MATHEMATISCHE GRUNDLAGEN DES QUANTENRECHNENS

Ulrich Faigle

Skriptum zur Vorlesung

Sommersemester 2003
Universität zu Köln

Universität zu Köln
Mathematisches Institut
Zentrum für Angewandte Informatik
Weyertal 80
faigle@zpr.uni-koeln.de
www.zaik.uni-koeln.de/AFS

Inhaltsverzeichnis

KAPITEL 1

Rechner, Bits und Schaltkreise

In einem (sehr allgemeinen) schematischen Modell ist ein „Rechner“ ein schwarzer Kasten R, der mit einer Eingabeeinheit X und einer Ausgabeeinheit Y versehen ist:

$$x \longrightarrow \boxed{\boxed{X} \rightarrow \boxed{R} \rightarrow \boxed{Y}} \longrightarrow y\ .$$

Die Arbeitsweise ist so: Bei Eingabe von x in X geschieht etwas, das durch „R“ beschrieben wird. Schliesslich kann man an der Ausgabeeinheit Y ein „Rechenergebnis“ y ablesen.

BEMERKUNG. Es wird an dieser Stelle *nicht* vorausgesetzt, dass der Rechner eine „Funktion“ im mathematischen Sinne repräsentiert, d.h. dass das Rechenergebnis y durch die Eingabe x eindeutig bestimmt ist. (Im Fall klassischer deterministischer Rechner ist das allerdings so.)

1. Bits, Sprachen und Boolesche Algebren

Wir nehmen an, dass die möglichen Eingaben x und abgelesenen Rechenergebnisse y im Rahmen eines definitiven Kontextes (dessen konkrete Interpretation hier nicht interessiert) in (endliche) Folgen von Symbolen „0“ und „1“ kodiert sind. Z.B.:

$$x = 10011 \quad \text{oder} \quad y = 001000111110 \quad \text{usw.}$$

Um dies etwas zu formalisieren, betrachten wir die Symbolmenge $A = \{0, 1\}$ als ein Alphabet, aus dem wir alle (formal möglichen) *Wörter* der *Länge* n bilden:

$$A^n := \{0, 1\}^n \quad (n \in \mathbb{N}) \quad (\text{mit } A^0 := \{\emptyset\})\ .$$

Wir erhalten damit die zu dem Alphabet A gehörende *Sprache*

$$A^* := \bigcup_{n=0}^{\infty} A^n\ .$$

TERMINOLOGIE: Die Buchstaben des Alphabets $A = \{0, 1\}$ heissen *(Boolesche) Bits*.

Abhängig von dem Standpunkt, den wir einnehmen, tragen die Wortmengen A^n (bzw. die gesamte Sprache A^*) gewisse algebraische Strukturen, die wir nun diskutieren.

1.1. Verkettungen. Ein Wort $w = w_1 w_2 \dots w_m \in A^m$ können wir mit einem Wort $v = v_1 v_2 \dots v_n \in \mathbb{A}^n$ zu einem neuen Wort *verketten*, indem wir einfach die Buchstaben hintereinanderreihen:

$$w \otimes v := w_1 w_2 \dots w_m v_1 v_2 \dots v_n \ \in \ A^{m+n} \ .$$

Damit ist $(A^*, \otimes)$ eine sog. *Halbgruppe mit Neutralelement und Kürzungsregel*, d.h. es gilt:

(1) $(u \otimes v) \otimes w = u \otimes (v \otimes w)$ (Assoziativität).
(2) $w\emptyset = \emptyset w = w$ (Neutralelement).
(3) $u \otimes w = v \otimes w \iff u = v$ (Kürzungsregel).

1.2. Addition (mod 2). Wir können uns auch vorstellen, dass 0 und 1 die Elemente des Körpers $\mathbb{Z}_2 = (\{0,1\}, \oplus)$ sind, die nach der folgenden Tabelle addiert werden:

$\oplus$	0	1
0	0	1
1	1	0

Wenn wir uns nun ein Wort $w \in A^*$ einfach als den Anfangsteil eines unendlich langen Vektors denken, dessen übrige Komponenten mit 0 aufgefüllt sind:

$$w = w_1 w_2 \dots w_n \in A^* \quad \longleftrightarrow \quad (w_1, w_2, \dots, w_n, 0, 0, \dots) \ \in \ \mathbb{Z}_2^{\mathbb{N}} \ ,$$

dann können wir Wörter (wie Vektoren) komponentenweise addieren:

$$(1,1,0,1,0,0,1,0,0,\dots) \oplus (1,0,0,1,0,0,\dots) = (0,1,0,0,\dots)$$

Insbesondere entspricht in dieser Sichtweise die Menge $A^n = \{0,1\}^n$ aller Wörter der Länge n der Menge $\mathbb{Z}_2^n$ aller Koordinatenvektoren mit n Komponenten (über dem Skalarbereich $\mathbb{Z}_2$).

1.3. Boolesche Algebra. Wir können uns ferner vorstellen, dass wir eine abzählbar-unendliche Grundmenge M (d.h. $|M| = |\mathbb{N}|$) vorliegen haben, deren endliche Teilmengen wir durch die Wörter aus A^* beschreiben:

$$w = w_1 w_2 \dots w_n \quad \longleftrightarrow \quad \{k \in \mathbb{N} \mid w_k = 1\}$$

Beschränken wir uns auf die Wörter der Länge n, dann können wir M als endlich ($|M| = n$) annehmen. Die mengentheoretischen Operationen von Vereinigung, Durchschnitt und Komplementbildung führen wieder auf Operationen, die „komponentenweise“ angewendet werden:

$\vee$	0	1
0	0	1
1	1	1

$\wedge$	0	1
0	0	0
1	0	1

$\neg$	0	1
	1	0

BEMERKUNG(LOGISCHE VERKNÜPFUNGEN). Die Verknüpfungen entsprechen sog. *logischen Verküpfungen* der Aussagagelogik bzgl. der sog. *Wahrheitswerte* „0“ und „1“. In

diesem Modell nimmt man

$$\begin{array}{lcl} \vee & \longleftrightarrow & ODER \\ \wedge & \longleftrightarrow & UND \\ \neg & \longleftrightarrow & NEGATION \end{array}$$

Die Addition mod 2 („$\oplus$“) wird in diesem Zusammenhang als sog. *AUSSCHLIESSENDES ODER* („$XODER$“) interpretiert.

Betrachten wir A^n als von den mengentheoretischen Operationen induzierte Struktur, dann nennen wir A^n auch eine *Boolesche Algebra* und benutzen die Notation

$$\mathcal{B}_n = (A^n; \vee, \wedge, \neg, \oplus) \ .$$

Tatsächlich könnten wir uns auf die Operationen $\wedge$ und $\neg$ beschränken, denn wir können die übrigen durch diese ausdrücken:

$$\begin{array}{rcl} x \oplus y & = & \neg(\neg(\neg x \wedge y) \wedge \neg(x \wedge \neg y)) \\ x \vee y & = & \neg(\neg x \wedge \neg y) \ . \end{array}$$

1.4. Natürliche Zahlen. Jede natürliche Zahl $d < 2^n$ hat eine eindeutige Darstellung zur Grundzahl $g = 2$:

$$d = d_0 2^{n-1} + d_1 2^{n-1} + \ldots + d_{n-1} \quad (d_k \in \{0, 1\}) \ .$$

Also können wir ein Wort $d = d_0 d_1 \ldots d_{n-1} \in A^n$ auch einfach als die Binärdarstellung einer natürlichen Zahl $< 2^n$ mit den Ziffern d_k ansehen. In diesem Sinn haben wir:

$$A^n \quad \longleftrightarrow \quad \{d \in \mathbb{N} \mid 0 \leq d < 2^n\} \ .$$

Mit der Binärdarstellung rechnet man genauso wie mit der (aus der Schule antrainierten) Dezimaldarstellung von (reellen) Zahlen. Z.B. kann man auch Brüche per Kommadarstellung wiedergeben:

$$d_0 d_1 \ldots d d_m . d_{m+1} + \ldots + d_{n-1} := \sum_{k=0}^{m} d_k 2^k + \sum_{j=1}^{n-1-m} \frac{d_{m+j}}{2^j}$$

BEMERKUNG. Um die Verwechslung mit dem Komma bei Vektorkoordinaten zu reduzieren, schreiben wir hier – im Gegensatz zu dem im Deutschen sonst üblichen – das „Binärkomma“ als „Punkt“.

2. Boolesche Funktionen

Unter einer Booleschen Funktion verstehen wir zuerst einfach eine Funktion

$$f : A^* \to \{0, 1\}$$

Für die Analyse ist es geschickt, f nach der Anzahl der Eingabebits zu unterscheiden. D.h. wir betrachten für alle $n \in \mathbb{N}$ die Funktionen

$$f^{(n)} : A^n \to \{0, 1\} \quad \text{mit} \quad f^{(n)}(x) := f(x) \ .$$

2.1. Normalformen. Für jede Komponente k der Vektoren (bzw. Wörter) in A^n führen wir nun eine Variable x_k mit Werten in $\{0,1\}$ ein und ordnen jedem $d = d_0 \dots d_{n-1} \in A^n$ die folgende Funktion zu:

$$X_d = y_0 \wedge \ldots \wedge y_{n-1} \quad \text{mit} \quad y_k := \begin{cases} x_k & \text{falls } d_k = 1, \\ \neg x_k & \text{falls } d_k = 0. \end{cases}$$

Dann erhalten wir eine Funktion $X_d : A^n \to \{0,1\}$ mit der Eigenschaft

$$X_d(x_0, \ldots, x_{n-1}) = 1 \quad \Longleftrightarrow \quad x_0 x_1 \dots x_{n-1} = d \; .$$

Also finden wir die Darstellung einer allgemeinen Funktion $f : A^n \to \{0,1\}$ in der sog. *disjunktiven Normalform* („DNF“):

$$\boxed{f(x) = \bigvee_{f(d)=1} X_d(x)}$$

NOTA BENE: *Die DNF zeigt, dass jede Boolesche Funktion allein mit den Operationen $\vee, \wedge$ und $\neg$ dargestellt werden kann (und deshalb im Prinzip schon mit $\wedge$ und $\neg$, wie oben bemerkt).*

2.2. Allgemeinere Boolesche Funktionen. In der Praxis interessiert man sich für Rechenmaschinen, die Funktionen f vom folgenden Typ berechnen:

$$f : A^n \to A^m \; .$$

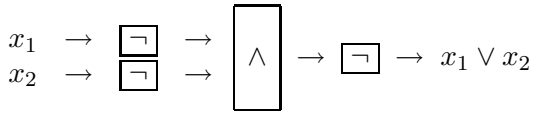
Eine solche Funktion setzt sich aus m Komponentenfunktionen $f_i : A^n \to \{0,1\}$ zusammen:

$$f_i(x) = i\text{te Komponente von } f(x) \quad (i = 1, \ldots, m) \; .$$

Das Problem, eine Maschine zur Berechnung von f zu bauen, ist folglich äquivalent zum Problem, eine Maschine zur Berechnung der m Booleschen Funktionen f_i zu konstruieren. Wir nennen deshalb auch die vektorwertige Funktion f eine *Boolesche Funktion*.

3. Boolesche Schaltkreise

Wir nehmen nun an, wir hätten Bauelemente (sog. *Gatter*), welche die Booleschen Operationen $\wedge$ und $\neg$ realisieren. Dann können wir natürlich auch die übrigen Operationen verwirklichen. Z.B. erhalten wir „$\vee$“ folgendermassen:



Wir können also annehmen, dass unser Baukasten auch Elemente für $\vee$ und $\oplus$ enthält. Ebenso ist klar, dass man eine Boolesche Operation mit n Eingabevariablen durch eine Reihe entsprechender Operationen mit nur 2 Eingabevariablen nachbauen kann. Z.B.

$$x_1 \wedge x_2 \wedge \ldots \wedge x_n = (\ldots((x_1 \wedge x_2) \wedge x_3) \wedge \ldots) \wedge x_n \; .$$

FOLGERUNG: *Aus den Bausteinen für $\vee$ und $\neg$ kann man im Prinzip Bauelemente zur Berechnung von beliebigen Funktionen $f : A^n \to \{0,1\}$ konstruieren.*

Wir formalisieren diese Konstruktionsidee und definieren einen *Booleschen Schaltkreis* als einen azyklischen gerichteten Graphen, dessen Knoten markiert sind gemäss

(1) Es gibt n *Eingabeknoten*, die mit den Eingabevariablen $x_0, \ldots, x_{n-1}$ markiert sind. Ein Eingabeknoten hat Eingangsgrad 0 und Ausgangsgrad 1.
(2) Es gibt 1 *Ausgabeknoten*, der mit einer Ausgabevariablen y markiert ist. Der Ausgabeknoten hat Eingangsgrad 1 und Ausgangsgrad 0.
(3) Alle übrigen Knoten sind mit Booleschen Operation markiert. Ausgangsgrad und Eingangsgrad sind entsprechend der markierten Booleschen Operation.

Wir sagen, der Boolesche Schaltkreis *berechnet* die Funktion $f : A^n \to \{0,1\}$, wenn bei Eingabe von x bei den Eingangsknoten die Booleschen Operationen zu dem Wert $y = f(x)$ am Ausgabeknoten führen.

Wir haben gesehen, dass es zu jeder Booleschen Funktion $f : A^n \to \{0,1\}$ einen Booleschen Schaltkreis gibt, der diese berechnet.

Die Definition des Booleschen Schaltkreises lässt sich natürlich sofort zur Berechnung von Booleschen Funktionen vom Typ

$$f : A^n \to A^m$$

erweitern, wenn man entsprechend viele Ausgabeknoten $y_1, \ldots, y_m$ einführt.

Steuerung von Gattern. Betrachten wir nochmals ein Gatter für die Operation $\oplus$:

$$\begin{array}{rcl} x & \to & \\ z & \to & \boxed{\oplus} \end{array} \to \begin{cases} x & \text{falls } z = 0 \\ \neg x & \text{falls } z = 1 \end{cases}$$

Wir können also „$\oplus$“ als ein Gatter für x(!) auffassen, das durch das *Kontrollbit* z gesteuert wird. Im Fall $z = 0$ zeigt das Gatter keine Wirkung. Im Fall $z = 1$ wirkt das Gatter wie die Negation „$\neg$“ auf x.

Verteilungsknoten. Um den Formalismus klein zu halten, wollen wir noch einen weiteren Typ von Knoten als sog. „Verteilungsknoten“ zulassen. Ein Verteilungsknoten hat Eingangsgrad 1 und Ausgangsgrad 2 und entspricht folgendem Bauelement:

$$x \to \boxed{V} \begin{array}{l} \to \; x \\ \to \; x \end{array}$$

Über Verteilungsknoten kann ein Kontrollbit an mehrere Gatter gleichzeitig geleitet werden, um diese zu steuern.

3.1. Komplexität von Booleschen Schaltkreisen und Funktionen. Wir haben gesehen, dass es zu jeder Booleschen Funktion $f : A^n \to \{0,1\}$ einen Booleschen Schaltkreis gibt, der diese berechnet. Der ist aber nicht eindeutig bestimmt. Insbesondere gibt es „einfachere“ und kompliziertere Realisierungen von f. Als Mass für die Komplexität des Schaltkreises nehmen wir die Anzahl seiner Gatter vom Typ „$\oplus$“, „$\wedge$“, „$\neg$“ oder „$\vee$“.

Mit dieser Begriffsbildung kann man auch (in etwa) erklären, was man darunter verstehen will, dass eine allgemeine Funktion

$$f : A^* \to A^*$$

„effizient“ berechnet werden kann. Zu jedem $n \in \mathbb{N}$ definieren wir

$$C_f(n) := \text{minimale Komplexität eines Schaltkreises, der } f^{(n)} \text{ berechnet .}$$

f soll nun *effizient berechenbar* heissen, wenn die Komplexitätsfunktion $C_f : \mathbb{N} \to \mathbb{N}$ „nicht allzu schnell wächst“. Dieser letztere Begriff wäre auch wieder zu definieren. Üblicherweise versteht man darunter, dass ein Polynom $p(t)$ existiert mit der Eigenschaft

$$C_f(n) \ \leq \ p(n) \quad \text{für alle } n \in \mathbb{N} \ .$$

4. Reversible Berechnungen

Typischerweise ist eine Boolesche Funktion $f : A^n \to A^m$ nicht injektiv. Aus dem Funktionswert $y = f(x)$ kann dann nicht sicher auf die Eingabe x geschlossen werden. Deshalb ist auch der Rechenprozess über einen Booleschen Schaltkreis nicht *reversibel*.

Es ist jedoch leicht, das Berechnungsproblem bzgl. f so zu formulieren, dass Reversibilität garantiert ist. Dazu betrachten wir die Operation

$$\sigma_f : \ x \otimes z \ \to x \otimes (z \oplus f(x)) \quad (x \in \mathbb{A}^n, z \in A^m) \ .$$

In diese Operation σ_f ist f einkodiert: Bei der Eingabe $z = 0^m$ können wir $f(x)$ aus den letzten m Komponenten ablesen:

$$\sigma_f(x \otimes 0^m) = x \otimes f(x) \quad (0^m := 00\ldots0 \in A^m) \ .$$

LEMMA 1.1. *$\sigma_f : A^{n+m} \to A^{n+m}$ ist eine Permutation, d.h. σ_f ist injektiv (und folglich auch surjektiv).*

Bew. Sei $x \otimes (z \oplus f(x)) = x' \otimes (z' \oplus f(x'))$. Dann gilt $x = x'$ und folglich $f(x') = f(x)$. Wegen $f(x) \oplus f(x) = 0^m$ und $z' \oplus f(x) = z \oplus f(x)$ finden wir deshalb

$$x \otimes z' = x \otimes (z' \oplus f(x) \oplus f(x)) = x \otimes z \quad \text{d.h.} \quad z = z' \ .$$

Damit erkennen wir σ_f als injektiv. Eine injektive Abbildung einer endlichen Menge in sich ist aber notwendigerweise auch surjektiv und somit bijektiv.

Eine Permutation einer Grundmenge kann man als eine Vertauschung der „Namen“ der Elemente interpretieren. In diesem Sinn bewirkt σ_f einfach eine „Umkodierung“ der Wörter der Länge $n+m$. Die Berechnung von $f(x)$ reduziert sich dann auf folgenden Beobachtungvorgang:

- Lies die letzten m Buchstaben des neuen Namens $\sigma_f(x \otimes 0^m)$ des Objektes mit dem ursprünglichen Namen $x \otimes 0^m$ ab.

4.1. Reversible Boolesche Schaltkreise. Nachdem wir erkannt haben, dass eine Boolesche Funktion (im obigen Sinne) über eine Permutation (d.h über eine reversible Boolesche Funktion) dargestellt werden kann, liegt es nahe, auch entsprechende Boolesche Schaltkreise aus reversiblen Gattern zu konstruieren.

Die Negation $\neg : A \to A$ ist reversibel. Die Operation $\wedge : A^2 \to A$ jedoch nicht. Wir können mit diesen Bauelementen noch nicht direkt einen reversiblen Booleschen Schaltkreis erwarten. Wir konstruieren deshalb aus diesen Bausteinen ein neues Gatter T[1] für eine reversible Boolesche Funktion (Permutation) auf A^3:

$$T: \quad (x_1, x_2, x_3) \to (x_1, x_2, x_3 \oplus (x_1 \wedge x_2))$$

d.h. schematisch

$$\begin{array}{ccccc} x_1 & \to & \boxed{} & \to & x_1 \\ x_2 & \to & \boxed{T} & \to & x_2 \\ x_3 & \to & \boxed{} & \to & x_3 \oplus (x_1 \wedge x_2) \end{array}$$

Nehmen wir hier x_1, x_2 oder x_3 als Kontrollbits, so verwirklicht das Gatter T *vier* Funktionen:

1. *logisches UND* (von x_1 und x_2): setze $x_3 = 0$;
2. *Addition* (mod 2) (von x_1 und x_3): setze $x_2 = 1$;
3. *Negation* (von x_3): setze $x_1 = 1$ und $x_2 = 1$;
4. *Kopieren* (von x_1): setze $x_2 = 1$ und $x_3 = 0$.

Nun ist es nicht schwer einzusehen, dass eine Boolesche Funktion $f : A^n \to A^m$ auch über einen sog. *reversiblen* Booleschen Schaltkreis berechnet werden kann, d.h. über einen Booleschen Schaltkreis, der nur reversible Gatter verwendet:

- Wir wissen, dass zu f ein Schaltkreis besteht, der nur Gattertypen mit den Funktionen 1.-4. verwendet. In diesem Schaltkreis ersetzen wir jedes Gatter durch ein reversibles T-Gatter und stellen durch das Setzen der Kontrollbits dessen Funktion sicher.

Bei dem Ersetzungsprozess benötigen wir für jedes der neuen Gatter allerdings noch möglicherweise Kontrollbits, sodass sich insgesamt die Anzahl der Eingabeknoten (und entsprechend der Ausgabeknoten) erhöht. Jedoch wächst dabei insgesamt die Anzahl der Knoten höchstens auf das 6-fache der ursprünglichen Anzahl, während die Anzahl der echten Rechenknoten konstant bleibt.

[1] T ist als *Toffoli-Funktion* bekannt, wurde aber tatsächlich schon vorher von Petri verwendet (und publiziert)

Jeder Knoten T weist „intern“ konstante Komplexität auf: Der Baustein T kann ja mit einer festen Zahl von Booleschen Schaltelementen herkömmlicher Art realisiert werden. Also wird insgesamt die Komplexität des reversiblen Schaltkreises höchstens um einen konstanten Faktor (der von n unabhängig ist!) anwachsen. Damit folgt

THEOREM 1.1. *Jede effizient berechenbare Boolesche Funktion $f : A^* \to A^*$ lässt sich auch mit reversiblen Booleschen Schaltkreisen effizient berechnen.*

◇

4.2. Weitere Bemerkungen zu reversiblen Booleschen Schaltkreisen. Es ist klar, dass ein reversibler Boolescher Schaltkreis tatsächlich eine Permutation realisiert: Jeder Knoten, der einem reversiblen Gatter entspricht, hat die Eigenschaft

$$\text{Eingangsgrad} = \text{Ausgangsgrad} .$$

Also ist auch die Anzahl der Eingabeknoten gleich der Anzahl der Ausgabeknoten, sagen wir: k. Da die Berechnung an jedem Gatter reversibel ist, ist sie insgesamt reversibel. Der reversible Schaltkreis für $f : A^n \to A^m$ realisiert also eine injektive Funktion

$$\bar{\sigma}_f : A^k \to A^k .$$

Die Funktion f ist in $\bar{\sigma}_f$ eingebettet. Der Funktionswert $f(x)$ kann somit bei Eingabe von $x \otimes 0^{k-n}$ an m geeigneten Ausgabeknoten abgelesen werden. (Die Bitwerte an den übrigen Ausgabeknoten sind für die Berechnung von f dann uninteressant.)

KAPITEL 2

Zustände, Stobits und Qubits

Um das Modell des reversiblen Booleschen Rechners zu verallgemeinern, betrachten wir unsere Rechenmaschine von einem systemtheoretischen Standpunkt aus:

$$x \longrightarrow \boxed{\boxed{X} \rightarrow \boxed{\varphi} \rightarrow \boxed{Y}} \longrightarrow y\ .$$

Den Rechner stellen wir uns also als ein System $\mathcal{S}$ mit folgender Funktionsweise vor:

(1) Ein Wort $x \in A^n$ wird über die Eingabeeinheit X eingegeben;
(2) Das System $\mathcal{S}$ geht dann in einen „Zustand“ $\varphi(x)$ über;
(3) Die Ausgabeeinheit Y nimmt eine „Messung“ am System $\mathcal{S}$ im Zustand $\varphi(x)$ vor und gibt ein Wort $y \in A^n$ aus.

Im Fall eines Booleschen Schaltkreises, der eine Permutation $\sigma : A^n \to A^n$ (deterministisch) realisiert, haben wir einfach

$$\varphi(x) = \sigma(x) = y\ .$$

Im nichtdeterministischen Fall wollen wir an dieser Stelle die genaue „Natur“ (bzw. Definition) der Zustandsfunktion φ nicht spezifizieren, sondern einfach durch ihre Auswirkung auf die Messung Y festhalten.

Die Auswirkung beschreiben wir durch die Wahrscheinlichkeit p_k, mit denen ein bestimmtes n-Bit-Wort k ausgegeben wird:

$$p_k \ := \ P(Y = k \mid \varphi(x)) \quad (k \in A^n)$$

Diese Wahrscheinlichkeiten fassen wir zu einem Vektor

$$\mathbf{p} = (p_0, \ldots, p_{N-1}) \ \in \ \mathbb{R}^N \quad (\text{mit } N = 2^n)$$

zusammen und haben dann

$$\varphi \quad \longleftrightarrow \quad \mathbf{p} = (p_0, \ldots, p_{N-1})\ .$$

Die Messeinheit Y ist also eine *Zufallsvariable* mit der *Verteilung* $\mathbf{p}$.

BEMERKUNG (WAHRSCHEINLICHKEITSVERTEILUNG.) Unter einer *Wahrscheinlichkeitsverteilung* auf der Menge

$$\{0, 1, \ldots, N-1\}$$

versteht man generell einen Vektor $\mathbf{p} = (p_0, \ldots, p_{N-1}) \in \mathbb{R}^N$ mit der Eigenschaft

$$\sum_{k=0}^{N-1} p_k = 1 \quad \text{und} \quad p_k \geq 0\ .$$

Ein nach den Prinzipien (1)-(3) funktionierendes System $\mathcal{S}$ heisst *stochastisches n-Bit-System*. Insbesondere sind dann reversible Boolesche Schaltkreise „triviale“ stochastische n-Bitsysteme, deren Zustände $\varphi(x)$ durch die entsprechenden Einheitsvektoren charakerisiert sind:

$$\varphi(x) = k \quad \longleftrightarrow \quad \mathbf{p} = (0, 0, \ldots, 0, 1, 0, \ldots, 0) = \mathbf{e}_k \ .$$

1. Stobits

Wir wollen nun eine Zufallsvariable Y mit Wertebereich A^n und (Wahrscheinlichkeits-)Verteilung

$$\mathbf{p} = (p_k)_{k \in A^n}$$

ein *stochastisches n-Bit* (oder kurz: *n-Stobit*[1]) nennen. Ein Element $k \in A^n$ heisst dann auch *reines* (oder *Boolesches*) *n-Bit.* Ihm entspricht als Wahrscheinlichkeitsverteilung der Einheitsvektor $\mathbf{e}_k$, der genau an der Stelle k den Koeffizienten 1 aufweist:

$$k \in A^n \quad \longleftrightarrow \quad \mathbf{e}_k = (0, \ldots, 0, 1, 0, \ldots, 0) \ \in \ \mathbb{R}^N \ .$$

Mit diesem Veständnis können wir das n-Stobit $\mathbf{p}$ in Summenform notieren:

$$\mathbf{p} = \sum_{k=0}^{N-1} p_k \mathbf{e}_k \quad \text{bzw.} \quad \mathbf{p} = \sum_{k \in A^n} p_k |k) \ .$$

NOTATION: Es wurde schon im ersten Kapitel bemerkt, dass ein n-Bit $k \in A^n$ auch als Binärdarstellung einer natürlichen Zahl $0 \leq k \leq 2^n - 1$ aufgefasst werden kann. Um diese Interpretationsnuancen etwas augenfälliger zu halten, benutzen wir auch die Notation

$$\text{„}|k)\text{“} \qquad (\ \longleftrightarrow \ \mathbf{e}_k \ ) \ ,$$

wenn wir $k \in A^n$ als reines n-Bit bzw. als den entsprechenden Einheitsvektor $\mathbf{e}_k \in \mathbb{R}^N$ betonen wollen.

Die Zufallsvariable (n-Stobit) Y induziert weitere Zufallsvariablen. Zum Beispiel sind die Variablen

$$Y_j := j\text{tes Bit } (j\text{te Stelle}) \text{ von } Y \quad (j = 1, \ldots, n)$$

aus Y abgeleitete 1-Stobits, wie sie bei Einzelbitmessungen an stochastischen n-Bitsystemen auftreten.

Ein 1-Stobit (oder kurz: *Stobit*) hat die allgemeine Form

$$\mathbf{p} = p_0|0) + p_1|1) \quad \text{mit} \quad p_0 + p_1 = 1, \ p_0, p_1 \geq 0 \ .$$

Ein 2-Stobit hätte die Form

$$\mathbf{p} = p_{00}|00) + p_{01}|01) + p_{10}|10) + p_{11}|11) \quad \text{mit} \quad \sum_{k \in \{0,1\}^2} p_k = 1, \ p_k \geq 0 \ .$$

[1] diese Bezeichnung hat sich international noch nicht durchgesetzt...

2. Bedingte Beobachtungen

Wenden wir uns nun den aus Y abgeleiteten Einzelbitmessungen zu und definieren (wie im vorigen Abschnitt)

$$Y_j := j\text{tes Bit in der Bitdarstellung von } Y \quad (j = 1, \ldots, n) \; .$$

Y_j hat Werte in $\{0, 1\}$) und ist somit ein Stobit. Ist Y ein 2-Stobit mit Verteilung $\mathbf{p}$, haben wir etwa

$$\begin{array}{rclcrcl} p_0^{(1)} = P(Y_1 = 0) & = & p_{00} + p_{01} & , & p_1^{(1)} = P(Y_1 = 1) & = & p_{10} + p_{11} \\ p_0^{(2)} = P(Y_2 = 0) & = & p_{00} + p_{10} & , & p_1^{(2)} = P(Y_2 = 1) & = & p_{01} + p_{11} \end{array}$$

Die induzierten Messvariablen Y_1 und Y_2 entsprechen also den beiden Stobits

$$\mathbf{p}^{(1)} = p_0^{(1)}|0) + p_1^{(1)}|)1 \quad \text{bzw.} \quad \mathbf{p}^{(2)} = p_0^{(2)}|0) + p_1^{(2)}|1) \; .$$

BEMERKUNG. In der Terminologie der Wahrscheinlichkeitsrechnung sind $\mathbf{p}^{(1)}$ und $\mathbf{p}^{(2)}$ die beiden *Randverteilungen* der Wahrscheinlichkeitsverteilung $\mathbf{p}$.

Wir können aus $\mathbf{p}$ auch die *bedingten Wahrscheinlichkeiten* ableiten. Z.B.:

$$P(Y_1 = h \mid Y_2 = \ell) := \frac{P(Y = h\ell)}{P(Y_2 = \ell)} = \frac{p_{h\ell}}{p_\ell^{(2)}}$$

Man nennt Y_1 und Y_2 *(stochastisch) unabhängig*, wenn Wissen über Y_2 keine Information bzgl. Y_1 liefert, d.h. wenn für alle beliebigen Bits h und ℓ gilt:

$$\boxed{P(Y_1 = h \mid Y_2 = \ell) = P(Y_1 = h) \quad \text{d.h.} \quad p_{h\ell} = p_h^{(1)} \cdot p_\ell^{(2)}}$$

Man macht sich leicht klar, dass die Einzelbitbeobachtungen Y_1 und Y_2 eines reinen (Booleschen) 2-Bits

$$h\ell = h \otimes \ell \quad (h, \ell \in \{0, 1\})$$

immer stochastisch unabhängig sind. Im Fall des 2-Stobits

$$\mathbf{p} = \frac{1}{2}|00) + \frac{1}{2}|11) \;\; [\, + 0|01) + 0|10) \,]$$

finden wir jedoch z.B.

$$P(Y_1 = 0 \mid Y_2 = 1) = 0 \; \neq \; \frac{1}{2} = P(Y_1 = 0) \; .$$

Die Einzelbitbeobachtungen sind also nicht voneinander unabhängig zu machen.

2.1. Verkettungen von Stobits. Motiviert von der Idee der stochastischen Unabhängigkeit können wir nun den Begriff der Verkettung von Bits auf Stobits verallgemeinern.

Gegeben die Stobits $\mathbf{p} = p_0|0) + p_1|1)$ und $\mathbf{q} = q_0|0) + q_1|1)$, setzen wir

$$\mathbf{p} \otimes \mathbf{q} \ := \ p_0q_0|00) + p_0q_1|01) + p_1q_0|10) + p_1q_1|11) \ .$$

Bzgl. der Verkettung $\mathbf{p} \otimes \mathbf{q}$ induzieren die Einzelbitmessungen Y_1 und Y_2 die Stobits $\mathbf{p}$ und $\mathbf{q}$ und sind somit (*per definitionem*!) stochastisch unabhängig. Wie wir schon oben gesehen haben, ist umgekehrt jedes 2-Stobit mit unabhängigen Einzelbitbeobachtungen die Verkettung der entsprechenden 1-Stobits.

2.2. Allgemeine Verkettung. Das bisher Festgestellte gilt sofort in der erwarteten Allgemeinheit. Gegeben ein $(m+n)$-Stobit

$$\mathbf{p} = \sum_{k \in \{0,1\}^{m+n}} p_k|k) \ ,$$

betrachten wir die Messvariablen

$$\begin{array}{rcl} Y^{(m)} & := & \text{die ersten } m \text{ Bits von } Y \\ Z^{(n)} & := & \text{die letzten } n \text{ Bits von } Y \end{array} \qquad (\text{d.h.} \quad Y = Y^{(m)} \otimes Z^{(n)}) \ ,$$

welche das m-Stobit $\mathbf{p}^{(m)}$ bzw. das n-Stobit $\mathbf{q}^{(n)}$ induzieren mögen. Dann findet man bzgl. der bedingten Wahrscheinlichkeiten völlig anolog wie zuvor:

$$P(Y^{(m)} = u | Z^{(n)} = v) = P(Y^{(m)} = u) \quad \text{für alle } u \in A^m, v \in A^n$$

ist gleichwertig mit der Bedingung

$$\sum_{k \in A^{m+n}} p_k|k) \ = \ \sum_{u \in A^m} \sum_{v \in A^n} p_u^{(m)} q_v^{(n)} |u \otimes v) \ =: \ \mathbf{p}^{(m)} \otimes \mathbf{q}^{(n)} \ .$$

Die Verkettung ist also für allgemeine Stobits wohldefiniert und reflektiert die Eigenschaft, dass die entsprechenden Teilmengen von Bits bzgl. Y stochastisch unabhängig voneinander gemessen werden können.

BEMERKUNG. Routinemässig rechnet sofort, wie erwartet, die Assoziativitätseigenschaft von Stobitverkettungen nach:

$$(\mathbf{u} \otimes \mathbf{v}) \otimes \mathbf{w} = \mathbf{u} \otimes (\mathbf{v} \otimes \mathbf{w}) \ .$$

3. Permutationen

Der wesentliche Aspekt des Berechnens Boolescher Funktionen f über reversible Boolesche Schaltkreise mit n Eingabeknoten war die Reduktion auf die Aufgabe, geeignete Einzelbitmessungen nach einer Permutation σ_f der Menge A^n aller n-Bits an den Ausgabeknoten auszuführen.

Im Prinzip lässt sich diese „Rechentechnik“ auch auf n-Stobits übertragen. Ist nämlich $\sigma : A^n \to A^n$ eine beliebige Permutation, dann gilt offenbar

$$\mathbf{p} = \sum_{k\in A^n} p_k|k) \text{ ist } n\text{-Stobit} \quad \Longleftrightarrow \quad \mathbf{p}^\sigma = \sum_{k\in A^n} p_k|\sigma(k)\,) \text{ ist } n\text{-Stobit}$$

Um des Spektrum potentieller „stochastischer reversibler Schaltkreise“noch zu erweitern, interessieren wir uns für weitere Transformationen, die n-Stobits in n-Stobits überführen. Diese Transformationen ergeben sich sehr natürlich, wenn wir einen leicht subtileren Blick auf n-Stobits werfen (s. nächster Abschnitt).

4. Qubits

Der springende Punkt ist die Beobachtung, dass das Quadrat einer reellen Zahl nichtnegativ ist. Um Wahrscheinlichkeitsverteilungen zu konstruieren, können wir also folgendermassen vorgehen:

Wir nehmen einen beliebigen Vektor $\mathbf{v} = (v_0, \ldots, v_{N-1}) \in \mathbb{R}^N$. Ist $\mathbf{v} \neq \mathbf{0}$, so berechnen wir die (quadrierte) Euklidische Norm

$$\|\mathbf{v}\|^2 := \sum_{k=0}^{N-1} v_k^2 \quad \text{und} \quad p_k := \frac{v_k^2}{\|\mathbf{v}\|^2} \quad (k = 0, 1, \ldots, N-1)\ .$$

Damit erhalten wir die von $\mathbf{v}$ induzierte Wahrscheinlichkeitsverteilung

$$\mathbf{p} = (p_0, p_1, \ldots, p_{N-1})\ .$$

In diesem abstrakten Formalismus ist ein *n-Qubit* („n-Quantenbit“) einfach ein Vektor $\mathbf{v} \in \mathbb{R}^N$ mit der Eigenschaft

$$\|\mathbf{v}\|^2 = \sum_{k=0}^{N-1} v_k^2 = 1\ .$$

Wie bei n-Stobits können wir ein n-Qubit formal direkt an die Booleschen n-Bits anbinden mit der Darstellung

$$\mathbf{v} = \sum_{k\in A^n} v_k|k\rangle\ .$$

BEMERKUNG (DIRAC-NOTATION). Man beachte, dass bei dem n-Qubit $\mathbf{v}$ die Messwahrscheinlichkeit des n-Bits $k \in A^n$ durch v_k^2 (und nicht durch v_k) modelliert ist! Der (unquadrierte) Koeffizient v_k heisst auch *Amplitude* des n-Bits k. (Die Begriffsbildung ist aus der Fourieranalyse motiviert, auf die später noch eingegegangen wird.) Wir weisen auf diesen Unterschied optisch hin, indem wir die sog. *Dirac-Notation* zur Darstellung der reinen n-Bits benutzen:

$$\text{„}|k\rangle\text{“} \quad (\longleftrightarrow\ \mathbf{e}_k\ \in\ \mathbb{R}^N\ )$$

Philosophie. *Wenn wir sagen, dass ein Quantenbit ein gewisser Koordinatenvektor $\mathbf{v} \in \mathbb{R}^N$ „*ist*“, so ist dies nicht exakt. Tatsächlich spezifizieren wir* nicht*, was ein Quantenbit „eigentlich “ sei, sondern gehen in unserem Rechenmodell einfach*

davon aus, dass die wesentlichen Eigenschaften von Quantenbits durch Koordinatenvektoren hinreichend beschrieben werden können.

Dies hat zur Folge, dass Quantenbits ebensogut auch durch andere Koordinatenvektoren beschrieben werden können, die sich aus etwaigen Basiswechseln ergeben. Tatsächlich besteht das Quantenrechnen im Wesen in einer geeigneten Folge von Basis- und Beschreibungswechseln des Eingabe-Quantenbits.

Dieselbe Sitation besteht schon bei den Booleschen Bits „0“ und „1“, die keine Bedeutung an sich *sondern nur beschreibende Funktion haben.* (Ein Vertauschen der Symbole hätte ausser einer geänderten Notation keinerlei praktische Konsequenzen.)

Quantenbitsysteme. In Verallgemeinerung der stochastischen n-Bitsysteme nennen wir nun das zu Anfang dieses Kapitels betrachtete Rechensystem $\mathcal{S}$ ein *n-Bit-Quantensystem*, wenn wir uns die Zustandsfunktion φ als eine Grösse vorstellen, welche ein n-Quantenbit Y induziert.

NOTA BENE: *Welches* Quantenbit durch Y gemessen wird, können wir nicht genau feststellen. Nach wie vor werden wir nur mit den *Wahrscheinlichkeiten* konfrontiert, mit denen ein bestimmtes Wort ausgegeben wird:

$$p_k \;=\; P(Y = |k\rangle \mid \varphi(x)\,) \;=\; |v_k|^2\,,$$

falls Y das n-Quantenbit $\mathbf{v} = (v_0, \ldots, v_k, \ldots, v_{N-1})$ misst. Verschiedene Quantenbits können durchaus dasselbe Stobit induzieren. Zum Beispiel ergeben

$$\mathbf{v} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \quad \text{und} \quad \mathbf{w} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

dasselbe Stobit $\mathbf{p} = \frac{1}{2}|0) + \frac{1}{2}|1)$.

4.1. Verkettung und Verschränkung von Qubits. Die Operation der Verkettung lässt sich sofort auf Qubits erweitern. Wir setzen

$$\mathbf{v} \otimes \mathbf{w} := \sum_{k \in \mathbb{A}^m} \sum_{\ell \in A^n} v_k w_\ell |k \otimes \ell\rangle \quad \text{für } \mathbf{v} = \sum_{k \in A^m} v_k |k\rangle \text{ und } \mathbf{w} = \sum_{\ell \in A^n} w_\ell |\ell\rangle$$

und finden, dass $\mathbf{v} \otimes \mathbf{w}$ ein $(m+n)$-Qubit ist, wenn $\mathbf{v}$ und $\mathbf{w}$ m- bzw. n-Qubits waren. Es gilt nämlich

$$\|\mathbf{v} \otimes \mathbf{w}\|^2 = \sum_{k \in A^m} \sum_{\ell \in A^n} |v_k w_\ell|^2 = \sum_{k \in A^m} |v_k|^2 \sum_{\ell \in A^n} |w_\ell|^2 = \|\mathbf{v}\|^2 \cdot \|\mathbf{w}\|^2\,.$$

Diese Beobachtung zeigt weiterhin, dass die Verkettung der Qubits die Verkettung der entsprechenden Stobits induziert und die Bestandteile $\mathbf{v}$ und $\mathbf{w}$ des $(m+n)$-Qubits $\mathbf{v} \otimes \mathbf{w}$ folglich von einander (stochastisch) unabhängig gemessen werden können. Denn für die Beobachtungswahrscheinlichkeiten gilt ja:

$$|v_k w_\ell|^2 = |v_k|^2 \cdot |w_\ell|^2\,.$$

Man nennt ein n-Qubit $\mathbf{v}$ *verschränkt*, wenn es *nicht* möglich ist, $\mathbf{v}$ als Verkettung von zwei anderen Qubits auszudrücken.

NOTA BENE: Bei einem n-Stobit bedeutet Verschränktheit, dass keine stochastisch unabhängigen komplementären Teilmessungen möglich sind. Es ist allerdings möglich, dass ein verschränktes(!) n-Qubit ein verkettetes(!) n-Stobit impliziert. Z.B. ist das folgende 2-Qubit

$$\mathbf{v} = \tfrac{1}{2}|00\rangle - \tfrac{1}{2}|01\rangle + \tfrac{1}{2}|10\rangle + \tfrac{1}{2}|11\rangle$$

verschränkt und impliziert das verkettete 2-Stobit

$$\mathbf{p} = \tfrac{1}{4}|00) + \tfrac{1}{4}|01) + \tfrac{1}{4}|10) + \tfrac{1}{4}|11) = [\tfrac{1}{2}|0) + \tfrac{1}{2}|1)] \otimes [\tfrac{1}{2}|0) + \tfrac{1}{2}|1)] \; .$$

4.2. Komplexe Qubits. Alles bisher über Qubits mit reellen Koeffizienten gesagte bleibt richtig, wenn wir als Skalarbereich die Menge $\mathbb{C}$ der komplexen Zahlen zulassen. Dazu erinnern wir an die komplexen Zahlen als die Menge aller Ausdrücke der Form

$$z = a + \mathrm{i}b \quad (a, b \in \mathbb{R}) \; .$$

wobei „i“ ein spezielles Symbol ist. Mit den komplexen Zahlen rechnet man wie mit den reellen Zahlen unter Beachtung der zusätzlichen Regel

$$\mathrm{i}^2 = -1 \; .$$

BEMERKUNG. Tatsächlich tritt die „imaginäre Zahl“ i in der praktischen Rechnung *nie*(!) auf. Man führt die numerischen Operationen immer nur mit den reellen Komponenten a und b der komplexen Zahlen $z = a + \mathrm{i}b$ aus. „i“ist dann nur ein „Indikator“ dafür, um welchen (reellen) Teil der komplexen Zahl es sich handelt.

Die zu $z = a + \mathrm{i}b$ *konjugierte* komplexe Zahl ist definiert als $\overline{z} := a - \mathrm{i}b$. Damit erhält man den Betrag als

$$|z| = |a + \mathrm{i}b| := \sqrt{a^2 + b^2} = \sqrt{z\overline{z}} \; .$$

Für die Anwendungen nützliche Interpretationen der komplexen Zahlen ergeben sich aus der *Eulerschen Formel*:

$$\boxed{e^{\mathrm{i}t} \;=\; \sum_{n=0}^{\infty} \frac{(\mathrm{i}t)^n}{n!} \;=\; \cos t + \mathrm{i}\sin t \quad (\; t \in \mathbb{R} \;)}$$

Insbesondere finden wir

$$\overline{e^{\mathrm{i}t}} = e^{-\mathrm{i}t} \quad \text{für alle } t \in \mathbb{R} \quad \text{und} \quad e^{2m\pi\mathrm{i}} = 1 \quad \text{für alle } m \in \mathbb{Z} \; .$$

Wir verzichten fürderhin bei Qubits auf das Wort „komplex“ und verstehen allgemein unter einem *n-Qubit* einen Ausdruck der Form

$$\mathbf{v} = \sum_{k \in A^n} v_k |k\rangle \quad \text{mit} \quad \|\mathbf{v}\|^2 = \sum_k |v_k|^2 = \sum_k v_k \overline{v}_k = 1 \quad (\; v_k \in \mathbb{C} \;) \; .$$

Aus der Sicht der linearen Algebra ist ein „n-Qubit“ im wesentlichen ein Vektor $\mathbf{v}$ der Länge 1 im Vektorraum $\mathbb{C}^N$ der Dimension $N = 2^n$.

KAPITEL 3

Transformationen auf Qubits und Quantenschaltkreise

Eine Permutation $\sigma : A^n \to A^n$ der reinen n-Bits führt auch ein n-Quantenbit in ein n-Quantenbit über:

$$\mathbf{v} = \sum_{k \in A^n} v_k |k\rangle \quad \rightsquigarrow \quad \mathbf{v}^\sigma = \sum_{k \in A^n} v_k |\sigma(k)\rangle \ .$$

Wir wollen noch weitere Transformationen in Erwägung ziehen, die n-Qubits erhalten. Die Hoffnung ist, solche Transformationen technisch realsieren zu können, um „Quantenrechner“ zu konstruieren, die nach einem ähnlichen Prinzip funktionieren wie (reversible) Boolesche Schaltkreise:

- Realisiere eine geeignete Transformation φ und lies das Rechenergebnis bzgl. der Eingabe eines n-Bits x über Y aus dem gemäss $\varphi(x)$ transformierten n-Quantenbit ab.

Eine naheliegende Klasse von Transformationen (welche insbesondere die Permutationen enthalten) sind die längenerhaltenen linearen Transformationen auf $\mathbb{C}^N$:

$$U : \mathbb{C}^N \to \mathbb{C}^N \quad \text{derart, dass} \quad \|U\mathbf{v}\|^2 = \|\mathbf{v}\|^2 \ .$$

Dieses sind die sog. *unitären Transformationen*, die durch Matrizen $U \in \mathbb{C}^{N \times N}$ beschrieben werden mit der Eigenschaft

$$\overline{U}^T U = I \quad \text{bzw.} \quad U^{-1} = \overline{U}^T \ ,$$

wobei $\overline{U}$ die zu U konjugiert-komplexe Matrix bedeute. In der Tat beobachtet man sofort:

$$\|U\mathbf{v}\|^2 = (\overline{U\mathbf{v}})^T (U\mathbf{v}) = \overline{\mathbf{v}}^T (\overline{U}^T U)\mathbf{v} = \overline{\mathbf{v}}^T \mathbf{v} = \|\mathbf{v}\|^2 \ .$$

Sind die Koeffizienten u_{ij} der unitären Matrix U alles reelle Zahlen (d.h. $u_{ij} = \overline{u}_i j$), dann gilt $\overline{U} = U$. Mit anderen Worten: Die reellen unitären Matrizen sind die sog. *orthogonalen Matrizen* $O \in \mathbb{R}^{N \times N}$ mit der Eigenschaft

$$O^T O = I \quad \text{bzw.} \quad O^{-1} = O^T \ .$$

Wie bei Booleschen Schaltkreisen wollen wir den angestrebten *Quantenschaltkreis* aus möglichst einfachen Bauelementen zusammensetzen. Wir interessieren uns deshalb besonders für unitäre Transformationen auf 1-Qubits.

1. Lokale unitäre Transformationen.

Im Fall $n = 1$ (d.h. $N = 2^n = 2$) wirkt eine Matrix $U = (u_{ij}) \in \mathbb{C}^{2\times 2}$ auf die reinen Bits $0, 1 \in A$ (d.h. auf die entsprechenden Einheitsvektoren in $\mathbb{C}^2$) folgendermassen:

$$U|0\rangle \quad \longleftrightarrow \quad \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} u_{11} \\ u_{21} \end{pmatrix} \quad \longleftrightarrow \quad u_{11}|0\rangle + u_{21}|1\rangle$$

und

$$U|1\rangle \quad \longleftrightarrow \quad \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} u_{12} \\ u_{22} \end{pmatrix} \quad \longleftrightarrow \quad u_{12}|0\rangle + u_{22}|1\rangle$$

Die Wirkung von U auf ein allgemeines 1-Qubit ergibt sich daraus sofort als entsprechend gewichtete Überlagerung (d.h. Linearkombination):

$$U(v_0|0\rangle + v_1|1\rangle) \longleftrightarrow \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \end{pmatrix} \longleftrightarrow v_0U|0\rangle + v_1U|1\rangle \ .$$

Die folgenden Beispiele unitärer Transformationen sind bei späteren Algorithmen besonders wichtig.

Drehspiegelungen in $\mathbb{R}^2$. Die orthogonale Matrix

$$S = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \in \mathbb{R}^{2\times 2}$$

vertauscht die Einheitsvektoren $(1, 0)$ und $(0, 1)$ in $\mathbb{C}^2$, bewirkt also einfach eine Koordinatenpermutation auf der Menge alle 1-Qubits, wie wir sie schon zur Konstruktion einer Rechenmaschine für die Boolesche Negation $\neg$ diskutiert hatten:

$$S|0\rangle = |1\rangle \quad \text{und} \quad S|1\rangle = |0\rangle \ .$$

BEMERKUNG. Geometrisch repräsentiert S die Spiegelung der Punkte in $\mathbb{R}^2$ an der ersten Winkelhalbierenden.

Zu jeder reellen Zahl $t \in \mathbb{R}$ beschreibt die orthogonale Matrix

$$D_t = \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix} \in \mathbb{R}^{2\times 2}$$

eine Drehung der Ebene $\mathbb{R}^2$ um einen (im Bogenmass t gemessenen) Winkel φ. Der Spezialfall einer Drehung um den Winkel $\varphi = 45^o$ (d.h. $t = \pi/4$) ergibt in Kombination mit der Spiegelungsmatrix S die sog. *Hadamardmatrix*

$$H \ := \ SD_{\pi/4} \ = \ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \cos \pi/4 & -\sin \pi/4 \\ \sin \pi/4 & \cos \pi/4 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \ .$$

mit der praktischen Eigenschaft

$$H = H^T \quad \text{d.h.} \quad H^2 = I \ .$$

Wir finden

$$H|0\rangle = \tfrac{1}{\sqrt{2}}|0\rangle + \tfrac{1}{\sqrt{2}}|1\rangle \quad \text{und} \quad H|1\rangle = \tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle \ .$$

BEMERKUNG („ENTWICKLUNG EINES QUBITSYSTEMS“.) Die Anwendung einer orthogonalen Transformation auf ein Qubit bezeichnet man auch als *Entwicklung* des Qubits. Die Transformation D_t motiviert etwas diese Bezeichung, wenn man sich vorstellt, dass t einen „Zeit“-Parameter darstellt. Wir kommen darauf im Zusammenhang mit der Schrödingerschen Wellengleichung nochmals zurück.

Phasenverschiebungen. Eine weitere elementare Transformation auf 1-Qubits wird durch (Diagonal-)Matrizen $W_{t_0} \in \mathbb{C}^{2\times 2}$ vom Typ

$$W_{t_0} = \begin{pmatrix} 1 & 0 \\ 0 & e^{\mathrm{i}t_0} \end{pmatrix} \quad (\ t_0 \in \mathbb{R}\ )$$

beschrieben, die folgende Wirkung haben:

$$W_{t_0}|0\rangle = |0\rangle \quad \text{und} \quad W_{t_0}|1\rangle = e^{\mathrm{i}t_0}|1\rangle \ .$$

Die „anschauliche Wirkung“ von W_{t_0} erkärt sich aus der Eulerschen Formel. Sei z.B.

$$\mathbf{v} = v_0|0\rangle + v_1|1\rangle \quad \text{mit} \quad v_1 = |v_1|e^{\mathrm{i}t} = |v_1|(\cos t + \mathrm{i}\sin t) \quad (\ v_0, v_1 \in \mathbb{C}\ )$$

ein 1-Qubit. Dann lässt W_{t_0} die Amplitude v_0 von $|0\rangle$ unverändert. Bei $|1\rangle$ wird zwar der Betrag $|v_1|$ der Amplitude beibehalten, der „Phasenwinkel“ t verschiebt sich jedoch um t_0:

$$\begin{aligned} W_{t_0}\mathbf{v} &= v_0|0\rangle + |v_1|e^{\mathrm{i}(t+t_0)}|1\rangle \\ &= v_0|0\rangle + |v_1|(\cos(t+t_0) + \mathrm{i}\sin(t+t_0))\,|1\rangle \ . \end{aligned}$$

Oft benutzt man eine Phasenverschiebung mit $t_0 = \pi/4$, die auch wiederholt angewendet werden kann:

$$W_{\pi/4}\begin{pmatrix} 1 & 0 \\ 0 & e^{\mathrm{i}\pi/4} \end{pmatrix} \quad \Longrightarrow \quad W^2_{\pi/4} = W_{\pi/2}\begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix}\ ,\ W^4_{\pi/4} = W_\pi = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

1.1. Einzelbitmessungen. Es ist nützlich, sich den Effekt dieser Transformationen auf die (Beobachtungwahrscheinlichkeiten) der Einzelbitmessungen klar zu machen.

Die Spiegelung S vertauscht sich einfach die Rolle von „0“ und „1“ (Negation!):

$$S(v_0|0\rangle + v_1|1\rangle) = v_1|0\rangle + v_0|1\rangle \ .$$

Die Transformation der Hadamardmatrix H der reinen Bits 0 und 1 führt (in beiden Fällen!) zu dem 1-Stobit $\mathbf{p}$ der statistischen Gleichverteilung der Beobachtungsergebnisse:

$$\left.\begin{matrix} H|0\rangle \\ H|1\rangle \end{matrix}\right\} \quad \rightsquigarrow \quad \mathbf{p} = \tfrac{1}{2}|0) + \tfrac{1}{2}|1)$$

Eine weitere Anwendung der Hadamardtransformation stellt wegen $H^2 = I$ den (deterministischen!) Ausgangszustand wieder her:

$$H^2|0\rangle = |0\rangle \quad \text{und} \quad H^2|1\rangle = |1\rangle \ .$$

Nach der Phasenverschiebung W_{t_0} eines 1-Qubits $\mathbf{v}$ produziert man „0“ bzw. „1“ mit denselben Wahrscheinlichkeiten wie zuvor:

$$|v_0|^2 = |v_0|^2 \quad \text{und} \quad |v_1 e^{\mathrm{i}(t+t_0)}|^2 = |v_1|^2 \ .$$

1.2. Lokal implizierte Transformationen. Sei $\mathbf{v}$ ein n-Qubit und $U \in \mathbb{C}^{2^n \times 2^n}$ eine unitäre Transformation. Dann gilt

$$\mathbf{v} = \sum_{k \in A^n} v_k |k\rangle \quad \Longrightarrow \quad U\mathbf{v} = \sum_{k \in A^n} v_k U|k\rangle \ .$$

Es genügt also, die Wirkung einer Transformation auf die reinen n-Bits $k \in A^n$. *Jede* Abbildung

$$U : A^n \to \{\mathbf{z} \mid \mathbf{z}\ n\text{-Qubit}\} \quad (\ \text{wobei} \quad |k\rangle \to U|k\rangle\ )$$

in die Menge aller n-Qubits induziert eine längenerhaltende lineare Abbildung

$$U : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n} \ ,$$

wenn die Bildmenge $\tilde{U} = U(A^n)$ ein System von paarweise orthogonalen Qubits bilden. Denn diese Bedingung heisst ja gerade, dass wir eine unitäre Transformation definiert haben:

$$\tilde{U}^* \tilde{U} = I \ .$$

Seien z.B. $U_0, \ldots, U_{n-1} \in \mathbb{C}^{2\times 2}$ beliebige unitäre Transformation bzgl. 1-Qubits und

$$k = k_0 k_1 \ldots k_{n-1} = k_0 \otimes k_1 \otimes \ldots \otimes k_{n-1} \ \in \ A^n$$

ein reines n-Bit. Dann erhalten wir über

$$U|k\rangle \ := \ U_0|k_0\rangle \otimes U_1|k_1\rangle \otimes \ldots \otimes U_{n-1}|k_{n-1}\rangle$$

eine Verkettung von n 1-Qubits und damit ein n-Qubit. Die lokal angewendeten unitären Transformationen U_j implizieren eine unitäre Transformation U auf der Menge aller n-Qubits:

$$\boxed{U : \quad \mathbf{v} = \sum_{k \in A^n} v_k |k\rangle \ \to \ U_n \mathbf{v} = \sum_{k \in A^n} v_k U|k\rangle}$$

BEMERKUNG. Dass die Abbildung U tatsächlich eine unitäre Transformation (d.h. $U(A^n)$ ist ein Orthogonalystem) kann man ohne grosse Mühe direkt beweisen. Wir verzichten an dieser Stelle auf den Beweis, da er in der Diskussion des *Tensorproduktes* im nächsten Kapitel gegeben wird.

Die Hadamardtransformierte und Münzwürfe. Wir illustrieren das lokale Transformationsprinzip anhand der *Hadamardtransformation* H_n, wobei wir ausgehen von

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \in \mathbb{C}^{2\times 2} .$$

Ist $k = k_0 k_1 \dots k_{n-1} \in A^n$ ein reines n-Bit, so erhalten wir

$$\begin{aligned} H_n|k\rangle &:= (H|k_0\rangle) \otimes (H|k_1\rangle) \otimes \dots \otimes (H|k_{n-1}\rangle) \\ &= [\frac{1}{\sqrt{2}}|0\rangle + (-1)^{k_0}\frac{1}{\sqrt{2}}|1\rangle] \otimes \dots \otimes [\frac{1}{\sqrt{2}}|0\rangle + (-1)^{k_{n-1}}\frac{1}{\sqrt{2}}|1\rangle] \\ &= \frac{1}{2^{n/2}}\big[(|0\rangle + (-1)^{k_0}|1\rangle) \otimes \dots \otimes (|0\rangle + (-1)^{k_{n-1}}|1\rangle)\big] \end{aligned}$$

d.h.

$$\boxed{H_n|k\rangle \;=\; \frac{1}{2^{n/2}} \sum_{w\in A^n} (-1)^{k\cdot w}|w\rangle}$$

(Hier ist $k \cdot w$ die Anzahl der 1-Koeffizienten (bzw. Ziffern), die k und w (als Vektoren in $\{0,1\}^n$ betrachtet) gemeinsam haben.)

Mit einer Rechenmaschine, welche die Hadamardmatrix H realisiert, kann man perfekte statistische Gleichverteilungen erzeugen. Wir erhalten nämlich nach Eingabe eines (beliebigen) reinen n-Bits $k \in A^n$ die Ausgabewahrscheinlichkeiten

$$P(Y = w) = \frac{1}{2^{n/2}} \quad \text{für alle } \; w \in A^n.$$

Dies bedeutet: Jedwede Eingabe eines reinen (Booleschen) 1-Qubits $x \in \{0,1\}$ in die Maschine „H“ produziert ein 1-Qubit (Zustand) $H|x\rangle = v_0|0\rangle + v_1|1\rangle)$ mit den Messwahrscheinlichkeiten

$$P(Y=0) \;=\; |v_0|^2 \;=\; \frac{1}{2} \;=\; |v_1|^2 \;=\; P(Y=1) \; .$$

Insbesondere haben wir bei der Beobachtung des $j$ten Einzelbits von $H_n|k\rangle$:

$$H|k_j\rangle = \frac{1}{\sqrt{2}}(0|\rangle + (-1)^{k_j}|1\rangle) \quad \text{d.h.} \quad P(Y_j = 0) = \frac{1}{2} = P(Y_j = 1) \; .$$

BEMERKUNG (ERZEUGUNG VON ZUFALLSZAHLEN). Es ist gegenwärtig *keine* Maschine bekannt, die eine (beweisbar) perfekte statistische Gleichverteilung erzeugt. Dass man dennoch hofft, eine solche Maschine konstruieren zu können, liegt an den Eigenschaften quantenmechanischer Systeme in der Physik, die bei vielen Experimenten beobachtet wurden.

Instruktives Beispiel. Wir betrachten nochmals die Boolesche Negation (d.h. die Vertauschung der Komponenten eines 1-Qubit). Wir nehmen an, uns stehen Rechner „H“ und „W_π“ für die folgenden Matrizen zur Verfügung:

$$H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix} \quad \text{und} \quad W_\pi = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} .$$

Wir basteln aus diesen Bauelementen den „Quanten-Schaltkreis“:

$$\mathbf{x} \longrightarrow \boxed{\boxed{X} \rightarrow \boxed{H} \rightarrow \boxed{W_\pi} \rightarrow \boxed{H} \rightarrow \boxed{Y}} \longrightarrow y$$

Bei der Eingabe des Qubits $\mathbf{x} = x_0|0\rangle + x_1|1\rangle$ produziert der Schaltkreis das Qubit

$$\mathbf{v} = HW_\pi H\mathbf{x} ,$$

welches das Messergebnis Y bestimmt. Man rechnet sofort nach:

$$HW_\pi H = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = S \quad \text{d.h.} \quad \mathbf{v} = S\mathbf{x} = x_1|0\rangle + x_0|1\rangle ,$$

wie gewünscht.

SPRINGENDER PUNKT: Obwohl (wie wir schon oben gesehen haben) eine Messung im obigen Rechenprozess nach dem ersten Bauelelemnt „H“ eine perfekte statistische Gleichverteilung der beiden reinen Bits $|0\rangle$ und $|1\rangle$ zu Tage brächte, führt der weitere Rechenprozess dennoch(!) zu dem gewünschten (deterministischen) Ziel einer Realisierung der Booleschen NEGATION-Funktion.

Allerdings darf man den Schaltkreis nicht „öffnen“, um zu kontrollieren, welches Teilrechenergebnis nach dem ersten Schaltelement H vorliegt, und mit diesem weiterrechnen. Denn dann würde man *in jedem Fall* ein nicht-deterministisches Endergebnis erzielen:

$$\begin{aligned} W_\pi H|0\rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \\ W_\pi H|1\rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) \end{aligned}$$

2. Quantenschaltkreise

Es ist nun vermutlich klar, was man unter einem *Quantenschaltkreis* verstehen wird. Wir verallgemeinern einfach die Klasse der (reversiblen) Booleschen Schaltkreise, indem wir in die Knoten mit Operationen markieren, die unitären Transformationen auf passenden m-Qubits entsprechen.

In der (immer noch theoretischen) Praxis kommt man oft mit der Hadamardtransformation H und der Phasenverschiebung W_t als Grundtyp von Operationen aus.

Zusätzlich nehmen wir an, dass uns ein Quantenbaustein $\bigoplus$ zur Verfügung steht, der folgende Permutation $\sigma_\oplus$ auf A^2 (und die dadurch implizierte unitäre Transformation auf den 2-Qubits) realisiert:

$$\sigma_\oplus : \; x_1 \otimes x_2 \quad \longrightarrow \quad x_1 \otimes (x_1 \oplus x_2) \quad (\, x_1, x_2 \in A = \{0,1\} \,)$$

d.h.

$$\begin{array}{cccccccc} |00\rangle & \mapsto & |00\rangle & , & |10\rangle & \mapsto & |11\rangle \\ |01\rangle & \mapsto & |01\rangle & , & |11\rangle & \mapsto & |10\rangle \end{array} \quad .$$

Als Matrix bzgl. der Basis $A^2 = \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ wird dieser Baustein durch die folgende Permutationsmatrix beschrieben:

$$\bigoplus \quad \longleftrightarrow \quad \Sigma_\oplus = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \in \mathbb{C}^{4\times 4}$$

2.1. Kopieren. Wir können $\bigoplus$ verwenden, um auf den reinen(!) Qubits zu negieren: Bei $x_1 = 1$ vollzieht sich in der zweiten Komponente des 2-Bits die Boolesche Negation $\neg x_2$.

Ausserdem kann mit dem Baustein $\bigoplus$ das reine Bit x_1 auf die zweite Komponente kopiert werden: $x_2 = 0$ aktiviert diese Kopierfunktion.

Es überrascht deshalb vielleicht, dass es aber *un*möglich ist, allgemeine 1-Qubits zu kopieren.

THEOREM 3.1 (Unmöglichkeit des Kopierens). *Es gibt* keine *unitäre Transformation U, die bei Eingabe eines beliebigen 1-Qubits x und des reinen Bits $|0\rangle$ das 2-Qubit $x \otimes x$ produziert:*

$$x \otimes |0\rangle \;\longrightarrow\; \boxed{U} \;\longrightarrow\; x \otimes x \; .$$

Bew. Nehmen wir an, die unitäre Transformation U habe doch die Kopiereigenschaft und betrachten die zu kopierenden 1-Qubits

$$|0\rangle \; , \; |1\rangle \quad \text{und} \quad z = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \; .$$

Dann finden wir (nach Annahme):

$$U|00\rangle = |00\rangle \; , \; U|10\rangle = |11\rangle \quad \text{und} \quad U(z \otimes |0\rangle) = z \otimes z = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \; .$$

Andererseits haben wir

$$z \otimes |0\rangle = \frac{1}{\sqrt{2}}\big(|00\rangle + |10\rangle\big)$$

und somit aufgrund der Linearität der Transformation U einen Widerspruch:

$$U(z \otimes |0\rangle) = \frac{1}{\sqrt{2}}\big(U|00\rangle + U|10\rangle\big) = \frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big) \; .$$

$\diamond$

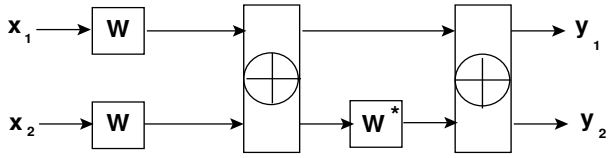


ABBILDUNG 1. x_1-kontrollierte V_t-Funktion

BEMERKUNG (FEHLERKORREKTUR). Die Unmöglichkeit, unbekannte Qubits zu kopieren, stellt ein Problem für die Konstruktion Übertragungsfehler korrigierender Quanten-Codes dar, da man die klassische Standardtechnik der Codewortwiederholungen nicht ohne weiteres verwenden kann. Dieses Problem kann zwar im Prinzip gelöst werden. Die Details sind jedoch zu aufwendig, um hier näher darauf einzugehen.

2.2. Kontrollierte Phasenverschiebungen. Zu einer gegebenen Phasenverschiebung W_t wollen wir nun ein Schaltelement V_t konstruieren mit der folgenden Eigenschaft

$$x_1 \otimes x_2 \;\to\; \boxed{V_t} \;\to\; y_1 \otimes y_2 \;=\; \begin{cases} x_1 \otimes x_2 & \text{falls } x_1 = |0\rangle \\ x_1 \otimes (W_t x_2) & \text{falls } x_1 = |1\rangle \end{cases}$$

Das Kontrollbit x_1 bewirkt im Fall $x_1 = |0\rangle$ keine Veränderung. Im Fall $x_1 = |1\rangle$ wird ein 2-Qubit erzeugt, das einer Phasenverschiebung $W_t x_2$ im zweiten Eingabequibit entspricht.
Wir benutzen einen Schaltkreis, wie in der Abbildung angegeben, mit Phasenverschiebung

$$W = \begin{pmatrix} 1 & 0 \\ 0 & e^{\mathrm{i}t/2} \end{pmatrix} \quad \text{mit} \quad \text{und} \quad W^* = W^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & e^{-\mathrm{i}t/2} \end{pmatrix}$$

Im Fall $x_1 = |0\rangle$ haben wir wegen $WW^* = I$ offenbar die gewünschte Eigenschaft

$$|0\rangle \otimes x_2 \quad \rightsquigarrow \quad y_1 \otimes y_2 = |0\rangle \otimes x_2 \; .$$

Setzen wir zur Abkürzung $\omega = e^{\mathrm{i}t/2}$, so finden wir weiterhin

$$\begin{array}{lllllllll} |1\rangle \otimes |0\rangle & \overset{W}{\to} & \omega|10\rangle & \overset{\oplus}{\to} & \omega|11\rangle & \overset{W^*}{\to} & \omega\omega^{-1}|11\rangle & \overset{\oplus}{\to} & |10\rangle = |1\rangle \otimes (W_t|0\rangle) \\ |1\rangle \otimes |1\rangle & \overset{W}{\to} & \omega^2|11\rangle & \overset{\oplus}{\to} & \omega^2|10\rangle & \overset{W^*}{\to} & \omega^2|10\rangle & \overset{\oplus}{\to} & \omega^2|11\rangle = |1\rangle \otimes (W_t|1\rangle) \; , \end{array}$$

welches die behauptete Funktion impliziert.

BEMERKUNG. Die kontrollierte Phasenverschiebung V_t wird auch bei der noch zu diskutierenden Quanten-Fouriertransformation wichtig werden.

BEMERKUNG. *Die obigen „Rechnungen“ im Quantenschaltkreis zeigen schon jetzt, dass man sich die Kanten des Schaltkreisgraphen* **nicht(!)** *wie bei Booleschen Rechnungen als physikalische Leitungen vorstellen darf, welche die Informationseinheiten „Bits“ von einem Schaltelement zum nächsten transportieren. Vielmehr breitet sich die Rechnung im Schaltkreis „wellenförmig“ von der Eingabeeinheit X zur Ausgabeeinheit Y aus.* (Das wird später vielleicht noch klarer werden.)

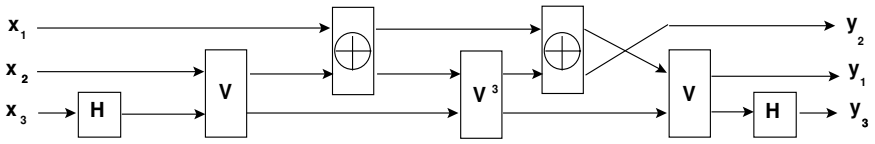


ABBILDUNG 2. Quantenschaltkreis für Toffolifunktion

2.3. Boolesche Schaltkreise. Mit dem Modell eines aus Quantenschaltelementen (d.h. unitären Transformationen auf Qubits) ausgerüsteten Schaltkreises haben wir ein Rechenmodell, das mindestens so effizient ist wie das der Booleschen Schaltkreise. Wir können nämlich die Toffolifunktion T auch durch einen Quantenschaltkreis simulieren und folglich alle reversiblen Booleschen Schaltkreise direkt übertragen.

Wir benötigen dafür als Bauteile nur die Hadamardtransformation H und die kontrollierte Phasenverschiebung um den Winkel $t = \pi/2$:

$$V = V_{\pi/2} \quad (\text{ d.h. } e^{\mathrm{i}t} = e^{\mathrm{i}\pi/2} = \mathrm{i}\ ) .$$

Damit können wir wie folgt einen Quantenschaltkreis für die Toffolifunktion

$$x_1 \otimes x_2 \otimes x_3 \quad \stackrel{T}{\longrightarrow} \quad x_1 \otimes x_2 \otimes (x_3 \oplus (x_1 \wedge x_2)) \quad (\ x_1, x_2, x_3 \in A = \{0,1\}\ ) ,$$

konstruieren. Um einzusehen, dass der angegebene Schaltkreis wirklich die Toffolifunktion T ämuliert, brauchen wir nur dessen Wirkung auf die möglichen Eingaben $x \in A^3$ zu überprüfen. Wir finden dann z.B.

$$\begin{array}{lllll} |110\rangle & \stackrel{H}{\to} & \frac{1}{\sqrt{2}}(|110\rangle + |111\rangle & \stackrel{V}{\to} & \frac{1}{\sqrt{2}}(|110\rangle + \mathrm{i}|111\rangle) \\ & \stackrel{V^3}{\to} & \frac{1}{\sqrt{2}}(|100\rangle + \mathrm{i}|101\rangle) & \stackrel{\oplus}{\to} & \frac{1}{\sqrt{2}}(|110\rangle + \mathrm{i}|111\rangle) \\ & \stackrel{H}{\to} & |111\rangle & & \end{array} \quad \begin{array}{l} \stackrel{\oplus}{\to} \ \frac{1}{\sqrt{2}}(|100\rangle + \mathrm{i}|101\rangle) \\ \stackrel{V}{\to} \ \frac{1}{\sqrt{2}}(|110\rangle - |111\rangle) \end{array}$$

und ebenso:

$$\begin{array}{lllll} |111\rangle & \stackrel{H}{\to} & \frac{1}{\sqrt{2}}(|110\rangle - |111\rangle & \stackrel{V}{\to} & \frac{1}{\sqrt{2}}(|110\rangle - \mathrm{i}|111\rangle) \\ & \stackrel{V^3}{\to} & \frac{1}{\sqrt{2}}(|100\rangle - \mathrm{i}|101\rangle) & \stackrel{\oplus}{\to} & \frac{1}{\sqrt{2}}(|110\rangle + \mathrm{i}|111\rangle) \\ & \stackrel{H}{\to} & |110\rangle & & \end{array} \quad \begin{array}{l} \stackrel{\oplus}{\to} \ \frac{1}{\sqrt{2}}(|100\rangle + \mathrm{i}|101\rangle) \\ \stackrel{V}{\to} \ \frac{1}{\sqrt{2}}(|110\rangle + |111\rangle) \end{array}$$

(Die übrigen Verifikationen vollziehen sich auf die gleiche Weise.)

2.3.1. **Das Problem von Deutsch.** Es soll entschieden werden, ob eine Boolesche Funktion $f : \{0,1\} \to \{0,1\}$ konstant ist oder nicht. Es wird angenommen, dass f nur indirekt als Schaltelement zur Verfügung steht.

- Kann ein Schaltkreis konstruiert werden, der die Frage entscheidet und f als Schaltelement nur einmal enthält?

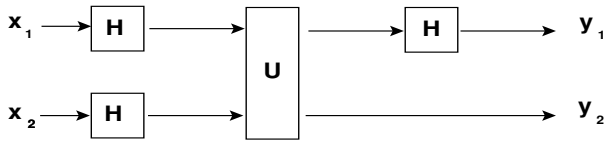


ABBILDUNG 3. Schaltkreis für Deutsch-Problem

Die Antwort ist (überraschenderweise?) „ja“, wenn f als Quantenbaustein $U = U_f$ (d.h. als unitäre Transformation) vorliegt, der die folgende Permutation der Elemente von A_2 realisiert:

$$x_1 \otimes x_2 \quad \stackrel{U}{\longrightarrow} x_1 \otimes (x_2 \oplus f(x_1)) \quad (\ x_1, x_2 \in A = \{0,1\}\ )\ .$$

Wir behaupten, dass der (in der Abbildung) angegebene Schaltkreis das Gewünschte leistet. Dazu beobachten wir zuerst

$$|01\rangle \quad \stackrel{H}{\longrightarrow} \quad \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle) = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)\ .$$

Sei nun z.B. $f(0) = 0 = f(1)$. Dann finden wir

$$|01\rangle \quad \stackrel{UH}{\longrightarrow} \quad \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle = \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle)$$

und deshalb

$$|01\rangle \stackrel{HUH}{\longrightarrow} \frac{1}{\sqrt{2}}|0\rangle \otimes (|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}(|00\rangle - |01\rangle)$$

Die Einzelbitmessung Y_1 an der Ausgabeeinheit wird folglich ein deterministisches Ergebnis liefern:

$$P(Y_1 = 0) = 1\ .$$

(Das Nachrechnen derselben Eigenschaft im Fall $f(0) = 1 = f(1)$ sei dem Leser zur Übung überlassen.)

Im Fall $f(0) = 0$ und $f(1) = 1$ finden wir jedoch

$$|01\rangle \quad \stackrel{HUH}{\longrightarrow} \quad \frac{1}{\sqrt{2}}(|10\rangle - |11\rangle)$$

und somit

$$P(Y_1 = 1) = 1\ .$$

2.3.2. **Quantenparallelität.** Die quantenrechnerische „Lösung“ des Problems von Deutsch legt vielleicht nahe, dass Quantenrechner tatsächlich „effizienter“ als herkömmliche Rechner sein könnten. Warum dies so sein sollte, wird oft mit der folgenden Plausibilitätsüberlegung zusätzlich illustriert.

Wir betrachten eine Boolesche Funktion

$$f : \{0,1\}^n \to \{0,1\} \ .$$

Gesetzt wir hätten ein Rechenelement U_f zur Verfügung, welches (auf A^{n+1}) die Permutation

$$(x, y) \mapsto (x, y \oplus f(x)) \quad (x \in A^n)$$

realisiert, dann würde *eine*(!) Anwendung von U_f (und der Hadamardtransformation H) im Prinzip *sämtliche* 2^n Werte von f berechnen:

$$\begin{aligned} [H(0) \otimes \ldots \otimes H(0)] \otimes (0) \quad &= \quad \frac{1}{2^{n/2}} \sum_{x \in A^n} |x\rangle \otimes |0\rangle \\ &\xrightarrow{U_f} \quad \frac{1}{2^{n/2}} \sum_{x \in A^n} |x\rangle \otimes |f(x)\rangle \ =: \ \varphi \ . \end{aligned}$$

Das $(n+1)$-Qubit φ ist eine Überlagerung der 2^n reinen $(n+1)$-Qubits

$$|x\rangle \otimes |f(x)\rangle \ \in \ A^{n+1} \ .$$

In ihm steckt die gesamte Information über f. Dieses Phänomen wird auch als *Quantenparallelität* bezeichnet.

KAPITEL 4

Hilberträume und Fouriertransformation

Für unsere Zwecke verstehen wir unter einem (komplexen) *Hilbertraum* einen Vektorraum $\mathcal{H}$ über dem Skalarbereich $\mathbb{C}$ mit einer (endlichen oder abzählbaren) *Basis* $B \subseteq \mathcal{H}$:

(B_1) B ist linear unabhängig.
(B_2) $\mathrm{lin}B = \mathcal{H}$

(Mit $\mathrm{lin}B$ bezeichnen wir die Menge aller *endlichen* Linearkombinationen von Vektoren in B.)

BEMERKUNG. Der Begriff „Hilbertraum“ ist in der Mathematik noch etwas allgemeiner als hier definiert. Für das Quantenrechenen ist das hier formulierte Modell jedoch völlig ausreichend.

Koordinatenräume. Der Koordinatenraum $\mathbb{C}^n$ ist ein Beispiel einers Hilbertraums der Dimension n mit der sog. *Standardbasis* von Einheitsvektoren

$$B = \{\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_{n-1}\} \ ,$$

wobei die $\mathbf{e}_k$ die Einheitsvektoren sind:

$$\mathbf{e}_1 = (1,0,0,0,\ldots),\ \mathbf{e}_2 = (0,1,0,0,\ldots),\ \mathbf{e}_3 = (0,0,1,0,\ldots) \quad \text{usw.}$$

Die entsprechenden Einheitsvektoren $\mathbf{e}_k \in \mathbb{C}^{\mathbb{N}}$ des unendlich-dimensionalen Koordinatenraums $\mathbb{C}^{\mathbb{N}}$ bilden eine Basis des Hilbertraums $\mathcal{P} \subseteq \mathbb{C}^{\mathbb{N}}$ aller Vektoren mit höchstens endlich vielen Koeffizienten $\neq 0$.

Polynome. Die eben genannten Koordinatenräume sind das „Beschreibungsgerüst“ von Vektorräumen, die bei konkreten Modellierungen auftreten. Ein wichtiges Beispiel hierzu ist die Menge $\mathbb{C}[x]$ aller Polynome mit komplexen Koeffizienten

$$p(x) = c_0 + c_1 x + \ldots + c_{N-2}x^{N-2} + c_{N-1}x^{N-1} \quad (\ c_k \in \mathbb{C}\ ) \ .$$

mit der Standardbasis

$$B = \{1, x, x^2, \ldots, x^k, \ldots\} \ ,$$

die wir durch die entsprechenden Koeffizientenvektoren beschreiben:

$$p(x) \in \mathbb{C}[x] \quad \longleftrightarrow \quad (c_0, c_1, \ldots, c_{N-1}, 0, \ldots) \in \mathcal{P} \ .$$

Aus dem Hilbertraum $\mathbb{C}[x]$ der Polynome gewinnt man durch Substitution andere Darstellungen, die zwar zum Polynomraum isomorph sind, aber durch ihre konkretere Gestalt für konkretere Anwendungen Vorteile bieten.

Periodische Funktionen. Wir substituieren $x = e^{\mathrm{i}t}$ in das Polynom $p(x)$ und erhalten die Funktion

$$f(t) = p(e^{\mathrm{i}t}) = \sum_{k=0}^{N-1} c_k e^{\mathrm{i}kt} = \sum_{k=0}^{N-1} c_k(\cos kt + \mathrm{i}\sin kt) \ .$$

$f(t)$ ist eine auf $\mathbb{R}$ definierte Funktion mit Periode 2π:

$$f(t) = f(t + 2\pi) \quad (\ t \in \mathbb{R}\ ) \ .$$

Die Koeffizienten c_k sind die *Fourierkoeffizienten* der Funktion $f(t)$.

Mit $\mathcal{F}$ bezeichnen wir den *Funktionenraum* aller solcher aus Polynomen per Substitution gewonnenen periodischen Funktionen. Die entsprechende Standardbasis von $\mathcal{F}$ sieht dann so aus:

$$B = \{1, e^{\mathrm{i}}t, e^{\mathrm{i}2t}, \ldots, e^{\mathrm{i}kt}, \ldots\} \ .$$

Quantenzustände. Wir substituieren in das Polynom $p(x) \in \mathbb{C}[x]$ (formal) die Binärdarstellung der natürlichen Zahl $k \in \mathbb{N}$ anstatt des Basiselements x^k und erhalten mit

$$c_0|0\rangle + c_1|1\rangle + \ldots + c_k|k\rangle + \ldots + c_{N-1}|N-1\rangle$$

im Fall $N = 2^n$ genau dann ein n-Qubit, wenn

$$|c_0|^2 + |c_1|^2 + \ldots + |c_{N-1}|^2 = 1 \ .$$

1. Das Tensorprodukt

Bezüglich der endlich-dimensionalen Hilberträume $\mathcal{H}_1$ und $\mathcal{H}_2$ mit Basen

$$B_1 = \{e_0, e_1, \ldots, e_{M-1}\} \quad \text{und} \quad B_2 = \{f_0, f_1, \ldots, e_{N-1}\}$$

ist das *Tensorprodukt* folgende Konstruktion:

Man bildet die Menge aller (formalen) Produkte von Paaren von Elementen in B_1 und B_2,

$$B_1 \otimes B_2 = \{e_k \otimes f_j | k = 0, 1, \ldots, M-1, j = 0, 1, \ldots, M-1, \}$$

und betrachtet den Vektorraum $\mathcal{H}_1 \otimes \mathcal{H}_1$ aller (formalen) Linearkombinationen

$$\sum_k \sum_j c_{kj}\ e_k \otimes f_j \quad (\ c_{kj} \in \mathbb{C}\ ) \ .$$

Wegen $|B_1 \otimes B_2| = |B_1| \cdot |B_2|$ ist der Vektorraum $\mathcal{H}_1 \otimes \mathcal{H}_2$ ein MN-dimensionaler Hilbertraum, in den sich das kartesische Produkt $\mathcal{H}_1 \times \mathcal{H}_2$ sehr natürlich abbilden lässt:

$$(v, w) \ \mapsto \ v \otimes w = \sum_{k=0}^{M-1} \sum_{j=0}^{N-1} (v_k w_j)\ f_k \otimes e_j \quad (\ v = \sum_k v_k e_k\ , w = \sum_j w_j f_j\ ) \ .$$

Das Element $v \otimes w \in \mathcal{H}_1 \otimes \mathcal{H}_2$ heisst *Tensorprodukt* der Vektoren $v \in \mathcal{H}_1$ und $w \in \mathcal{H}_2$ (obwohl dieses Produkt weder in $\mathcal{H}_1$ noch in $\mathcal{H}_2$ liegt). Aus der Definition folgen sogleich die Rechenregeln, die zeigen, dass man in erwarteter Weise

bei Vektoren $v_1, v_2 \in \mathcal{H}_1$, $w_1, w_2 \in \mathcal{H}_2$ und Skalaren $\lambda, \mu \in \mathbb{C}$ „ausmultiplizieren“ darf:

$$\begin{aligned}(v_1 + v_2) \otimes (w_1 + w_2) &= v_1 \otimes w_1 + v_1 \otimes w_2 + v_2 \otimes w_1 + v_2 \otimes w_2 \\ (\lambda v) \otimes (\mu w) &= (\lambda\mu) v \otimes w\end{aligned}$$

Identifizieren wir den Vektor $e_k \in B_1$ mit der Binärdarstellung $|k\rangle$ der natürlichen Zahl $k < M$ und f_j mit der Darstellung $|j\rangle$ von $j < N$ (bzgl. $M = 2^m$ und $N = 2^n$), so können wir das Tensorprodukt mit der Verkettung

$$e_k \otimes f_j \quad \longleftrightarrow \quad |k\rangle \otimes |j\rangle \;=\; |k \otimes j\rangle$$

von Qubits identifizieren:

$$\sum_{k=0}^{M-1} \sum_{j=0}^{N-1} v_k w_j \; e_k \otimes f_j \quad \longleftrightarrow \quad \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} v_k w_j |k \otimes |j\rangle \; .$$

Das Tensorprodukt kann auch für den abzählbar-unendlichen Hilbertraum aller Polynome erklärt werden. Hierzu gehen wir genauso vor wie im endlich-dimensionalen Fall: Wir nehmen zwei Exemplare der der Standardbasis

$$B_1 = \{x^0, x^1, \ldots, x^k, \ldots\} \quad \text{und} \quad B_2 = \{y^0, y^1, \ldots, y^j, \ldots\}$$

und bilden die Menge $B_1 \otimes B_2$ aller (formalen) Produkte

$$x^k \otimes y^j \quad \longleftrightarrow \quad x^k y^j \; .$$

Dem Tensorprodukt $p(x) \otimes q(y)$ entspricht dann das normale Produkt:

$$\begin{aligned}\Big(\sum_{k=0}^{m} a_k x^k\Big) \otimes \Big(\sum_{j=0}^{n} b_j y^j\Big) &= \sum_{k=0}^{m} \sum_{j=0}^{n} a_k b_j \; x^k \otimes y^j \\ \Big(\sum_{k=0}^{m} a_k x^k\Big) \cdot \Big(\sum_{j=0}^{n} b_j y^j\Big) &= \sum_{k=0}^{m} \sum_{j=0}^{n} a_k b_j \; x^k y^j\end{aligned}$$

Transformationen auf dem Tensorprodukt. Wählen wir eine andere Basis von $\mathcal{H}_1$, z.B. $\mathcal{B}_1'$, so erhalten wir *denselben* Tensorproduktraum

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \; ,$$

denn der von $\mathcal{B}_1' \otimes \mathcal{B}_2$ erzeugte Raum $\mathcal{H}'$ ist in $\mathcal{H}$ enthalten (da wir jedes $b' \in \mathcal{B}_1'$ als Linearkombination von $\mathcal{B}_1$ ausdrücken können). Dasselbe Argument zeigt umgekehrt

$$\mathcal{H} \subseteq \mathcal{H}' \quad \text{d.h.} \quad \mathcal{H} = \mathcal{H}' \; .$$

Daraus ersehen wir, dass beliebige lineare Transformationen $T_i : \mathcal{H}_i \to \mathcal{H}_i$ eine lineare Transformation

$$T_1 \otimes T_2 : \mathcal{H}_1 \otimes \mathcal{H}_2 \to \mathcal{H}_1 \otimes \mathcal{H}_2$$

induzieren, die durch ihre Wirkung auf die Basiselemente festgelegt ist:

$$b_1 \otimes b_2 \;\to\; T_1(b_1) \otimes T_2(b_2) \quad (b_1 \in \mathcal{B}_1, b_2 \in \mathcal{B}_2) \; .$$

2. Innere Produkte und unitäre Transformationen

In Bezug auf die Basis $B = \{e_0, e_1, \ldots\}$ des Hilbertraumes $\mathcal{H}$ definieren wir das *innere Produkt* von Elementen $v, w \in \mathcal{H}$ wie folgt:

$$v = \sum_{e \in B} \alpha_e e \ , \ w = \sum_{e \in B} \beta_e e \quad \longrightarrow \quad v^* w := \sum_{e \in B} \overline{\alpha}_e \beta_e \ .$$

Daraus leiten wir die übliche *Hermitesche Norm* (die im Fall eines reellen Skalarbereichs der Euklidischen Norm entspricht) ab:

$$\|v\|^2 := \sum_{e \in B} |\alpha_e|^2 = v^* v \ .$$

NOTA BENE: $\overline{\alpha}_e$ ist die zu α_e konjugiert-komplexe Zahl. Die Summen sind alle wohldefiniert, da nach unserer Voraussetzung über Hilberträume jede Summe nur endlich viele von 0 verschiedene Summanden umfasst.

Eine *lineare Transformation* von $\mathcal{H}$ ist ein Operator $U : \mathcal{H} \to \mathcal{H}$ mit den Eigenschaften

(LT_1) $U(v + w) = Uv + Uw$.
(LT_2) $U(\lambda v) = \lambda U v$.
(LT_3) Die Umkehrtransformation $U^{-1} : \mathcal{H} \to \mathcal{H}$ existiert.

Die lineare Transformation U heisst *unitär*, wenn für alle $v \in \mathcal{H}$ gilt:

$$\|Uv\| = \|v\| \ .$$

BEMERKUNG. Eine (lineare) unitäre Transformation U erhält nicht nur die Länge von Vektoren, sondern auch das innere Produkt:

$$v^* w = (Uv)^* (Uw) \ .$$

Ist $\mathcal{H}$ endlich-dimensional, so hat U eine Matrixdarstellung (z.B. bzgl. der Basis B). In diesem Fall finden wir für alle $v, w \in \mathcal{H}$

$$v^* w = (Uv)^* (Uw) = v^* (U^* U) w \quad \text{und folglich} \quad U^* U = I \ .$$

Unter einer linearen Transformation $U : \mathcal{H} \to \mathcal{H}$ geht die Basis B in eine Basis

$$UB = \{Ue \mid e \in B\}$$

über. Um das einzusehen, betrachten wir die Darstellung des Hilbertraums $\mathcal{H}$ als Koordinatenraum $\mathcal{P}$ mit Basis

$$B = \{e_0, \ldots, e_k, \ldots\}$$

und U als die entsprechende Transformation auf $\mathcal{P}$. Die Invertierbarkeit von U besagt, dass die Gleichung

$$\mathbf{v} = U\mathbf{x}$$

mit einem eindeutig bestimmten Koordinatenvektor $\mathbf{x} \in \mathcal{P}$ lösbar ist. Das heisst aber gerade, dass jedes Element des Hilbertraums auch durch einen Koordinatenvektor $\mathbf{x}$ bzgl. UB angegeben werden kann.

Ist die Transformation U unitär, dann bleibt zudem die Länge der Einheitsvektoren erhalten:

$$\|e\| = 1 = \|Ue\| \quad (\, e \in B \,) \, .$$

U führt darum (im Fall $N = 2^n$) insbesondere n-Qubits in n-Qubits über.

2.1. Unitäre Transformationen und Eigenwerte. Ist $\lambda \in \mathbb{C}$ ein Eigenwert der unitären Transformation $U : \mathcal{H} \to \mathcal{H}$ mit Eigenvektor $v \in \mathcal{H}$, dann gilt

$$\|v\| = \|Uv\| = |\lambda| \cdot \|v\| \quad \text{d.h.} \quad |\lambda| = 1 \, .$$

Es gibt also ein $0 \leq t_\lambda < 2\pi$ derart, dass

$$\lambda = e^{\mathrm{i}t_\lambda} = \cos t_\lambda + \mathrm{i} \sin t_\lambda \, .$$

Ist $U \in \mathbb{C}^{N \times N}$ eine unitäre Matrix, so ist U *normal* (d.h. $U^*U = UU^*$). Aus der linearen Algebra wissen wir deshalb, dass $\mathbb{C}^N$ eine Basis P besitzt, die nur aus Eigenvektoren von U besteht und selber eine unitäre Matrix darstellt. Bezgl. dieser Basis P wird die unitäre Transformation U durch eine Diagonalmatrix D beschrieben, deren Diagonale die Eigenwerte von U enthält:

$$P^*UP \; = \; \begin{pmatrix} e^{\mathrm{i}t_1} & 0 & 0 & \dots & 0 \\ 0 & e^{\mathrm{i}t_2} & 0 & \dots & 0 \\ \vdots & & \ddots & & \\ 0 & & \dots & & e^{\mathrm{i}t_N} \end{pmatrix} \; = \; D$$

Eine unitäre Transformationen $U : \mathcal{H} \to \mathcal{H}$ eines Hilbertraums, der durch den Koordinatenraum $\mathbb{C}^N$ beschrieben wird, kann man also (von der Basis $P \subseteq \mathcal{H}$ der Eigenvektoren von U aus betrachtet) so interpretieren: U bewirkt in jeder Komponente k eine Phasenverschiebung (gemäss dem zugehörigen Eigenwert $\lambda_k = e^{\mathrm{i}t_k}$):

$$v = \sum_{p_k \in P} v_k p_k \quad \Longrightarrow \quad Uv = \sum_{p_k \in P} (e^{\mathrm{i}t_k} v_k) p_k \quad (\, v_k \in \mathbb{C} \,) \, .$$

Philosophie. Lineare Transformationen $U : \mathcal{H} \to \mathcal{H}$ auf dem Hilbertraum $\mathcal{H}$ (bzw. $U : \mathcal{P} \to \mathcal{P}$ im entsprechenden Koordinatenraum $\mathcal{P}$) gestatten zwei grundsätzlich verschiedene (mathematisch aber völlig äquivalente) philosophische Interpretationen der Gleichungen

$$y = Ux \quad \text{bzw. (in } \mathcal{H} \text{)} \quad \mathbf{y} = U\mathbf{x} \text{ (im Koordinatenraum } \mathcal{P} \text{) .}$$

(D) *Dynamische Interpretation:* Das Element $x \in \mathcal{H}$ wird mittels U in das Element $y = Ux \in \mathcal{H}$ transformiert.

(S) *Statische Interpretation:* Das Element $y \in \mathcal{H}$ wird beim Übergang von der Basis B zur Basis UB von „**y**“ in „**x**“ umbenannt.

BEMERKUNG. Welche der beiden Interpretationen (D) und (S) „richtig“ ist, ist persönliche Auffassungssache. Weder Mathematik (noch Physik) können hier eine Entscheidungshilfe liefern.

2.2. Die Schrödingersche Wellengleichung. Betrachten wir eine differenzierbare Funktion $\varphi : \mathbb{R} \to \mathbb{C}$ in the reellen Variablen t, die zu dem gegebenen Parameter $a \in \mathbb{C}$ folgende Differentialgleichung erfüllen möge:

$$\frac{\partial \varphi(t)}{\partial t} = a\varphi(t) \ .$$

Dann ist $\varphi(t)$ von der Form

$$\varphi(t) \ = \ Ke^{at} = K \sum_{k=0}^{\infty} \frac{a^k t^k}{k!} \quad (\text{mit } K = \varphi(0) \ ) \ .$$

Diese Beobachtung gilt allgemeiner für vektorwertige $\varphi : \mathbb{R} \to \mathbb{C}^N$ und Diagonalmatrizen

$$A = \begin{pmatrix} a_1 & 0 & 0 & \dots & 0 \\ 0 & a_2 & 0 & \dots & 0 \\ 0 & 0 & a_3 & \dots & \\ \vdots & \dots & & & \vdots \\ 0 & 0 & \dots & & a_N \end{pmatrix}$$

$$\frac{\partial \varphi}{\partial t} = A\varphi \quad \Longrightarrow \quad \varphi(t) = \begin{pmatrix} e^{a_1 t} & 0 & 0 & \dots & 0 \\ 0 & e^{a_2 t} & 0 & \dots & 0 \\ \vdots & \dots & & & \vdots \\ 0 & 0 & \dots & & e^{a_N t} \end{pmatrix} \begin{pmatrix} \varphi_1(0) \\ \varphi_2(0) \\ \vdots \\ \varphi_N(0) \end{pmatrix} = e^{At}\varphi(0)$$

Sei nun $P \in \mathbb{C}^{N \times N}$ unitär und $\psi(t) = P\varphi(t)$. Dann gilt

$$\frac{\partial \psi(t)}{\partial t} = P\frac{\partial \varphi(t)}{\partial t} = PA\varphi(t) = PAP^*\psi(t) = H\psi(t) \quad (\text{mit } H = PAP^*) \ .$$

Wegen $A = P^*HP$ und $A^k = P^*H^kP$ finden wir die Lösung in der Form

$$\psi(t) = P\varphi(t) = Pe^{At}\varphi(0) = P(P^*e^{Ht}P)\varphi(0) = e^{Ht}\psi(0) \ .$$

Im Schrödingerschen Modell der Quantenmechanik entwickelt sich ein Quantensystem aus einem Anfangszustand $\psi(0)$ in der Zeit t zu $\psi(t)$ gemäss

$$\psi(t) = U_t\psi(0) \ ,$$

wobei U_t als differenzierbare Matrixfunktion „ohne Gedächtnis“ angenommen wird, d.h.

$$\psi(t+h) \ = \ U_h\varphi(t) \quad \text{für alle } \ h > 0 \ .$$

Daraus ergibt sich

$$\frac{\partial \psi(t)}{\partial t} = \lim_{h \to 0} \frac{\psi(t+h) - \psi(t)}{h} = \lim_{h \to 0} \frac{1}{h}(U_h - I)\psi(t) = H\psi(t)$$

mit

$$H = \lim_{h \to 0} \frac{1}{h}(U_h - I) \ .$$

Der Operator H wird als *normal* angenommen (d.h. $H^*H = HH^*$). Aus der linearen Algebra weiss man, dass diese Eigenschaft gleichbedeutend ist mit der Diagonalisierbarkeit von H durch eine unitäre Matrix P:

$$P^*HP = A \quad (A \text{ Diagonalmatrix}) .$$

Normalisieren wir mit der imaginären Einheit $\mathrm{i} \in \mathbb{C}$ und dem *Planckschen Wirkungsquantumg* $\hbar$, so erhalten wir die Schrödingersche Differentialgleichung, welche die Maxwellschen Wellengleichungen verallgemeintert:

$$\boxed{\mathrm{i}\hbar\frac{\partial\psi(t)}{\partial t} \;=\; H\psi(t) \quad \text{bzw.} \quad \frac{\partial\psi(t)}{\partial t} \;=\; -\frac{\mathrm{i}}{\hbar}H\psi(t)}$$

Die Lösung der Schrödingergleichung hat nach den vorangehenden Überlegungen die Form

$$\psi(t) = U_t\psi(0) \quad \text{mit} \quad U_t = e^{-\mathrm{i}Ht/\hbar} .$$

Die Matrix U_t ist als Produkt unitärer Matrizen selber unitär:

$$U_t = e^{-\mathrm{i}Ht} = PAP^* = P\begin{pmatrix} e^{-\mathrm{i}a_1t/\hbar} & 0 & 0 & \dots & 0 \\ 0 & e^{-\mathrm{i}a_2t/\hbar} & 0 & \dots & 0 \\ \vdots & \dots & & & \vdots \\ 0 & 0 & \dots & & e^{-\mathrm{i}a_Nt/\hbar} \end{pmatrix} P^*$$

Stellen wir $\psi(t)$ in den Koordinaten bzgl. der Basis P dar, so erhalten wir eine Funktion, die in jeder Komponente periodisch ist. (Die Perioden der einzelnen Komponenten können allerdings durchaus verschieden sein!)

BEMERKUNG.*Die Ausbreitung von Wellen, die den Schrödingerschen (Maxwellschen) Gesetzmässigkeiten genügen, erfolgt also nach dem gleichen mathematischen Prinzip wie die Entwicklung von Qubits.*

3. Diskrete Fouriertransformation (DFT) im Koordinatenraum

Die sog. *diskrete Fouriertransformation* (DFT) beruht auf der folgenden simplen (durch Ausmultiplizieren zu bestätigenden) Polynomidentität:

$$\boxed{(1-x)(1+x+x^2+\ldots+x^{N-1}) \;=\; 1-x^N}$$

Substituieren wir in diese Identität eine sog. (komplexe) N-te *Einheitswurzel*, nämlich ein Element $z \in \mathbb{C}$ mit $z^N = 1$, so ergibt sich mit $x = z$

$$\frac{1}{N}\sum_{k=0}^{N-1} z^k \;=\; \frac{1+z+\ldots+z^{N-1}}{N} \;=\; \begin{cases} 1 & \text{falls } z = 1, \\ 0 & \text{falls } z \neq 1. \end{cases}$$

Ein in diesem Zusammenhang wichtiges Beispiel einer Einheitswurzel ist

$$\omega := e^{2\pi\mathrm{i}/N} = \cos\frac{2\pi}{N} + \mathrm{i}\sin\frac{2\pi}{N} .$$

ω ist nicht nur eine Einheitswurzel sondern sogar eine *primitive Einheitswurzel* in dem Sinn, dass

$$\omega^k \neq 1 \quad \text{für alle } k = 1, \ldots, N-1 \; .$$

BEMERKUNG. Primitive Einheitswurzeln interessieren auch bei ganzen Zahlen. Viele Verfahren der Kryptographie bauen ihre Sicherheit auf die Annahme, dass es zu gegebenen Zahlen $x, n \in \mathbb{Z}$ im allgemeinen schwierig ist, ein $1 \leq k \leq n-1$ zu berechnen mit der Eigenschaft

$$x^k \equiv 1 \mod n \quad \text{aber} \quad x^m \not\equiv 1 \mod n \quad \text{für alle } 1 \leq m \leq k-1 \; .$$

Wir werden später sehen, dass ein Quantenrechner dieses Problem im Prinzip effizient lösen könnte.

Mit ω ist natürlich auch jede Potenz ω^k eine N-te Einheitswurzel und wir haben

$$\omega^{-k} = \overline{\omega}^k \quad (\; k \in \mathbb{Z} \;) \; .$$

ω induziert somit die (symmetrische) unitäre Matrix

$$\Omega_N := \frac{1}{\sqrt{N}} (\omega^{kj})_{k,j=0,\ldots,N-1} = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega & \omega^2 & \ldots & \omega^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^k & \omega^{k2} & \ldots & \omega^{k(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{(N-1)2} & \ldots & \omega^{(N-1)(N-1)} \end{pmatrix}$$

Dass Ω_N tatsächlich unitär ist, rechnet man leicht nach, denn das Produkt der k-ten Zeile von $\Omega_N^* = \overline{\Omega}_N$ mit der j-ten Spalte von Ω_N ist

$$\frac{1}{\sqrt{N}} \frac{1}{\sqrt{N}} \sum_{\ell=0}^{N-1} \overline{\omega}^{k\ell} \omega^{\ell j} \; = \; \frac{1}{N} \sum_{\ell=0}^{N-1} (\omega^{j-k})^\ell \; = \; \begin{cases} 1 & \text{falls } j = k, \\ 0 & \text{falls } j \neq k. \end{cases}$$

Die durch $\Omega_N : \mathbb{C}^N \to \mathbb{C}^N$ gegebene unitäre Transformation heisst *(diskrete) Fouriertransformation* des Koordinatenraums $\mathbb{C}^N$. Der Vektor $\Omega_N \mathbf{c} \in \mathbb{C}^{\mathbb{N}}$ ist die *Fouriertransformierte* des Vektors $\mathbf{c} \in \mathbb{C}^{\mathbb{N}}$. Insbesondere finden wir

$$\mathbf{d} = \Omega_N \mathbf{c} \quad \Longleftrightarrow \quad \mathbf{c} = \overline{\Omega}_N \mathbf{d} \; .$$

Betrachten wir zum Beispiel das Polynom

$$p(x) = c_0 + c_1 x + \ldots + c_{N-1} x^{N-1} \quad \longleftrightarrow \quad \mathbf{c} = (c_0, \ldots, c_{N-1}) \in \mathbb{C}^{\mathbb{N}} \; .$$

Dann ergibt sich der Koeffizient d_k der Fouriertransformierten $\Omega_N \mathbf{c}$ als (normierte) Auswertung des Polynoms $p(x)$ an der Stelle $x = \omega^k$:

$$d_k \sqrt{N} \; = \; \sum_{j=0}^{N-1} \omega^{kj} c_j \; = \; p(\omega^k) \; = \; p(e^{2\pi \mathrm{i} k/N}) \; .$$

Die klassische Fouriertransformierte. Für das praktische Rechnen mit einer 2π-periodischen Funktion $f : \mathbb{R} \to \mathbb{C}$, unterteilt man das Intervall $[0, 2\pi]$ in N gleiche Intervalle mit linken Endpunkten

$$t_k = \frac{2\pi k}{N} \quad (\ k = 0, 1, \ldots, N-1)$$

und misst die Funktionswerte $f(t_k)$. Nun sucht man ein interpolierendes Polynom

$$p(x) = c_0 + c_1 x + \ldots + c_{N-1} x^{N-1} \quad \text{mit} \quad p(e^{\mathrm{i}t_k}) = f(t_k) \quad (\ k = 0, \ldots, N-1\ )$$

d.h. man löst das lineare Gleichungssystem in den Unbekannten $c_0, \ldots, c_{N-1}$:

$$1c_0 + e^{\mathrm{i}t_k} c_1 + e^{\mathrm{i}2t_k} c_2 + \ldots + e^{\mathrm{i}(N-1)t_k} c_{N-1} = f(t_k) \quad (\ k = 0, \ldots, N-1\ )\ .$$

Nach unseren Erkenntnissen über die diskrete Fouriertransformation erhalten wir sofort die Lösung als die inverse Fouriertransformierte $\mathbf{c} = \overline{\Omega}_N \mathbf{d}$ des Vektors

$$\mathbf{d} = \frac{1}{\sqrt{N}}(f(t_0), \ldots, f(t_{N-1}))\ .$$

d.h.

$$c_j = \frac{1}{N} \sum_{k=0}^{N-1} e^{-\mathrm{i}jt_k} f(t_k) \quad (\ j = 0, \ldots, N-1\ )\ .$$

Die $f(t)$ so interpolierende Funktion ist ein Element des Funktionenraums $\mathcal{F}$:

$$f_N(t) \ := \ \sum_{j=0}^{N-1} c_j e^{\mathrm{i}jt}\ .$$

Die sog. *klassischen Fourierkoeffizienten* der Funktion $f(t)$ sind die Parameter

$$\hat{f}(s) := \frac{1}{2\pi} \int_0^{2\pi} e^{-\mathrm{i}st} f(t) dt \quad (\ s \in \mathbb{Z}\ )\ .$$

BEMERKUNG. Die Funktion $s \mapsto \hat{f}(s)$ ist die *(klassische) Fouriertransformierte* von f.

Diskretisieren wir zur praktischen Berechnung das Integral mit $\Delta = 2\pi/N$ und den Stützstellen t_k , dann erhalten wir

$$\hat{f}(s) \ = \ \frac{1}{2\pi} \int_0^{2\pi} e^{-\mathrm{i}st} f(t) dt \ \approx \ \frac{1}{2\pi} \sum_{k=0}^{N-1} e^{-\mathrm{i}st_k} f(t_k) \Delta t \ = \ \frac{1}{N} \sum_{k=0}^{N-1} e^{-\mathrm{i}st_k} f(t_k) \ = \ c_s\ .$$

Die Interpolationsmethode und die Diskretisierung der klassischen Fourierkoeffizienten laufen auf dieselbe praktische Darstellung von $f(t)$ hinaus.

4. Die Quanten-Fouriertransformierte (QFT)

Nachdem wir die DFT im Koordinatenraum $\mathbb{C}^N$ eingeführt haben, ist es klar, wie man im allgemeinen Hilbertraum $\mathcal{H}$ der Dimension $\dim \mathcal{H} = N$ vorgehen wird: Wir wählen uns eine feste Bezugsbasis $B = \{b_0, b_1, \ldots, b_{N-1}\} \subseteq \mathcal{H}$ einer Koordinatisierung und stellen fest, dass das Basiselement $b_d \in B$ durch den Einheitsvektor $\mathbf{e}_d \in \mathbb{C}^N$ beschrieben wird.

Die diskrete Fouriertransformation liefert nun

$$\Omega_N \mathbf{e}_d = (1, \omega^d, \omega^{2d}, \ldots, \omega^{(N-1)d})^T$$

und induziert auf $\mathcal{H}$ die durch die Transformation der Basiselemente

$$DFT_N(b_d) \;:=\; \sum_{b_k \in B} \omega^{dk} b_k = \sum_{k=0}^{N-1} e^{2\pi \mathrm{i} dk/N} b_k \quad (\; d = 0, 1, \ldots, N-1 \;)$$

festgelegte unitäre Transformation $DTF_N : \mathcal{H} \to \mathcal{H}$.

Wir stellen uns n-Qubits als Elemente einer Hilbertraums mit Dimension $N = 2^n$ der Basis A^n aller reinen n-Qubits vor. Also erhalten wir nach dem obigen Vorgehen die *Quanten-Fouriertransformation* (QFT) als

$$QFT_N|d\rangle \;:=\; \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi \mathrm{i} dk/N} |k\rangle \quad (\; |d\rangle \in A^n \;)$$

und davon impliziert gemäss

$$\hat{f} \;=\; \Omega_N f$$

die allgemeine unitäre Transformation

$$QFT_N \;:\; \sum_{d=0}^{N-1} f_d |d\rangle \;\to\; \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \widehat{f_k} |k\rangle \quad \text{mit} \quad \widehat{f_k} = \frac{1}{\sqrt{N}} \sum_{d=0}^{N-1} e^{2\pi \mathrm{i} dk/N} f_d$$

ZUR ERINNERUNG: *Wir identifizieren eine natürliche Zahl* $0 \le d \le 2^n - 1$ *mit dem der Binärdarstellung entsprechenden reinen Qubit* $|d\rangle$*:*

$$d = d_0 2^{n-1} + d_1 2^{n-1} + d_2 2^{n-3} + \ldots d_{n-1} \quad \longleftrightarrow \quad |d\rangle = d_0 d_1 d_2 \ldots d_{n-1} \in A^n$$

Analog zu den Dezimalbrüchen benutzen wir auch die Darstellung mit Binärbrüchen:

$$0.d_0 d_1 \ldots d_k \;:=\; d_0 2^{-1} + d_1 2^{-2} + \ldots + d_k 2^{-(k+1)}$$

Insbesondere liefert der Fall $d = 0$ die Hadamardtransformation (gleichmässige Überlagerung der reinen n-Qubits):

$$QFT_N|0^n\rangle = H_n|0^n\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle \;.$$

Als unitäre Transformation erhält QFT_N natürlich auch allgemein n-Qubits.

4.1. Berechnung von QFT_N. Um einzusehen, dass sich die Quanten-Fouriertransformierte

$$QFT_{2^n}|d\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi \mathrm{i} dk/2^n} |k\rangle$$

eines reinen n-Bits $d \in A^n$ im vorliegenden Rechenmodell schnell finden lässt, betrachten wir d und ein beliebiges k in Binärdarstellung:

$$|d\rangle = d_0 d_1 \dots d_{n-1} \text{ (bzw. } d/2^n = 0.d_0 d_1 \dots, d_{n-1}) \,, \quad |k\rangle = k_0 k_1 \dots k_{n-1} \,.$$

Damit finden wir

$$\begin{array}{rcl} dk/2^n & = & k_0 \cdot (d_0 \dots d_{n-2}.d_{n-1}) \\ & + & k_1 \cdot (d_0 \dots d_{n-3}.d_{n-2}d_{n-1}) \\ & \vdots & \dots \\ & + & k_{n-2} \cdot (d_0.d_1 \dots d_{n-1}) \\ & + & k_{n-1} \cdot (0.d_0 d_1 \dots d_{n-1}) \,. \end{array}$$

Berücksichtigen wir noch die Identität $e^{2\pi \mathrm{i} m} = 1$ (für alle $m \in \mathbb{Z}$) und somit

$$e^{2\pi \mathrm{i}(d_0 d_1 \dots, d_{j-1}.d_j \dots d_{n-1})} = e^{2\pi \mathrm{i}(d_0 d_1 \dots, d_{j-1})} \cdot e^{2\pi \mathrm{i}(0.d_j \dots d_{n-1})} = e^{2\pi \mathrm{i}(0.d_j \dots d_{n-1})} \,,$$

so sehen wir sofort, dass sich $QFT_{2^n}|d\rangle$ als Verkettung von n 1-Qubits schreiben lässt:

$$\begin{array}{rcl} QFT_{2^n}|d\rangle & = & \frac{1}{\sqrt{2}}[|0\rangle + e^{2\pi \mathrm{i}(0.d_{n-1})}|1\rangle] \\ & \otimes & \frac{1}{\sqrt{2}}[|0\rangle + e^{2\pi \mathrm{i} 0.d_{n-2}d_{n-1}}|1\rangle] \\ & \vdots & \\ & \otimes & \frac{1}{\sqrt{2}}[|0\rangle + e^{2\pi \mathrm{i}(0.d_1 \dots d_{n-1})}|1\rangle] \\ & \otimes & \frac{1}{\sqrt{2}}[|0\rangle + e^{2\pi \mathrm{i}(0.d_0 d_1 \dots d_{n-1})}|1\rangle] \,. \end{array}$$

4.2. Implementierung von QFT_N. Um eine Implementierung der Quanten-Fouriertransformierten zu bewerkstelligen, wählen wir eine (Diagonal-)Matrix, die auf der zweiten Komponente eine (kontrollierte) Phasenverschiebung um $e^{2\pi \mathrm{i}/2^j}$ bewirkt:

$$V_j := \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi \mathrm{i}/2^j} \end{pmatrix}$$

Betrachten wir nun den Rechenvorgang im j-ten „Quantenregister“:

$$d_j \;\to\; \boxed{H} \;\to \boxed{V_1} \;\to \boxed{V_2} \;\to\; \dots \;\to \boxed{V_{n-(j+1)}} \;\to$$

Dabei nehmen wir an, dass die Phasenverschiebungen durch die übrigen Eingabebits d_{j+k}, mit $k = 1, \dots, n-(j+1)$, gesteuert sind:

- Die Recheneinheit V_k ist genau dann aktiviert, wenn $d_{j+k} = 1$.

Erinnert man sich an die Definition

$$0.d_j d_{j+1} \dots d_{n-1} \;=\; \frac{d_j}{2} + \frac{d_{j+1}}{2^2} + \dots + \frac{d_{n-1}}{2^{n-j}} \,,$$

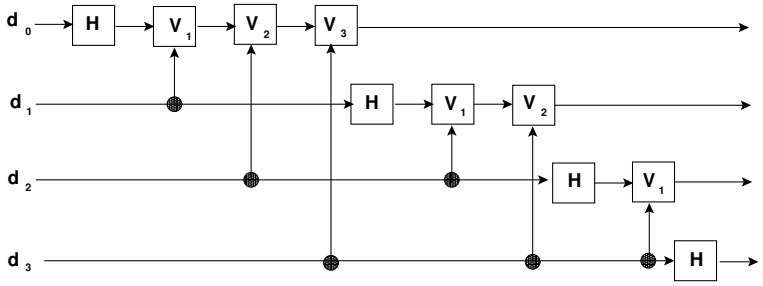


ABBILDUNG 1. Schaltkreis für QFT_{2^4}

so rechnet man nun ohne Mühe nach, dass die Eingabe von d_j das entsprechende Quantenregister in den Zustand

$$\frac{1}{\sqrt{2}}[|0\rangle + e^{2\pi \mathrm{i}(0.d_j d_{j+1} \ldots d_{n-1})}|1\rangle]$$

versetzt. Da $QFT_N|d\rangle$ eine Verkettung der Zustände der insgesamt n solchermassen angesetzten Quantenregister ist, können diese unabhängig voneinander „gelesen“ werden.

KAPITEL 5

Der Algorithmus von Shor

Das bisher meistversprechende Resultat der Quantenrechenmodells betrifft die Faktorisierung von natürlichen Zahlen, für die Shor im Quantenrechenmodell einen Algorithmus angegeben hat, der mit hoher Wahrscheinlichkeit einen Faktor findet (sofern ein solcher existiert).

Der Algorithmus ist sehr einfach. Etwas detailaufwendiger ist der Nachweis, dass der Algorithumus tatsächlich das Gewünschte leistet. Der Kern des Algorithmus löst folgendes Problem:

Gegeben zwei teilerfremde natürliche Zahlen $x, m \geq 2$, bestimme man die sog. *Ordnung* r von x bzgl. m, d.h. man berechne die natürliche Zahl

$$r := \min\{\ell \geq 1 \mid x^\ell \equiv 1 \bmod m\}$$

BEMERKUNG. Man kann (ohne Quantenrechnung) zeigen, dass eine „zufällig“ gewählte Zahl $1 < x < m$ mit hoher Wahrscheinlichkeit eine gerade Ordnung r hat mit der Eigenschaft:

$$x^{r/2} + 1 \not\equiv 0 \mod m$$

Aus der Produktzerlegung

$$(x^{r/2} + 1)(x^{r/2} - 1) \equiv x^r - 1 \equiv 0 \mod m$$

erkennt man dann, dass m mit einem der beiden Faktoren einen nichttrivialen gemeinsamen Teiler besitzt (der mit dem Euklidischen Algorithmus leicht berechnet werden kann). Wir konzentrieren uns deshalb hier allein auf die Berechnung der Ordnung r einer Zahl x.

Der Euklidische Algorithmus. Dass die Ordnung r von x wohldefiniert ist, erkennt man sofort am *Euklidischen Algorithmus*, der zu beliebigen gegebenen Zahlen $x, m \in \mathbb{N}$ ganze Zahlen a und b berechnet mit der Eigenschaft

$$ax + bm = \text{ggT}(x, m) \quad \text{(d.h. hier: } ax + bm = 1) \ .$$

Bew. Wir argumentieren per Induktion über $|x|$ und schreiben x als Vielfaches von m mit Rest:

$$x = km + q \quad (0 \leq q \leq m - 1) \quad \text{bzw.} \quad q = x - km \ .$$

Jeder gemeinsame Teiler von x und m muss auch q teilen. Also gilt

$$\text{ggT}(q, m) = \text{ggT}(x, m) \ .$$

Im Fall $q = 0$ haben wir ggT$(x, m) = m$ und dann trivialerweise

$$0x + 1m = m = \text{ggT}(x, m) \ .$$

Im Fall $q \geq 1$ folgt aus $q < x$ und $q = x - km$ per Induktion

$$\text{ggT}(x, m) = \text{ggT}(q, m) = a'q + b'm = a'x + (b' - k)m \; .$$

$\diamond$

Die Relation des Euklidischen Algorithmus können wir im Fall $\text{ggT}(x, m) = 1$ auch so schreiben:

$$ax \equiv 1 \bmod m \; .$$

Nun gibt es höchstens $m - 1$ verschiedene Reste modulo m. Bilden wir also die Potenzen $x, x^2, x^3, ...$ (immer modulo m), so muss es Zahlen $k < h$ geben mit der Eigenschaft

$$x^k \equiv x^h \bmod m \quad \text{d.h} \quad x^{h-k} \equiv x^h a^k \equiv x^k a^k \equiv 1 \bmod m \; .$$

Die Ordnung r von x existiert also. Im Prinzip kann sie nach spätestens $m - 2$ Schritten gefunden werden, indem man der Reihe nach die Potenzen bildet:

$$x^2, x^3, \ldots, x^{r-1}, x^r \quad (\mod m)$$

BEMERKUNG (INEFFIZIENZ DES POTENENZIERUNGSVERFAHRENS). Die einzelnen Potenzen x^k (modulo m) können zwar effizient berechnet werden. Dennoch ist das Verfahren *nicht* effizient im Sinne der Komplexitätstheorie, da die Anzahl der Iterationen möglicherweise die Grössenordnung von m erreichen und damit *exponentiell* bzgl. der Eingabegrösse (= Anzahl der Ziffern in der Bitdarstellung von x und m (etwa $\log_2 m$) anwachsen:

$$m = 2^{\log_2 m} \; .$$

1. Quantenberechnung der Ordnung r von x mod m

Es ist bekannt, dass Potenzen x^ℓ modulo m zu gegebenem $1 \leq x, \ell \leq m - 1$ mit einem Booleschen Schaltkreis effizient berechnet werden können. Also gibt es auch einen reversiblen Booleschen Schaltkreis, der die folgende Funktion (bzgl. n) effizient berechnet

$$(x, m, \ell, z) \quad \to \quad (x, m, \ell, z \oplus (x^\ell \bmod m)) \quad (x, m, \ell, z \in A^n) \; .$$

NOTATION: Da wir Potenzen x^ℓ im Folgenden immer nur modulo m berechnen, lassen wir von nun an der Term „mod m“ einfach weg.

Also gibt es auch einen entsprechenden effizienten Quantenschaltkreis (d.h. eine von n abhängende unitäre Transformation) $U = U_n$:

$$\boxed{|x\rangle \otimes |m\rangle \otimes |\ell\rangle \otimes |z\rangle \quad \xrightarrow{U} \quad |x\rangle \otimes |m\rangle \otimes |\ell\rangle \otimes |z \oplus x^\ell\rangle}$$

Bei Eingabe von $z = 0^n \in A^n$ können wir also auf den letzten n Stellen der Ausgabe die Binärdarstellung von x^ℓ (modulo m) ablesen:

$$\boxed{|x\rangle \otimes |m\rangle \otimes |\ell\rangle \otimes |0^n\rangle \quad \xrightarrow{U} \quad |x\rangle \otimes |m\rangle \otimes |\ell\rangle \otimes |x^\ell\rangle}$$

NOTATION: Im Folgenden sind x und m immer fest. Wir lassen deshalb der Einfachheit halber diese Symbole in der Notation fallen und schreiben nur

$$|\ell\rangle \otimes |z\rangle \quad \stackrel{U}{\longrightarrow} \quad |\ell\rangle \otimes |z \oplus x^\ell\rangle \quad \text{bzw.} \quad |\ell\rangle \otimes |0^n\rangle \quad \stackrel{U}{\longrightarrow} \quad |\ell\rangle \otimes |x^\ell\rangle \ .$$

Da der (Boolesche) Schaltkreis zur Berechnung der x^ℓ reversibel ist, steckt „im Prinzip“ die gesamte Information über die entsprechende Permutation

$$\sigma : A^{4n} \to A^{4n}$$

im Schaltkreis und damit insbesondere das Wissen um die Ordnung r von x. Die entscheidende Idee von Shor ist nun, diese Information durch Anwendung der Fouriertransformierten $QFT = QFT_{2^n}$ ablesbar zu machen.

Der Algorithmus geht also (in der abgekürzten Notation) folgendermassen vor:

$$\begin{array}{lll} |0^n\rangle \otimes |0^n\rangle & \stackrel{QFT}{\longrightarrow} & \dfrac{1}{2^{n/2}} \displaystyle\sum_{\ell=0}^{2^n-1} |\ell\rangle \otimes |0^n\rangle \\ & \stackrel{U}{\longrightarrow} & \dfrac{1}{2^{n/2}} \displaystyle\sum_{\ell=0}^{2^n-1} |\ell\rangle \otimes |x^\ell\rangle \\ & \stackrel{QFT}{\longrightarrow} & \dfrac{1}{2^n} \displaystyle\sum_{\ell=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{2\pi \mathrm{i} \ell k/2^n} |k\rangle \otimes |x^\ell\rangle \end{array}$$

Wir wollen nun zeigen, wie aus dem auf den vorletzten n Stellen der Ausgabebeinheit des Quantenschaltkreises „mit hoher Wahrscheinlichkeit“ die Ordnung r von x ermittelt werden kann.

2. Analyse des Quantenalgorithmus

Zur Analyse des Algorithmus schreiben wir wieder abkürzend $N := 2^n$ und unterteilen die Menge

$$\{0, 1, \ldots, N-1\} \quad \longleftrightarrow \quad A^n$$

nach Resten t modulo der gesuchten Ordnung r.

$$[t] := \{t + jr \mid t + jr \leq N-1\} \quad (t = 0, 1, \ldots, r-1) \ .$$

Denn wir stellen fest:

$$x^\ell \equiv x^{t+jr} \equiv x^t (x^r)^j \equiv x^t \quad \text{für alle} \ \ \ell \in [t] \ .$$

Die Anzahl ρ_t der Elemente der Äquivalenzklasse $[t]$ beträgt

$$\rho_t = 1 + \Big\lfloor \frac{N-(t+1)}{r} \Big\rfloor \quad \text{d.h.} \quad \rho_t \approx N/r \ .$$

2.1. Heuristische Analyse. Um die Idee des Algorithmus klar zu machen, nehmen wir der Einfachheit halber an, dass N/r eine ganze Zahl ist, d.h.

$$\rho := N/r = \rho_t \quad \text{für alle } \ t = 0, 1, \ldots, r-1 \ .$$

BEMERKUNG Die Annahme $N/r \in \mathbb{N}$ ist meist nicht erfüllt. Wir werden deshalb unten eine detailliertere Analyse vornehmen, die zeigt, dass die bei dieser Annahme auftretenden Fehler auf die Korrektheit des Rechenresultats des Quantenalgorithmus keinen Einfluss haben.

Wir setzen nun ein und rechnen

$$\begin{aligned}\frac{1}{2^n}\sum_{\ell=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{2\pi \mathrm{i}\ell k/2^n}|k\rangle \otimes |x^\ell\rangle &= \frac{1}{N}\sum_{t=0}^{r-1}\sum_{j=0}^{\rho-1}\sum_{k=0}^{N-1} e^{2\pi \mathrm{i}(t+jr)k/N}|k\rangle \otimes |x^t\rangle \\ &= \frac{1}{N}\sum_{t=0}^{r-1}\sum_{k=0}^{N-1} e^{2\pi \mathrm{i}tk/N}\sum_{j=0}^{\rho-1}(e^{2\pi \mathrm{i}k/\rho})^j|k\rangle \otimes |x^t\rangle\end{aligned}$$

Unter Berücksichtigung von

$$\sum_{j=0}^{\rho-1}(e^{2\pi \mathrm{i}k/\rho})^j = \begin{cases}\rho & \text{falls } k \text{ Vielfaches von } \rho \text{ (d.h. } \rho|k), \\ 0 & \text{andernfalls}\end{cases}$$

ergibt sich somit

$$\begin{aligned}\frac{1}{2^n}\sum_{\ell=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{2\pi \mathrm{i}\ell k/2^n}|k\rangle \otimes |x^\ell\rangle &= \frac{\rho}{N}\sum_{t=0}^{r-1}\sum_{\rho|k} e^{2\pi \mathrm{i}tk/N}|k\rangle \otimes |x^t\rangle \\ &= \frac{1}{r}\sum_{t=0}^{r-1}\sum_{\rho|k} e^{2\pi \mathrm{i}tk/N}|k\rangle \otimes |x^t\rangle\end{aligned}$$

Eine Beobachtung des Quantensystems liefert also jeden der Zustände

$$|k\rangle \otimes x^t \ , \quad \text{wobei } \ \rho|k \ \text{ und } \ 0 \le t \le r-1 \ ,$$

mit Wahrscheinlichkeit

$$\Big|\frac{e^{2\pi \mathrm{i}tk/N}}{r}\Big|^2 = \frac{1}{r^2} \ .$$

Insbesondere sehen wir mit Wahrscheinlichkeit $1/r$ das kleinste dieser k, nämlich

$$|\rho\rangle \ ,$$

und können daraus die Ordnung $r = N/\rho$ erschliessen. Ist r klein, so ist die Wahrscheinlichkeit gross, r auf diese Weise direkt ermitteln zu können.

Ist r gross, so können wir die Erfolgswahrscheinlichkeit deutlich erhöhen, wenn wir bei jedem beobachteten $|k\rangle$ kürzen:

$$k = q\rho = qN/r \quad \longleftrightarrow \quad k/N = q/r \ .$$

Man kann (mit der Eulerschen Φ-Funktion und ohne Quantentheorie) zeigen: Ist $1 \leq q \leq r-1$ zufällig gewählt, so ist die Wahrscheinlichkeit, dass q und r teilerfremd sind, mindestens von der Grössenordnung

$$\frac{1}{\log\log r}\ .$$

Mit mindestens dieser Wahrscheinlichkeit ergibt sich also r als eindeutiger Nenner nach der Kürzung. Wenn wir die Rechnung also K mal wiederholen, wird die Wahrscheinlichkeit, r *nicht* ermittelt zu haben, verschwindend klein:

$$(1 - 1/\log\log r)^K \quad \longrightarrow \quad 0.$$

BEMERKUNG (DIE EULERSCHE Φ-FUNKTION). Die Eulersche Φ-Funktion ist definiert als

$$\Phi(r) := \{1 \leq k \leq n-1 \mid \mathrm{ggT}(k,r) = 1\}\ .$$

Aus der elementaren Zahlentheorie weiss man, dass eine Konstante $\delta > 0$ existisiert mit der Eigenschaft

$$\frac{\Phi(r)}{r} > \frac{\delta}{\log\log r} \quad \text{(wenn } r \text{ genügend gross)}\ .$$

2.2. Detaillierte Analyse. Die Wahrscheinlichkeit $p(k,t)$, am Ende der Rechnung einen bestimmten Zustand $|k\rangle \otimes |x^t\rangle$ zu beobachten, ist

$$\begin{aligned} p(k,t) &= \Big|\frac{1}{N}\sum_{j=0}^{\rho_t-1} e^{2\pi \mathrm{i}(t+jr)k/N}\Big|^2 \;=\; \frac{|e^{2\pi \mathrm{i}t/N}|}{N^2}\cdot\Big|\sum_{j=0}^{\rho_t-1} e^{2\pi \mathrm{i}jrk/N}\Big|^2 \\ &= \frac{1}{N^2}\Big|\sum_{j=0}^{\rho_t-1} e^{2\pi \mathrm{i}jrk/N}\Big|^2\ . \end{aligned}$$

Schreiben wir rk in der Form

$$rk = hN + c \quad \text{mit} - N/2 \leq c \leq N/2\ ,$$

so erhalten wir (wegen $e^{2\pi \mathrm{i}hN} = 1$):

$$p(k,t) = \frac{1}{N^2}\Big|\sum_{j=0}^{\rho_t-1} e^{2\pi \mathrm{i}jc/N}\Big|^2$$

Wir wollen zuerst zeigen:

$$\boxed{|c| \leq r/2 \quad \Longrightarrow \quad p(k,t) > \frac{1}{4r^2}}$$

Für $c = 0$ ist das klar. Denn:

$$\Big|\frac{1}{N}\sum_{j=0}^{\rho_t-1} e^0\Big| = \frac{\rho_t^2}{N^2} > \frac{1}{4r^2}\ .$$

Sei also oBdA also $0 < c \leq r/2$ (der Fall $-r/2 \leq c < 0$ ist vollkommen analog). Wir setzen zur Abkürzung

$$\zeta_c := e^{2\pi \mathrm{i} jc/N}$$

und erinnern an die Summationsformel

$$\sum_{j=0}^{\rho_t - 1} \zeta_c^j \;=\; \frac{1-\zeta_c^{\rho_t}}{1-\zeta_c} \;.$$

Im Fall $2\pi(\rho_t - 1)c/N \leq \pi/2$ liegen alle ζ_c^j auf dem Einheitskreisbogen im ersten Quadranten des $\mathbb{R}^2$. Die Vektoren

$$\zeta_c^j + \zeta_c^{(\rho_t-1)-j} \quad j = 0, 1, \ldots$$

deuten alle in dieselbe Richtung ihrer Winkelhalbierenden

$$e^{2\pi \mathrm{i}(\rho_t - 1)/2N}$$

und haben (nach Pythagoras) mindestens Länge $\sqrt{2}$. Also hat der Summenvektor eine Länge von mindestes

$$\Big|\sum_{j=0}^{\rho_t - 1} \zeta_c^j\Big| \;\geq\; \sqrt{2}\lfloor \rho_t/2 \rfloor \;.$$

Daraus folgt

$$p(k,t) \;=\; \frac{1}{N^2}\Big|\sum_{j=0}^{\rho_t - 1} \zeta_c^j\Big|^2 \;>\; \frac{1}{4r^2} \;.$$

Im Fall $2\pi(\rho_t - 1)c/N > \pi/2$ liegt $\zeta_c^{\rho_t}$ (wegen $c \leq r/2$) auf dem Einheitskreisbogen im zweiten Quadraten des $\mathbb{R}^2$. Deshalb gilt

$$|1 - \zeta_c^{\rho_t}| > |1 - e^{\mathrm{i}\pi/2}| = \sqrt{2} \quad \text{und} \quad |1 - \zeta_c| \leq 2\pi c/N \leq \pi r/N$$

und folglich

$$\Big|\sum_{j=0}^{\rho_t - 1} \zeta_c^j\Big| \;\geq\; \frac{\sqrt{2}N}{\pi r}$$

d.h.

$$p(k,t) \;=\; \frac{1}{N^2}\Big|\sum_{j=0}^{\rho_t - 1} \zeta_c^j\Big|^2 \;\geq\; \frac{2}{\pi^2 r^2} \;>\; \frac{1}{4r^2} \;.$$

Die Schranke $p(k,t) > 1/4r^2$ haben wir nun für den Fall geleitet, wo das aus k (modulo N) errechnete c die Eigenschaft hat:

$$-r/2 \leq c \leq r/2 \;.$$

Das Erfülltsein dieser Ungleichung ist äquivalent mit der Existenz einer Zahl d derart, dass

$$-r/2 \leq rc - dN \leq r/2 \quad \text{bzw.} \quad \Big|\frac{c}{N} - \frac{d}{r}\Big| \leq \frac{1}{2N} \;,$$

(wie man nach Division durch rN sieht). In dieser Form sieht man leicht, dass es zu jedem $0 \leq d \leq r-1$ ein c gibt, sodass die Ungleichung gilt: Wenn wir das Intervall $[0,1]$ mit Intervallen der Länge $1/N$ überdecken, so fällt d/r in eines dieser Intervalle und hat dann zu einem der Endpunkte c/N maximal den Abstand $1/2N$ der halben Intervalllänge.

Wählen wir ausserdem den Dimensionsparameter n so gross, dass gilt

$$r^2 \leq m^2 < 2^n = N \quad (\text{z.B. } n = 1 + 2\lceil \log m + 1 \rceil),$$

dann gibt es zu jedem c auch *höchstens* ein d, das die Ungleichung erfüllt. Daraus folgt:

- *Wir können den Bruch d/r berechnen, indem wir den Bruch c/N bestmöglich durch einen Bruch mit Nenner $< N$ approximieren.*

BEMERKUNG. Diese Approximation kann über die Kettenbruchentwicklung von c/N leicht berechnet werden, auf die wir hier nicht weiter eingehen wollen.

Wenn wir also d/r gefunden haben, und d zu r teilerfremd ist (d.h. der Bruch kann nicht weiter gekürzt werden), dann ist die Ordnung r als Nenner des Bruchs eindeutig bestimmt. Wir wollen nun noch abschätzen, mit welcher Wahrscheinlichkeit dieser günstige Fall eintritt.

Es gibt r mögliche Werte für x^t, da x Ordnung r hat. Also gibt es $r\Phi(r)$ viele Zustände

$$|k\rangle \otimes |x^t\rangle \ ,$$

aus denen r ermittelt werden könnte. Jeder dieser Zustände tritt mit Wahrscheinlichkeit mindestesn $1/4r^2$ auf. Also ist die Wahrscheinlichkeit für die erfolgreiche Bestimmung von r mindestens

$$\frac{\Phi(r)}{4r} \ \sim \ \frac{\delta}{\log\log r} \ .$$