)) \leq \alpha, \tag{13}$$

wobei χ_i^2 die Verteilungsfunktion der Chi-Quadrat-Verteilung mit i Freiheitsgraden sei.

Es bezeichne $\mathcal{K}$ den kritischen Bereich des Gewichtungstests für $\eta(\ldots\pi;\varrho) \leq \varepsilon$ mit Schranke F, Stichprobengröße M und Blocklänge L, und $\hat{n}$ sei wie in Definition 7.1.

Ist $\eta^{\{X\}}(\xi\pi;\varrho) \geq \varepsilon$ für alle $\xi \in \Sigma^*$, so gilt für große M approximativ

$$P(X \in \mathcal{K} \text{ und } \hat{n}(X) \geq M) \leq \alpha.$$ □

Begründung: Heuristische Überlegungen sind im folgenden kursiv gesetzt. Wir betrachten zunächst den Fall $\varepsilon < \infty$. Abkürzend schreiben wir $\eta := \eta^{\{X\}}$ und $N' := \lfloor L/N \rfloor$. Es gelte $\eta(\xi\pi;\varrho)$ für alle $\xi \in \Sigma^*$.

Es sei für $\varphi \in \Sigma^{L^*}$

$$H_i^\varphi := \delta(b_i(X) = \varphi\pi\varrho),$$
$$\hat{H}_i^\varphi := \delta(\exists\tilde{\varrho} \in \Sigma^{|\varrho|} : b_i(X) = \varphi\pi\tilde{\varrho}),$$
$$\hat{N}^\varphi := \sum_{i=1}^{N'} \hat{H}_i^\varphi.$$

Es ist dann wegen $\eta(\xi\pi;\varrho) \leq \varepsilon$ für $\xi \in \Sigma^*$

$$P(H_i^\varphi = 1 \mid \hat{H}_i^\varphi = 1,\ H_1^\varphi \ldots H_{i-1}^\varphi = h) \leq 2^{-\varepsilon} \qquad (h \in \{0,1\}^{(i-1)}),$$

und $H_i^\varphi = 0$, falls $\hat{H}_i^\varphi = 0$.

Es seien dann $B_i^\varphi := \hat{H}_i^\varphi \bar{B}_i^\varphi$ mit unabhängigen $\mathrm{Bin}(2^{-\varepsilon})$-verteilten Zufallvariablen $\bar{B}_i^\varphi$. Es ist dann

$$P\big(B_i^\varphi = 1 \mid \hat{H}_i^\varphi = 1\big) = 2^{-\varepsilon},$$

und $B_i^\varphi = 0$, falls $\hat{H}_i^\varphi = 0$.

Es folgt dann für $t \in \mathbb{R}$

$$P(f_\varphi(X) \geq t) = P\left(\frac{\sum_{i=1}^{N'} H_i^\varphi - 2^{-\varepsilon}\hat{N}^\varphi}{\sqrt{(2^{-\varepsilon} - 2^{-2\varepsilon})\hat{N}^\varphi}} \geq t\right) \leq P\left(\underbrace{\frac{\sum_{i=1}^{N'} B_i^\varphi - 2^{-\varepsilon}\hat{N}^\varphi}{\sqrt{(2^{-\varepsilon} - 2^{-2\varepsilon})\hat{N}^\varphi}}}_{=:B^\varphi} \geq t\right).$$

Da B_i^φ für $\hat{H}_i^\varphi = 1$ $\mathrm{Bin}(2^{-\varepsilon})$-Verteilung hat und für $\hat{H}_i^\varphi = 0$ verschwindet, können wir $\sum_i B_i^\varphi$ als Summe von $\hat{N}^\varphi$ $\mathrm{Bin}(2^{-\varepsilon})$-verteilten Zufallsvariablen betrachten, und nach dem Zentralen Grenzwertsatz für großes $\hat{N}^\varphi$ die Zufallsvariablen B^φ als approximativ standardnormalverteilt und unabhängig annehmen.

Dann ist also für unabhängige, standardnormalverteilte Zufallsvariablen N_φ approximativ

$$P(f_\varphi(X) \geq t) \leq P(B^\varphi \geq t) = P(N_\varphi \geq t),$$

also für $F > 0$

$$P\big(\max\{0, f_\varphi(X)\}^2 \geq F\big) \leq P\big(\max\{0, B^\varphi\}^2 \geq F\big).$$

Wir nehmen weiterhin an, daß die $f_\varphi(X)$ voneinander weitgehend unabhängig sind, und erhalten

$$P\Big(\underbrace{\sum_{\varphi \in \Sigma^{L^*}} \max\{0, f_\varphi(X)\}^2}_{=f_T(X)} \geq F\Big) \leq P\Big(\sum_{\varphi \in \Sigma^{L^*}} \max\{0, B^\varphi\}^2 \geq F\Big). \tag{91}$$

Sind U^φ voneinander und von den B^φ unabhängige $\mathrm{Bin}(\frac{1}{2})$-verteilte Zufallsvariablen, so haben $U^\varphi(B^\varphi)^2$ und $\max\{0, B^\varphi\}^2$ die gleiche Verteilung (denn die Dichte von B^φ ist symmetrisch um 0), es folgt

$$\begin{aligned}
P(f_T(X) \geq F) &\overset{(91)}{\leq} P\Big(\sum_{\varphi \in \Sigma^{L^*}} U^\varphi (B^\varphi)^2 \geq F\Big) \\
&= \sum_{u \in \{0,1\}^{\Sigma^{L^*}}} P\Big(\sum_{\varphi \in \Sigma^{L^*}} U^\varphi (B^\varphi)^2 \geq F,\ (U_\varphi)_\varphi = u\Big) \\
&= \sum_{u \in \{0,1\}^{\Sigma^{L^*}}} P\Big(\sum_{\substack{\varphi \in \Sigma^{L^*} \\ u_\varphi = 1}} (B^\varphi)^2 \geq F,\ (U_\varphi)_\varphi = u\Big)\ P\big((U_\varphi)_\varphi = u\big) \\
&= \sum_{\substack{u \in \{0,1\}^{\Sigma^{L^*}} \\ \omega_1(u) \neq 0}} (1 - \chi^2_{\omega_1(u)}(F))\ 2^{-\#\Sigma^{L^*}} \\
&= 2^{-\#\Sigma^{L^*}} \sum_{i=1}^{\#\Sigma^{L^*}} \binom{\#\Sigma^{L^*}}{i} (1 - \chi_i^2(F)) \\
&\leq \alpha.
\end{aligned} \tag{92}$$

Es folgt schließlich

$$\begin{aligned}
P\big(X \in \mathcal{K}\ \wedge\ \hat{n}(X) \geq M\big) &= P\big((f_T(X) \geq F\ \vee\ \hat{n}(X) < M)\ \wedge\ \hat{n}(X) \geq M\big) \\
&\leq P\big(f_T(X) \geq F\big) \overset{(92)}{\leq} \alpha.
\end{aligned}$$

Es bleibt der Fall $\varepsilon = \infty$ zu untersuchen. Es ist dann $P(H_i^\varphi = 0) = 1$ für alle $i = 1, \dots, N'$, $\varphi \in \Sigma^{L^*}$, also

$$\begin{aligned}
P(X \in \mathcal{K} \ \wedge \ \hat{n}(X) \geq M) &= P\big((\exists \varphi \in \Sigma^{L^*}\colon n_{\varphi\pi\varrho}(X) \neq 0 \ \vee \ \hat{n}(X) < M) \ \wedge \ \hat{n}(X) \geq M\big) \\
&= P\Bigg(\bigg(\exists \varphi \in \Sigma^{L^*}\colon \sum\nolimits_{i=1}^{N'} H_i^\varphi \neq 0 \ \vee \ \hat{n}(X) < M\bigg) \ \wedge \ \hat{n}(X) \geq M\Bigg) \\
&= P\big(\hat{n}(X) < M \ \wedge \ \hat{n}(X) \geq M\big) \\
&= 0 \leq \alpha.
\end{aligned}$$

■

Anhang B

Konfiguration von `randomextract`

Als Eingabe nimmt `randomextract` eine ASCII-Textdatei, deren Format im folgenden definiert wird. Alternativ zu einer Lektüre dieser Spezifikation ist es auch möglich, `RandomExtraction` zu verwenden, um Beispieldateien zu erhalten.

In der folgenden Spezifikation bezeichnet ⟨*integer*⟩ eine ganze Zahl, ⟨*real*⟩ eine Gleitkommazahl, und ⟨*word*⟩ eine Zeichenkette aus Buchstaben und Ziffern. Der Inhalt der Datei ist das Nichtterminal ⟨*file*⟩:

⟨*file*⟩ ::= ⟨*action*⟩ *

⟨*action*⟩ ::= ⟨*test*⟩ | ⟨*showsource*⟩ | ⟨*showweight*⟩

B.1 Quellen

Um eine Stichprobe einer Quelle anzeigen zu lassen, verwenden wir folgende Syntax:

⟨*showsource*⟩ ::= '`showsource`' '`{`' (⟨*showsource-source*⟩ | ⟨*showsource-len*⟩ | '`verbose`' | '`totallen`') * '`}`'

⟨*showsource-source*⟩ ::= '`source`' ⟨*source*⟩

⟨*showsource-len*⟩ ::= '`len`' ⟨*integer*⟩

Hierbei gibt ⟨*showsource-source*⟩ die zu verwendende Quelle an (zur Syntax von ⟨*source*⟩ siehe unten) und ⟨*showsource-len*⟩ die Anzahl auszugebener Symbole. Ist zusätzlich '`verbose`' angegeben, so werden zusätzliche Informationen mit ausgegeben, und ist '`totallen`' spezifiziert, so wird am Ende ausgegeben, wieviele Symbol die Quelle insgesamt produziert hat (selbst wenn dies ⟨*showsource-len*⟩ übersteigt).[37]

Eine Quelle wird in der folgenden Form spezifiziert:

⟨*source*⟩ ::= ⟨*lsbfile-source*⟩ | ⟨*stupiddeterministic-source*⟩ | ⟨*linuxkernel-source*⟩ | ⟨*fixedlen-source*⟩ | ⟨*crng-source*⟩ | ⟨*skipprefix-source*⟩ | ⟨*hash-source*⟩ | ⟨*drop-source*⟩ | ⟨*adversarialchmm-source*⟩ | ⟨*explicit-source*⟩ | ⟨*biased-source*⟩ | ⟨*adaptive-source*⟩ | ⟨*autocorrelation-source*⟩

Die verschiedenen Quellentypen werden jeweils durch eines dieser Nichtterminale ausgewählt.

Um binäre Zufallsdaten aus einer Datei zu lesen, kann die folgende Quelle benutzt werden:

⟨*lsbfile-source*⟩ ::= '`lsbfile`' ⟨*word*⟩

Hierbei gibt der Parameter ⟨*word*⟩ den Namen einer lesbaren Datei an. Diese wird byteweise gelesen, und in jedem Byte wird beim niederwertigsten Bit begonnen.

Zu Testzwecken mag die folgende binäre Quelle dienen:

⟨*stupiddeterministic-source*⟩ ::= '`stupiddeterministic`' ⟨*stupiddeterministic-type*⟩

⟨*stupiddeterministic-type*⟩ ::= '`concat`' | '`fixedlen`'

Diese Quelle ist deterministisch, d. h. die Stichprobe ist immer die gleiche. Hat ⟨*stupiddeterministic-type*⟩ den Wert '`concat`', so werden die Zahlen $1, 2, \ldots$ binär dargestellt (mit dem niederwertigsten Bit zuerst) und dann konkateniert. Bei '`fixedlen`' wird ebenso verfahren, jedoch werden die Zahlen vor der Konkatenation mit vorlaufenden Nullen dargestellt, so daß alle eine Länge von 32 Bit haben.

Unter Linux kann die folgende Quelle genutzt werden:

⟨*linuxkernel-source*⟩ ::= '`linuxkernel`' ⟨*linuxkernel-type*⟩

⟨*linuxkernel-type*⟩ ::= '`pseudo`' | '`real`'

Hier wird der Zufallszahlengenerator des Linux-Kernels verwendet. Ist '`pseudo`' angegeben, so wird der Pseudozufallszahlengenerator verwendet (`/dev/urandom`), sonst der Zufallszahlengenerator (`/dev/random`).

Um eine Quelle nach einer bestimmten Länge terminieren zu lassen, verwende man die folgende Syntax:

[37] Hört die Quelle nicht auf, Daten zu liefern, so terminiert `randomextract` mit der Option '`totallen`' nicht.

⟨*fixedlen-source*⟩ ::= ‘`fixedlen`’ ‘`{`’ (⟨*fixedlen-subsource*⟩ | ⟨*fixedlen-len*⟩) `*` ‘`}`’
⟨*fixedlen-subsource*⟩ ::= ‘`source`’ ⟨*source*⟩
⟨*fixedlen-len*⟩ ::= ‘`length`’ ⟨*integer*⟩

Diese Quelle entnimmt die Stichprobe der durch ⟨*fixedlen-subsource*⟩ angegebenen Quelle, und wenn diese Stichprobe länger als ⟨*fixedlen-len*⟩ Symbole ist, wird sie auf diese Länge gekürzt (es wird also ein Präfix der entsprechenden Länge ausgegeben).

Zufällige Quellen beliebiger Alphabetsgröße simulieren wir mittels:

⟨*crng-source*⟩ ::= ‘`crng`’ ⟨*integer*⟩

Es bestimmt ⟨*integer*⟩ die Alphabetgröße n, zum Erzeugen der Folge wird die C-Funktion `rand()%n` verwandt.

Um von einer Quelle einen Prefix fester Länge zu entfernen, dient

⟨*skipprefix-source*⟩ ::= ‘`skipprefix`’ ‘`{`’ (⟨*skipprefix-subsource*⟩ | ⟨*skipprefix-prefixlen*⟩) `*` ‘`}`’
⟨*skipprefix-subsource*⟩ ::= ‘`source`’ ⟨*source*⟩
⟨*skipprefix-prefixlen*⟩ ::= ‘`prefix`’ ⟨*integer*⟩

Diese Quelle nimmt eine Stichprobe von ⟨*skipprefix-subsource*⟩ und entfernt die ersten ⟨*skipprefix-prefixlen*⟩ Symbole.

Soll zu Testzwecken eine bestimmte Symbolfolge als Quelle verwendet werden, so kann man folgende Syntax nutzen:

⟨*explicit-source*⟩ ::= ‘`explicit`’ ‘`{`’(⟨*explicit-alphabet*⟩ | ⟨*explicit-data*⟩ | ⟨*explicit-repeatfrom*⟩) `*` ‘`}`’
⟨*explicit-alphabet*⟩ ::= ‘`alphabet`’ ‘`{`’ ⟨*word*⟩ `+` ‘`}`’
⟨*explicit-data*⟩ ::= ‘`data`’ ‘`{`’ ⟨*word*⟩ `*` ‘`}`’
⟨*explicit-repeatfrom*⟩ ::= ‘`repeatfrom`’ ⟨*integer*⟩

Die in ⟨*explicit-alphabet*⟩ angegebene Folge von ⟨*word*⟩ spezifiziert das Alphabet der Quelle, die auszugebenen Symbole werden durch ⟨*explicit-data*⟩ bestimmt.

Wird ⟨*explicit-repeatfrom*⟩ angegeben, so wiederholt sich die Folge immer wieder, wobei ab dem zweiten Durchlauf die in ⟨*explicit-repeatfrom*⟩ angegebene Anzahl von Symbolen weggelassen wird. Wurde hingegen ⟨*explicit-repeatfrom*⟩ nicht angegeben, so terminiert die Quelle nach einmaliger Ausgabe der Folge.

Wird ⟨*explicit-alphabet*⟩ nicht angegeben, so wird das Alphabet automatisch aus ⟨*explicit-data*⟩ erzeugt. Die interne Numerierung des Alphabets (wichtig z. B. für ⟨*autocorrelation-source*⟩ u. a.) richtet sich dann nach der Reihenfolge des Vorkommens in ⟨*explicit-data*⟩.

Eine unabhängig identisch verteilte binäre Quelle beschreibt

⟨*biased-source*⟩ ::= ‘`biased`’ ‘`{`’ ⟨*biased-bias*⟩ ‘`}`’
⟨*biased-bias*⟩ ::= ‘`bias`’ ⟨*real*⟩

Die in ⟨*biased-bias*⟩ angegebene Wahrscheinlichkeit ist die für das Ausgeben einer 1.

Um Korrelationen in den Daten zu untersuchen, bietet sich folgendes an:

⟨*autocorrelation-source*⟩ ::= ‘`autocorrelation`’ ‘`{`’ (⟨*autocorrelation-subsource*⟩ | ⟨*autocorrelation-delay*⟩) `*` ‘`}`’
⟨*autocorrelation-subsource*⟩ ::= ‘`source`’ ⟨*source*⟩
⟨*autocorrelation-delay*⟩ ::= ‘`delay`’ ⟨*integer*⟩

Sei X die durch ⟨*autocorrelation-subsource*⟩ spezifizierte Quelle. Dann gibt ⟨*autocorrelation-source*⟩ die Folge $(X_i + X_{i+\tau} \bmod n)$ aus, wobei n die Kardinalität des Alphabets und τ durch ⟨*autocorrelation-delay*⟩ bestimmt sei. Für die Addition wird hier (wie auch bei allen weiter unten beschriebenen Quellen, die mit Symbolen rechnen) die interne Numerierung des Alphabets zugrundegelegt, die evtl. von der durch die Symbolnamen implizierten abweichen kann.

Zur Simulation von CHMM-Quellen dient

⟨*adversarialchmm-source*⟩ ::= ‘`adversarialchmm`’ ‘`{`’ (⟨*adversarialchmm-chmm*⟩ | ⟨*adversarialchmm-strategylen*⟩) `*` ‘`}`’

⟨*adversarialchmm-chmm*⟩ ::= ‘`chmm`’ ⟨*chmm*⟩

⟨*adversarialchmm-strategylen*⟩ ::= ‘`strategylen`’ ⟨*integer*⟩

Diese Quelle generiert Symbolfolgen, deren Verteilung der dem in ⟨*adversarialchmm-chmm*⟩ spezifizierten CHMM genügen (siehe Definition 5.3).

Dabei legt sich die Quelle auf eine Strategie zum Generieren der Symbole fest (z. B. die Ausgabe möglichst wenig zu verändern, möglichst viel zu verändern, ein festes Symbol möglichst oft auszugeben), die sie dann unter Beachtung der vom CHMM vorgeschriebenen Transitionsbereiche zu verfolgen versucht.

Die gewählte Strategie wird nach einer festen Anzahl von Symbolen neu gewählt, diese Anzahl ist durch ⟨*adversarialchmm-strategylen*⟩ zu bestimmen.

Zur Syntax von ⟨*chmm*⟩ siehe Abschnitt B.4.

Das blockweise Anwenden von Hashfunktionen ist wie folgt realisierbar:

⟨*hash-source*⟩ ::= ‘`hash`’ ‘`{`’ (⟨*hash-subsource*⟩ | ⟨*hash-sourceblock*⟩ | ⟨*hash-targetblock*⟩ | ⟨*hash-hash*⟩) `*` ‘`}`’

⟨*hash-subsource*⟩ ::= ‘`source`’ ⟨*hash*⟩

⟨*hash-sourceblock*⟩ ::= ‘`sourceblock`’ ⟨*integer*⟩

⟨*hash-targetblock*⟩ ::= ‘`targetblock`’ ⟨*integer*⟩

⟨*hash-hash*⟩ ::= ‘`hash`’ ⟨*hash*⟩

Die Ausgabe der Quelle ⟨*hash-subsource*⟩ wird in Blöcke der Länge ⟨*hash-sourceblock*⟩ zerlegt und auf jeden dieser Blöcke die Hashfunktion ⟨*hash-hash*⟩ angewandt, so daß Blöcke der Größe ⟨*hash-targetblock*⟩ herauskommen. (Zur Syntax von ⟨*hash*⟩ siehe Abschnitt B.5.)

Um aus einem Datenstrom nur Blöcke mit einer gewissen minimalen Symbolgewichtung auszuwählen, verwende man:

⟨*drop-source*⟩ ::= ‘`drop`’ ‘`{`’ (⟨*drop-subsource*⟩ | ⟨*drop-weight*⟩ | ⟨*drop-blocklen*⟩ | ⟨*drop-limit*⟩) `*` ‘`}`’

⟨*drop-weight*⟩ ::= ‘`weight`’ ⟨*weighting*⟩

⟨*drop-blocklen*⟩ ::= ‘`blocklen`’ ⟨*integer*⟩

⟨*drop-limit*⟩ ::= ‘`limit`’ ⟨*real*⟩

Dieser Extraktor zerlegt die Daten von ⟨*drop-subsource*⟩ in Blöcke der Länge ⟨*drop-blocklen*⟩ und gibt davon nur die aus, deren durch ⟨*drop-weight*⟩ bestimmte Symbolgewichtung mindestens ⟨*drop-limit*⟩ beträgt. Zur Syntax von ⟨*weight*⟩ siehe Abschnitt B.2.

Die adaptive Extraktion (Definition 4.7) schließlich implementiert:

⟨*adaptive-source*⟩ ::= ‘`adaptive`’ ‘`{`’ (⟨*adaptive-subsource*⟩ | ⟨*adaptive-blocklen*⟩ | ⟨*adaptive-weight*⟩ | ⟨*adaptive-hash*⟩ | ⟨*adaptive-spare*⟩) `*` ‘`}`’

⟨*adaptive-subsource*⟩ ::= ‘`source`’ ⟨*source*⟩

⟨*adaptive-blocklen*⟩ ::= ‘`blocklen`’ ⟨*integer*⟩

⟨*adaptive-weight*⟩ ::= ‘`weight`’ ⟨*weighting*⟩

⟨*adaptive-hash*⟩ ::= ‘`hash`’ ⟨*hash*⟩

⟨*adaptive-spare*⟩ ::= ‘`spare`’ ⟨*real*⟩

Es spezifizieren

- ⟨*adaptive-subsource*⟩ die Quelle X,
- ⟨*adaptive-blocklen*⟩ die Blocklänge n,
- ⟨*adaptive-weight*⟩ die Symbolgewichtung η,
- ⟨*adaptive-hash*⟩ die Familie von Hashfunktionen h,
- ⟨*adaptive-spare*⟩ die Konstante $c \geq 0$.

Dann ist ⟨*adaptive-source*⟩ die Quelle $\Xi^{n,m}_{\eta,h}(R, X)$, wobei m wie in Korollar 4.9 definiert sei.

Um die Aussage von Korollar 4.9 anwenden zu können, muß man zwischen den Extraktor und die Ursprungsquelle noch ein ⟨*fixedlen-source*⟩ schalten, es sei denn, man nimmt die in Korollar 4.9 verwendete Maximallänge l als so groß an, daß ein Erreichen dieser Länge unrealistisch ist.

Dieser Extraktor kann auch verwendet werden, um die Extraktionsrate für eine bestimmte Quelle praktisch zu bestimmen. Hierzu beschränkt man den aus der Quelle ausgegebenen Datenstrom auf eine bekannte Länge N (z. B. $N = 10^8$) und wendet den Extraktor darauf an. Aus der resultierenden Länge (welche z. B. mit ⟨*showsource*⟩ in Verbindung mit der Option '`totallen`' bestimmt werden kann) und N kann dann die Rate bestimmt werden. Da nur die Länge der Ausgabe relevant ist, bietet sich aus Geschwindigkeitsgründen die Verwendung der Hashfunktion ⟨*hash-fake*⟩ (siehe Abschnitt B.5) an.

B.2 Symbolgewichtungen

Um die Daten aus einer Quelle anzeigen und gewichten zu lassen, verwende man

⟨*showweight*⟩ ::= '`showweight`' '`{`' (⟨*showweight-weight*⟩ | ⟨*showweight-source*⟩ | ⟨*showweight-len*⟩ | '`verbose`' | ⟨*showweight-blocklen*⟩) * '`}`'

⟨*showweight-weight*⟩ ::= '`weight`' ⟨*weighting*⟩

⟨*showweight-source*⟩ ::= '`source`' ⟨*source*⟩

⟨*showweight-len*⟩ ::= '`len`' ⟨*integer*⟩

⟨*showweight-blocklen*⟩ ::= '`blocklen`' ⟨*integer*⟩

Es spezifiziere ⟨*showweight-weight*⟩ die Symbolgewichtung η und ⟨*showweight-source*⟩ die Quelle X. Weiter sei N durch ⟨*showweight-len*⟩ und n durch ⟨*showweight-blocklen*⟩ gegeben. Es werden dann die Gewichtungen

$$\eta(X_1 \dots X_{(i-1)n};\ X_{(i-1)n+1} \dots X_{in})$$

für $i = 1, \dots, \lfloor N/n \rfloor$ angezeigt. Ist noch '`verbose`' gegeben, wird die Ausgabe mit zusätzlichen Informationen versehen.

Die Symbolgewichtung selbst wird spezifiziert durch

⟨*weighting*⟩ ::= ⟨*weighting-explicit*⟩ | ⟨*weighting-table*⟩ | ⟨*weighting-relax*⟩ | ⟨*weighting-chmm*⟩

Explizites Angeben einer Symbolgewichtung ist dann möglich mit

⟨*weighting-explicit*⟩ ::= '`explicit`' '`{`' (⟨*explicit-symbols*⟩ | ⟨*explicit-prefixlen*⟩ | ⟨*explicit-blocklen*⟩ | ⟨*explicit-weightdata*⟩) * '`}`'

⟨*explicit-symbols*⟩ ::= '`symbols`' '`{`' ⟨*word*⟩ + '`}`'

⟨*explicit-prefixlen*⟩ ::= '`prefixlen`' ⟨*integer*⟩

⟨*explicit-blocklen*⟩ ::= '`blocklen`' ⟨*integer*⟩

⟨*explicit-weightdata*⟩ ::= '`data`' '`{`' ⟨*explicit-entry*⟩ * '`}`'

⟨*explicit-entry*⟩ ::= ⟨*word*⟩ * '`;`' ⟨*word*⟩ + '`->`' ⟨*real*⟩

Es seien n und m durch ⟨*explicit-blocklen*⟩[38] bzw. ⟨*explicit-prefixlen*⟩ und Σ durch ⟨*explicit-symbols*⟩ spezifiziert. Dann spezifiziert ⟨*weighting-explicit*⟩ eine Symbolgewichtung η über Σ mit

$$\eta(\xi\pi; \varrho) = \eta(\pi; \varrho) \qquad (\pi \in \Sigma^m,\ \varrho \in \Sigma^n,\ \xi \in \Sigma^*).$$

Dabei wird jedes $\eta(\pi; \varrho)$ ($\pi \in \Sigma^m, \varrho \in \Sigma^n$) durch ein ⟨*explicit-entry*⟩ bestimmt. Dies besteht aus π vor '`;`' und ϱ nach '`;`', hinter '`->`' steht der Wert von $\eta(\pi; \varrho)$ (also π '`;`' ϱ '`->`' $\eta(\pi; \varrho)$).

Ist ein $\eta(\pi; \varrho)$ nicht spezifiziert, so wird $\eta(\pi; \varrho) = 0$ angenommen.

Für $\eta(\alpha; \beta)$ mit $|\alpha| \neq n$ wird mittels Lemma 4.2 aus den $\eta(\xi\pi; \varrho)$ eine untere Abschätzung berechnet.

Insbesondere zur effizienteren Nutzung von konditioniert links-zeitinvarianten Quellen dient

⟨*weighting-table*⟩ ::= '`table`' '`{`' (⟨*table-subweighting*⟩ | ⟨*table-prefixlen*⟩ | ⟨*table-blocklen*⟩) * '`}`'

⟨*table-subweighting*⟩ ::= '`weight`' ⟨*weight*⟩

[38] In der aktuellen Version kann nur $n = 1$ sein. Deshalb darf ⟨*explicit-blocklen*⟩ auch weggelassen werden.

$\langle$*table-symbols*$\rangle$ ::= '`symbols`' '`{`' $\langle$*word*$\rangle$ + '`}`'
$\langle$*table-prefixlen*$\rangle$ ::= '`prefixlen`' $\langle$*integer*$\rangle$
$\langle$*table-blocklen*$\rangle$ ::= '`blocklen`' $\langle$*integer*$\rangle$

Für die durch $\langle$*weighting-table*$\rangle$ bestimmte Symbolgewichtung gilt alles oben für $\langle$*weighting-explicit*$\rangle$ gesagtes, mit der Ausnahme, daß Σ und alle $\eta(\pi;\varrho)$ $(\pi \in \Sigma^m, \varrho \in \Sigma^n)$ von der durch $\langle$*table-subweighting*$\rangle$ spezifizierten Symbolgewichtung η' übernommen werden, also

$$\eta(\xi\pi;\varrho) = \eta(\pi;\varrho) = \eta'(\pi;\varrho) \qquad (\pi \in \Sigma^m,\ \varrho \in \Sigma^n,\ \xi \in \Sigma^*).$$

Ist eine Symbolgewichtung gegeben, und soll diese um einen gewissen Betrag vermindert werden (z. B. um die Chancen zu verbessern, daß es sich um eine untere Schranke für die tatsächliche Symbolgewichtung handelt), so kann folgende Syntax verwendet werden:

$\langle$*weighting-relax*$\rangle$ ::= '`relax`' '`{`' ($\langle$*relax-subweighting*$\rangle$ | $\langle$*relax-amount*$\rangle$ | $\langle$*relax-type*$\rangle$) * '`}`'
$\langle$*relax-subweighting*$\rangle$ ::= '`weight`' $\langle$*weight*$\rangle$
$\langle$*relax-amount*$\rangle$ ::= '`amount`' $\langle$*real*$\rangle$
$\langle$*relax-type*$\rangle$ ::= '`persymbol`' | '`perblock`'

Ist dann η' die durch $\langle$*relax-subweighting*$\rangle$ spezifizierte Symbolgewichtung und ε durch $\langle$*relax-amount*$\rangle$ gegeben, so repräsentiert $\langle$*weighting-relax*$\rangle$ die Symbolgewichtung η mit

$$\eta(\alpha;x) := \begin{cases} \max\{\eta'(\alpha;x) - |x|\varepsilon,\ 0\}, & \text{falls } \langle \textit{relax-type} \rangle \text{ den Wert '}\texttt{persymbol}\text{' hat,} \\ \max\{\eta'(\alpha;x) - \varepsilon,\ 0\}, & \text{falls } \langle \textit{relax-type} \rangle \text{ den Wert '}\texttt{perblock}\text{' hat.} \end{cases}$$

Ist ein CHMM gegeben, so interessiert die zugehörige Symbolgewichtung:

$\langle$*weighting-chmm*$\rangle$ ::= '`chmm`' $\langle$*chmm*$\rangle$

Hierdurch wird die Gewichtung $\eta^{\mathcal{C}}$ angegeben,[39] wenn $\mathcal{C}$ das durch $\langle$*chmm*$\rangle$ definierte CHMM ist. Siehe Abschnitt B.4 zur Syntax von $\langle$*chmm*$\rangle$.

B.3 Tests

Tests können mittels folgender Syntax durchgeführt werden:

$\langle$*test*$\rangle$ ::= $\langle$*frequency-test*$\rangle$ | $\langle$*autocorrelation-test*$\rangle$ | $\langle$*serial-test*$\rangle$ | $\langle$*run-test*$\rangle$ | $\langle$*maurer-test*$\rangle$ |
$\langle$*weighting-test*$\rangle$
$\langle$*generic-test-param*$\rangle$ ::= $\langle$*test-param-source*$\rangle$ | $\langle$*test-param-length*$\rangle$ | $\langle$*test-param-level*$\rangle$
$\langle$*test-param-source*$\rangle$ ::= '`source`' $\langle$*source*$\rangle$
$\langle$*test-param-length*$\rangle$ ::= '`length`' $\langle$*integer*$\rangle$
$\langle$*test-param-level*$\rangle$ ::= '`level`' $\langle$*real*$\rangle$

Hierbei bestimmt $\langle$*test-param-source*$\rangle$ die zu testende Quelle (zum Format siehe Abschnitt B.1), weiterhin $\langle$*test-param-length*$\rangle$ die Länge der Stichprobe und $\langle$*test-param-level*$\rangle$ das Niveau des Tests.

Der Häufigkeitstest (Abschnitt 7.1.1) wird wie folgt beschrieben:

$\langle$*frequency*$\rangle$ ::= '`frequency`' '`{`' $\langle$*generic-test-param*$\rangle$ * '`}`'

Der Autokorrelationstest (Abschnitt 7.1.4) hat diese Syntax:

$\langle$*autocorrelation-test*$\rangle$ ::= '`autocorrelation`' '`{`' ($\langle$*generic-test-param*$\rangle$ | $\langle$*test-param-delay*$\rangle$) * '`}`'
$\langle$*test-param-delay*$\rangle$ ::= '`delay`' $\langle$*integer*$\rangle$

Die Verzögerung im Autokorrelationstest (in Abschnitt 7.1.4 mit τ bezeichnet) wird durch $\langle$*test-param-delay*$\rangle$ bestimmt.

Den Serientest (Abschnitt 7.1.2) erhält man durch

[39] siehe Definition 5.3

⟨serial-test⟩ ::= ‘serial’ ‘{’ (⟨generic-test-param⟩ | ⟨test-param-blocklen⟩) * ‘}’
⟨test-param-blocklen⟩ ::= ‘blocklen’ ⟨integer⟩

Hier wird die Blocklänge (L in Abschnitt 7.1.2) durch ⟨test-param-blocklen⟩ bestimmt.

Der Lauflängentest (Abschnitt 7.1.3) wird konfiguriert durch

⟨run-test⟩ ::= ‘run’ ‘{’ (⟨generic-test-param⟩ | ⟨test-param-maxrunlen⟩) * ‘}’
⟨test-param-maxrunlen⟩ ::= ‘maxlen’ ⟨integer⟩

Die maximale Lauflänge (L in Abschnitt 7.1.3) wird durch ⟨test-param-maxrunlen⟩ angegeben.

Ueli Maurers Universaltest (Abschnitt 7.1.5) beschreibt:

⟨maurer-test⟩ ::= ‘maurer’ ‘{’ (⟨generic-test-param⟩ | ⟨test-param-blocklen⟩ | ⟨test-param-prefixlen⟩) * ‘}’
⟨test-param-blocklen⟩ ::= ‘blocklen’ ⟨integer⟩
⟨test-param-prefixlen⟩ ::= ‘prefix’ ⟨integer⟩

Dabei gibt ⟨test-param-blocklen⟩ die Blocklänge (L in Abschnitt 7.1.5) und ⟨test-param-prefixlen⟩ die Länge des Präfixes für die Vorverarbeitung (in Symbolen, nicht in Blöcken; in Abschnitt 7.1.5 als Q bezeichnet) an.

Den Gewichtungstest aus Abschnitt 7.2 erhält man über

⟨weighting-test⟩ ::= ‘weighting’ ‘{’ (⟨generic-test-param⟩ | ⟨test-param-blocklen⟩ |
    ⟨test-param-weightprefix⟩ | ⟨test-param-weightblocklen⟩ | ⟨test-param-weight⟩ |
    ⟨test-param-samplesize⟩ | ⟨test-param-estimate⟩) * ‘}’
⟨test-param-blocklen⟩ ::= ‘blocklen’ ⟨integer⟩
⟨test-param-weightprefix⟩ ::= ‘weightprefix’ ⟨integer⟩
⟨test-param-weightblocklen⟩ ::= ‘weightblocklen’ ⟨integer⟩
⟨test-param-weight⟩ ::= ‘weight’ ⟨weighting⟩
⟨test-param-samplesize⟩ ::= ‘samplesize’ ⟨integer⟩
⟨test-param-estimate⟩ ::= ‘estimate’

Hierbei bestimmen ⟨test-param-blocklen⟩ die Blocklänge (L in Definition 7.1), ⟨test-param-weightprefix⟩ die Länge von π, ⟨test-param-weightblocklen⟩ die Länge von ϱ. Weiter bestimmt ⟨test-param-weight⟩ eine Symbolgewichtung $\tilde{\eta}$, siehe Abschnitt B.2 zum Format von ⟨weight⟩. Die Stichprobengröße M wird schließlich durch ⟨test-param-samplesize⟩ bestimmt ($M := 0$, falls keine Stichprobengröße angegeben wird).

Es wird dann für alle $\pi, \varrho \in \Sigma^*$ der vorgegebenen Länge der Gewichtungstest für $\eta(\ldots\pi;\varrho) \geq \varepsilon$ mit Stichprobengröße M durchgeführt, wobei Σ das Alphabet der durch ⟨test-param-source⟩ spezifizierten Quelle ist und $\varepsilon := \tilde{\eta}(\pi;\varrho)$.

Ist ⟨test-param-estimate⟩ gegeben, so muß keine Symbolgewichtung angegeben werden, sondern es wird eine Schätzung für η erstellt, die gerade so groß ist, daß sie für die gegebene Stichprobe den Test für alle π, ϱ besteht.

Die Ausgabe dieses Tests enthält am Ende noch eine Ausgabe der geschätzten bzw. getesteten Quelle in für `randomextract` lesbarer Form (Nichtterminal ⟨weighting⟩, siehe Abschnitt B.2), wobei im Falle der getesteten Quelle alle Funktionswerte von η, die den Test nicht bestanden haben, auf 0 gesetzt werden.

B.4 CHMM

Ein CHMM $\mathcal{C}$ wird wie folgt angegeben:

⟨chmm⟩ ::= ‘{’ (⟨chmm-symbols⟩ | ⟨chmm-states⟩ | ⟨chmm-transitions⟩) * ‘}’
⟨chmm-symbols⟩ ::= ‘symbols’ ‘{’ ⟨word⟩ + ‘}’
⟨chmm-states⟩ ::= ‘states’ ‘{’ ⟨word⟩ + ‘}’
⟨chmm-transitions⟩ ::= ‘transitions’ ‘{’ ⟨transition-domain⟩ * ‘}’
⟨transition-domain⟩ ::= ⟨word⟩ ⟨probability-set⟩

Diese Deklaration besteht aus drei Teilen: Zuerst werden mit ⟨*chmm-symbols*⟩ bzw. ⟨*chmm-states*⟩ das Alphabet $\Sigma = \Sigma_{\mathcal{C}}$ und die Zustandsmenge $Q = Q_{\mathcal{C}}$ angegeben. Danach werden in ⟨*chmm-transitions*⟩ die Transitionsbereiche definiert. Jedes ⟨*transition-domain*⟩ besteht aus dem Namen eines Zustands q und danach einer Teilmenge von $\mathbb{R}_1^{\Sigma\times Q}$, angegeben durch ⟨*probability-set*⟩.

Eine Teilmenge von $\mathbb{R}_1^{\Sigma\times Q}$ kann auf zwei Arten geschrieben werden:

⟨*probability-set*⟩ ::= ⟨*interval-probability-set*⟩ | ⟨*convex-probability-set*⟩

Die erste Art eignet sich zur Darstellung von Quadern (genaugenommen zur Darstellung von Quadern geschnitten mit $\mathbb{R}_1^{\Sigma\times Q}$):

⟨*interval-probability-set*⟩ ::= '`interval`' '`{`' ⟨*arrow-probability*⟩ * '`}`'

⟨*arrow-probability*⟩ ::= ⟨*transition*⟩ ⟨*interval*⟩

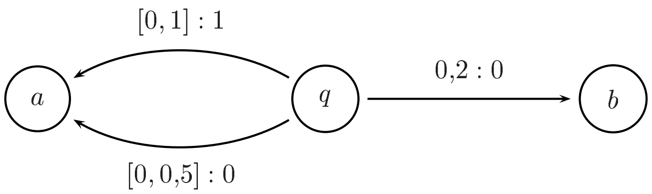
⟨*transition*⟩ ::= ⟨*word*⟩ ⟨*word*⟩

⟨*interval*⟩ ::= ⟨*real*⟩ | '`[`' ⟨*real*⟩ '`,`' ⟨*real*⟩ '`]`'

Diese Darstellung lehnt direkt an die Darstellung des CHMM durch ein Diagramm an. Zu jedem vom aktuellen Zustand ausgehenden Pfeil existiert eine ⟨*arrow-probability*⟩. Dieses Nichtterminal besteht wiederum aus der Bezeichnung des Pfeils (⟨*transition*⟩, bestehend aus Zielzustand und auszugebendem Symbol) und einem abgeschlossenen Teilintervall von $[0, 1]$ (fallen Minimum und Maximum des Intervalls zusammen, so kann stattdessen auch eine Zahl angegeben werden). Ein Beispiel für ⟨*transition-domain*⟩ wäre dann

```
q interval { a 1 [0,1]    a 0 [0,.5]    b 0 .2 }
```

Dies entspräche folgendem Teildiagramm:



Beliebige endlich repräsentierbare CHMM lassen sich bis auf konvexe Äquivalenz mit folgender Syntax beschreiben:

⟨*convex-probability-set*⟩ ::= '`convex`' '`{`' ⟨*vertex*⟩ + '`}`'

⟨*vertex*⟩ ::= '`point`' '`{`' ⟨*arrow-probability*⟩ * '`}`'

Ein ⟨*convex-probability-set*⟩ besteht aus mehreren Eckpunkten ⟨*vertex*⟩, von denen jeder einen Quader oder Punkt im $\mathbb{R}_{\geq 0}^{\Sigma\times Q}$ beschreibt (die Semantik ist analog zu ⟨*interval-probability-set*⟩ oben). Dann ist der von ⟨*convex-probability-set*⟩ beschriebene Transitionsbereich $\mathcal{C}_q$ die konvexe Hülle aller dieser Quader und Punkte.

Die Quader sollten möglichst klein sein, man versuche, beim Entwurf mit Punkten zu arbeiten, und danach die Punkte, die sich nicht exakt auf dem Computer darstellen lassen, durch einen minimalen umschließenden Quader zu ersetzen (z. B. schreiben wir $[0.33333333, 0.33333334]$ für $\frac{1}{3}$).

Als Beispiel diene die Datei `notinterval.chmm`.

Alle in dieser Arbeit vorgestellten CHMM sind der Software beigefügt, sie haben Dateinamen der Form `*.chmm`.

B.5 Hashfunktionen

Eine Hashfunktion wird wie folgt spezifiziert:

⟨*hash*⟩ ::= ⟨*hash-toeplitz*⟩ | ⟨*hash-fake*⟩

⟨*hash-toeplitz*⟩ ::= '`toeplitz`' '`{`' (⟨*toeplitz-random*⟩ | '`triangle`' | '`check`') '`}`'

⟨*toeplitz-random*⟩ ::= '`random`' ⟨*source*⟩

⟨*hash-fake*⟩ ::= '`fake`'

Man sieht, daß bei keiner der Hashfunktionen die Länge der Ein- oder Ausgabe spezifiziert ist. Dies liegt daran, daß diese von verwendenden Kontext bestimmt wird (z. B. von der Blocklänge des Extraktors). Andererseits enthält das so konstruierte Objekt bereits den in die Hashfunktion einfließenden initialen Zufall, es handelt sich formal also um eine Zufallsvariable H der Form

$$\begin{array}{rccc} H: & \mathbb{N} \times \mathbb{N} \times \Sigma^* & \longrightarrow & \Sigma^* \\ & n,\ m,\ (a_1, \ldots, a_n) & \longmapsto & (b_1, \ldots, b_m). \end{array}$$

Der einzige implementierte Typ von universellen Quasi-Hashfunktionen ist ⟨*hash-toeplitz*⟩, dieser realisiert das Anwenden von Toeplitz-Matrizen (wie in Lemma 3.10). Mit ⟨*toeplitz-random*⟩ kann eine Quelle angegeben werden, der der initiale Zufall zu entnehmen ist, Voreinstellung ist unter Linux der Kernel-Zufallszahlengenerator (`linuxkernel real`), sonst der C-Pseudozufallszahlengenerator (`crng 2`).

Ist '`check`' gegeben, so wird die Applikation der Toeplitz-Matrix zusätzlich zu dem optimierten Verfahren mit einem langsamen aber einfachen Verfahren durchgeführt und die Ergebnisse verglichen. Dies dient zur Verifikation der Software und hat keinen Einfluß auf das Ergebnis.

Ist '`triangle`' gegeben, so wird statt einer zufälligen Toeplitz-Matrix eine zufällige Toeplitz-Matrix mit konstant 1 auf der Diagonale und konstant 0 unterhalb der Diagonale verwendet. Dann ist ⟨*hash-toeplitz*⟩ allerdings keine universelle Quasi-Hashfunktion mehr.

Die aktuelle Implementation von ⟨*hash-toeplitz*⟩ kann nur auf einem Alphabet der Größe 2 operieren.

Interessiert das Ergebnis der Hashfunktion nicht, sondern nur dessen Länge, so kann ⟨*hash-fake*⟩ genutzt werden. Hier ist die Ausgabe nicht spezifiziert, dafür ist die Evaluation dieser Funktion wesentlich schneller möglich. Man kann ⟨*hash-fake*⟩ beispielsweise benutzen, um die Rate eines Extraktionsverfahrens effizienter zu bestimmen.

Anhang C

Definitionen und Aussagen

D Literatur

[Bea91] BEAVER, D.: *Secure Multiparty Protocols and Zero-Knowledge Proof Systems Tolerating a Faulty Minority.* Journal of Cryptology: the journal of the International Association for Cryptologic Research, 4(2):75–122, 1991.

[Blu86] BLUM, M.: *Independent Unbiased Coin Flips From a Correlated Biased Source: A Finite State Markov Chain.* Combinatorica, 6(2):97–108, 1986.

[Can00] CANETTI, RAN: *Universally Composable Security: A New Paradigm for Cryptographic Protocols.* `http://eprint.iacr.org/2000/067` und ECCC TR 01-24. Ausführliche Zusammenfassung in 42nd FOCS, 2000.

[CG85] CHOR, BENNY und ODED GOLDREICH: *Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity (Extended Abstract).* In: *IEEE Symposium on Foundations of Computer Science*, Seiten 429–442, 1985. Online verfügbar unter `ftp://theory.lcs.mit.edu/pub/people/oded/sources.ps`.

[CG88] CHOR, BENNY und ODED GOLDREICH: *Unbiased bits from sources of weak randomness and probabilistic communication complexity.* SIAM Journal on Computing, 17(2):230–261, 1988. Ein ausführliche Zusammenfassung findet sich in [CG85].

[Eli72] ELIAS, P.: *The Efficient Construction of an Unbiased Random Sequence.* Ann. Math. Statist., 43(3):865–870, 1972.

[GW94] GOLDREICH, ODED und AVI WIGDERSON: *Tiny Families of Functions with Random Properties: A Quality-Size Trade-off for Hashing (Preliminary Version).* In: *ACM Symposium on Theory of Computing*, Seiten 574–583, 1994.

[Haa02] HAAG, MATHIAS: *Extraktion von Zufall aus physikalischen Quellen.* Diplomarbeit, Institut für Algorithmen und Kognitive Systeme, Universität Karlsruhe, Juli 2002.

[HILL93] HÅSTAD, J., R. IMPAGLIAZZO, L. LEVIN und M. LUBY: *Construction of a Pseudorandom Generator from any One-Way Function*, 1993. Online verfügbar unter `http://www-cse.ucsd.edu/users/russell/sicomp.ps`. Eine neuere Version ist [HILL99].

[HILL99] HÅSTAD, JOHAN, RUSSELL IMPAGLIAZZO, LEONID A. LEVIN und MICHAEL LUBY: *A Pseudorandom Generator from any One-way Function.* SIAM J. Comput., 28(4):1364–1396, 1999. Online verfügbar unter `http://www.icsi.berkeley.edu/~luby/PAPERS/hill.ps`.

[Mau90] MAURER, UELI: *A Universal Statistical Test for Random Bit Generators.* In: *Advances in Cryptology — CRYPTO '90*, Band 537 der Reihe *Lecture Notes in Computer Science*, Seiten 409–420. Springer-Verlag, 1990. Endgültige Fassung: [Mau92].

[Mau92] MAURER, UELI: *A Universal Statistical Test for Random Bit Generators.* Journal of Cryptology, 5(2):89–105, 1992. Vorläufige Fassung: [Mau90].

[Nis96] NISAN, NOAM: *Extracting randomness: How and why: A survey.* In: *Proceedings of the 11th Annual IEEE Conference on Computational Complexity*, Seiten 44–58, Philadelphia, Pennsylvania, Mai 1996. IEEE Computer Society Press. Online verfügbar unter `http://www.cs.huji.ac.il/~noam/dispersers.ps`.

[NTS95] NISAN, NOAM und AMNON TA-SHMA: *Extracting Randomness: A Survey and New Constructions.* Journal of Computer and System Sciences, 52(1):43–52, 1995.

[Per92] PERES, YUVAL: *Iterating Von Neumann's Procedure for Extracting Random Bits.* Annals of Statistics, 20(1):590–597, März 1992. Online verfügbar unter `http://www.stat.berkeley.edu/~peres/mine/vn.pdf`.

[PW94] PFITZMANN, BIRGIT und MICHAEL WAIDNER: *A General Framework for Formal Notions of "Secure" Systems, Hildesheimer Informatik-Berichte 11/94.* Technischer Bericht, Universität Hildesheim, 1994. Online verfügbar unter `http://www.semper.org/sirene/publ/PfWa_94FormalItsecIB.ps.gz`.

[Rab90] Rabiner, L. R.: *A Tutorial on Hidden Markov Models and Selected Apllications in Speech Recognition*. In: Waibel, A. und K.-F. Lee (Herausgeber): *Readings in Speech Recognition*, Seiten 267–296. Kaufmann, San Mateo, CA, 1990.

[Sha48] Shannon, Claude E.: *A mathematical theory of communication*. Bell Systems Technical Journal, 27(3):379–423, Juli 1948. Fortgesetzt in 27(4):623-656, Oktober 1948. Online verfügbar unter `http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html`.

[Sti02] Stinson, Douglas R.: *Universal hash families and the leftover hash lemma, and applications to cryptography and computing*. J. Combin. Math. Combin. Comput., 42:3–31, 2002. Online verfügbar unter `http://www.cacr.math.uwaterloo.ca/~dstinson/papers/leftoverhash.ps`.

[SV86] Santha, Miklos und Umesh Vazirani: *Generating quasi-random sequences from semi-random sources*. Journal of Computer and System Sciences, 33(1):75–87, 1986.

[SZ94] Srinivasan, Aravind und David Zuckerman: *Computing with Very Weak Random Sources*. In: *IEEE Symposium on Foundations of Computer Science*, Seiten 264–275, 1994.

[Tre99] Trevisan, Luca: *Construction of extractors using pseudo-random generators (extended abstract)*. In: *Proceedings of STOC*, Seiten 141–148, 1999.

[Unr02] Unruh, Dominique: *Formal Security in Quantum Cryptography*. Studienarbeit, Institut für Algorithmen und Kognitive Systeme, Universität Karlsruhe, Dezember 2002. Online verfügbar unter `http://www.unruh.de/DniQ/ueb/quantum_security.ps.gz`.

[vN51] Neumann, John von: *Various techniques used in connection with random digits*. Applied Mathematics Series, 12:36–38, 1951.

[Zuc97] Zuckerman, David: *Randomness-optimal oblivious sampling*. Random Structures and Algorithms, 11(4):345–367, 1997. Online verfügbar unter `http://www.cs.utexas.edu/users/diz/pubs/sampler.ps`.

E Symbolverzeichnis

M^*	Abbrechende Folgen über M	13
A^*	initiale Verteilung des CHMM-Adversaries A	30
xy	Konkatenation von x und y	13
$\|x\|$	Länge der Folge x	13
$\perp$	undefiniertes Ergebnis	12
$\\|x\\|_1$	Betragssummennorm von x	12
$\mathbb{1}_n$	Einheitsmatrix in $\mathbb{F}^{n\times n}$	12
$\mathrm{Adv}_{\mathcal{C}}$	Menge aller Adversaries zum CHMM $\mathcal{C}$	30
$\mathcal{C}_q$	Transitionsbereich des CHMM $\mathcal{C}$ vom Zustand q aus	29
$\mathcal{F}_{\mathrm{ARnd},\Sigma}$	Funktionalität. Modelliert abbrechende Zufallsquelle über Σ	39
$\mathcal{F}_{\mathrm{Rnd},\Sigma}$	Funktionalität. Modelliert nicht abbrechende Zufallsquelle über Σ	39
$\mathcal{F}_{\mathcal{X}}$	Funktionalität. Modelliert die parametrische Quellenfamilie $\mathcal{X}$	40
$H(X)$	(Shannon-)Entropie von X	14
$H_\infty(X)$	min-Entropie von X	14
$H_{\mathrm{Ren}}(X)$	Renyi-Entropie von X	14
$I_{\mathcal{X}}$	Indexmenge der parametrischen Familie von Quellen $\mathcal{X}$	37
inf	Infimum	12
log	Logarithmus zur Basis 2 ($\log_2$)	12
max	Maximum	12
min	Minimum	12
$M^{\mathbb{N}}$	Folgen über M	13
$\mathbb{N}$	Menge der natürlichen Zahlen ohne 0	12
$M^{\mathbb{N}_0}$	Folgen über M, erster Index ist 0	13
$\mathbb{N}_0$	Menge der natürlichen Zahlen einschließlich der 0	12
$P_{B=b}$	Wahrscheinlichkeitsverteilung mit $B=b$	13
$Q_{\mathcal{C}}$	Zustandsmenge des CHMM $\mathcal{C}$	29
$\mathbb{R}$	Menge der reellen Zahlen	12
$R(\mathcal{X})$	Rate der Quellenfamilie $\mathcal{X}$	25
$R(\mathcal{X},\eta)$	Rate der Quellenfamilie $\mathcal{X}$ mit der Symbolgewichtung η	25
$R(X,\mathcal{X})$	Rate der Quelle X in der Quellenfamilie $\mathcal{X}$	25
$R(\eta,X)$	Rate der Quelle X mit der Symbolgewichtung η	25
$\mathbb{R}_{\geq 0}$	Menge der nichtnegativen reellen Zahlen	12
$\mathbb{R}_{>0}$	Menge der positiven reellen Zahlen	12
$\mathbb{R}_1^M$	normierte Elemente von $\mathbb{R}_{\geq 0}^M$	12
$\mathrm{SD}(X;Y\\|E)$	Kurzform für $\mathrm{SD}(X\|E;Y\|E)$	15
$\mathrm{SD}(X;Y)$	statistischer Abstand zwischen X und Y	15
sup	Supremum	12
$\mathrm{Toeplitz}(\mathbb{F}^{m\times n})$	$(m\times n)$-Toeplitz-Matrizen	12
X^A	CHMM-Quelle zum Adversary A	30
$\mathcal{X}^{\mathcal{C}}$	Familie aller $\mathcal{C}$-Quellen ($\mathcal{C}$ CHMM)	31
$\delta(A)$	Ist die Aussage A wahr, dann $\delta(A)=1$, sonst $\delta(A)=0$	13
$\eta^{\mathcal{C}}$	Symbolgewichtung zum CHMM $\mathcal{C}$	31
$\eta^{\mathcal{X}}$	Symbolgewichtung der Familie $\mathcal{X}$ von Quellen	21
λ	leeres Wort	13
$\Xi_{\eta,h}^{n,m}$	adaptiver Hash-Extraktor	23

F Index