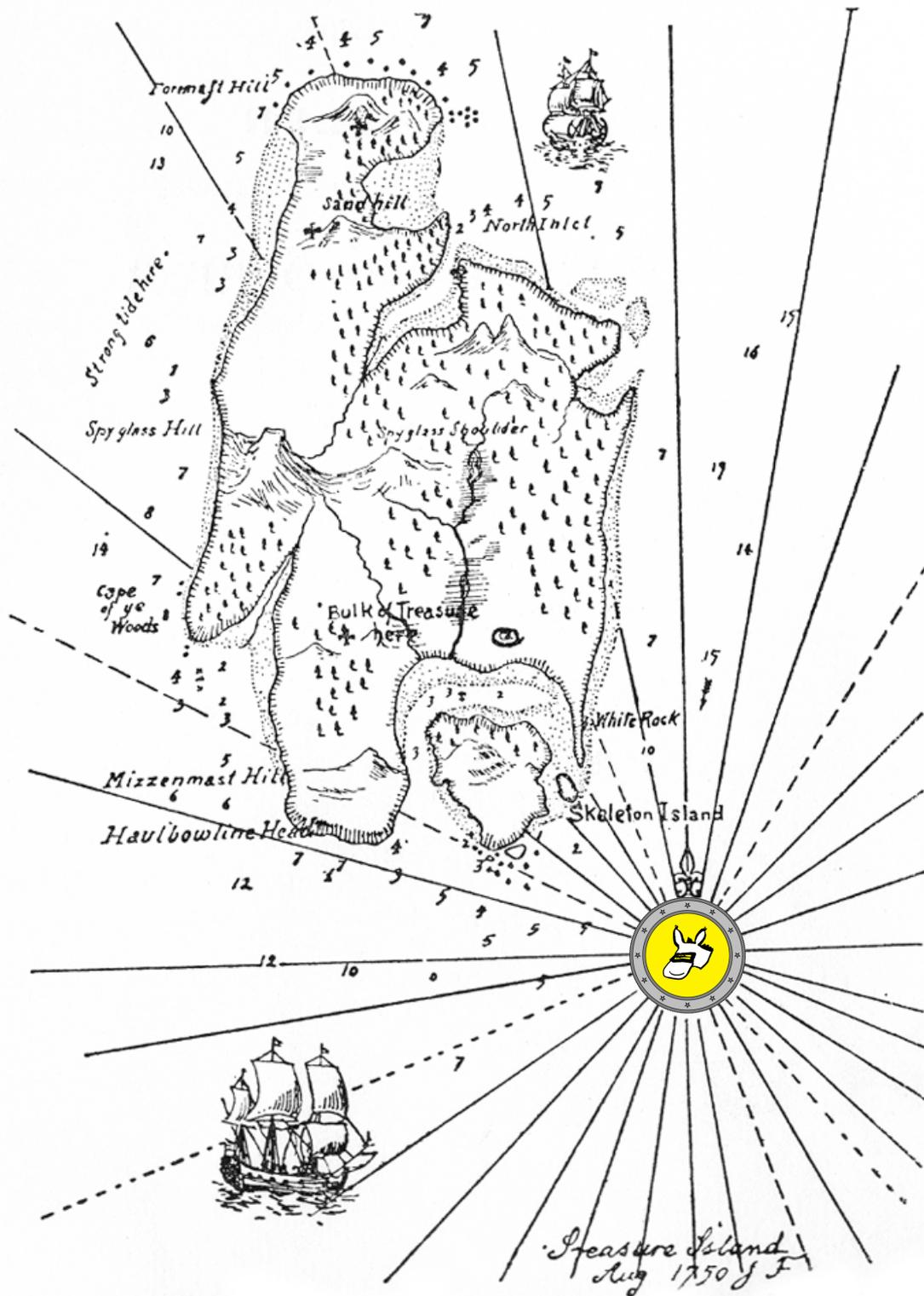


# The Quantum Quest



# Willkommen beim *Quantum Quest*!

Das Ziel dieses fünfwöchigen Abenteuers ist es, dir die Grundlagen der Quanteninformatik beizubringen. Am Ende der Reise wirst du verstehen, was Quantenbits und Quantenalgorithmen sind und wofür sie gut sind. Auf dem Weg wirst du Alice und Bob kennen lernen, die im Jahr 2058 leben und (genau wie du) an Quantencomputern basteln, wenn die Schule vorbei ist. Heutzutage sind Wissenschaftler:innen immer noch dabei, echte Quantencomputer in ihren Laboren zu bauen, aber im Jahr 2058 gibt es Quantencomputer überall – sogar in deiner Hosentasche! Leider gibt es auch Menschen, die diese Technologie für böartige Zwecke nutzen wollen, also musst du deinen beiden Freund:innen – Alice und Bob – dabei helfen, sich vor der bösen Hackerin Eve zu beschützen. Viel Erfolg auf deinem Abenteuer!

Mit herzlichen Quantengrüßen,  
Maris Ozols & Michael Walter

---

## Leitfaden für Lesende

In diesen Vorlesungsnotizen findest du zahlreiche **Übungsaufgaben** (in grünen Kästen) sowie **Hausaufgaben** (in roten Kästen). Die **Übungsaufgaben** sind dazu gedacht, dein Wissen und Verständnis zu testen, während du liest. Lösungen für sie bekommst du immer am Ende einer Quest. Die **Hausaufgaben** dagegen sind dazu gedacht, am Ende jeder Woche abgegeben zu werden. Manche dieser Aufgaben sind als “optional” markiert – sie sind nicht notwendig um dem Kurs folgen zu können, aber bieten eine gute Möglichkeit weiter zu üben. Manche sind sogar als “herausfordernd” markiert – diese sind ein bisschen schwieriger als die anderen!

## Danksagung

Wir möchten uns bei den folgenden Personen bedanken, die uns helfen, diesen Online-Kurs anzubieten: Doutzen Abma, Maya Bach, Valerie Bettaque, Amalia Böttger, Milo Camardese, Arjan Cornelissen, Bas Dirkse, Oliver Dorogi, Jari Egbers, Yassine Ferjani, Marten Folkertsma, Koen Groenland, Galina Pass, Philip Verduyn Lunel, Anurudh Peduri, Simon Schmidt, Quinten Tupker, Mees de Vries, Jordi Weggemans, Peter Ypma. Vielen Dank an Maya Bach für die deutschsprachige Übersetzung dieses Texts. Wir sind auch Craig Gidney äußerst dankbar, dessen Quantensimulator **QUIRK** wir angepasst haben, um **QUIRKY** zu bauen. Zu guter Letzt wollen wir uns bei euch, all den enthusiastischen Schüler:innen bedanken, die an diesem Kurs teilgenommen haben!

# Der Quantum Quest

Maris Ozols und Michael Walter

November 2023

## Inhaltsverzeichnis

<b>Der Quantum Quest</b>	<b>1</b>
<b>1 Quest 1: Maestro der Wahrscheinlichkeit</b>	<b>3</b>
1.1 Probabilistische Bits . . . . .	3
1.1.1 Multiplizieren von Wahrscheinlichkeiten . . . . .	5
1.1.2 Wahrscheinlichkeiten addieren . . . . .	6
1.1.3 Wahrscheinlichkeiten und Berechnungen . . . . .	6
1.2 Operationen auf probabilistischen Bits . . . . .	7
1.2.1 Erweitern durch Linearität . . . . .	8
1.2.2 Zufällige Operationen . . . . .	9
1.3 Ein probabilistisches Bit messen . . . . .	11
1.4 Der QUIRKY Simulator . . . . .	13
1.4.1 Erste Schritte . . . . .	14
1.4.2 Eigene Operationen erstellen . . . . .	15
1.4.3 Eine mysteriöse Operation . . . . .	16
1.5 Lösungen der Übungsaufgaben . . . . .	17
<b>2 Quest 2: Das Qubit bezwingen</b>	<b>21</b>
2.1 Quantenbits . . . . .	21
2.1.1 Wahrscheinlichkeiten versus Amplituden . . . . .	21
2.1.2 Ein Qubit als ein Kreis . . . . .	22
2.2 Ein Qubit messen . . . . .	23
2.3 Qubits mit QUIRKY simulieren . . . . .	25
2.4 Operationen auf einem Qubit . . . . .	26
2.4.1 Rotationen . . . . .	28
2.4.2 Zusammensetzen von Quanten-Operationen . . . . .	30
2.4.3 Spiegelungen . . . . .	32
2.5 Quantenzustände unterscheiden . . . . .	32
2.5.1 Noch eine mysteriöse Operation . . . . .	35
2.6 Exkurs zur Physik (optional) . . . . .	36
2.6.1 Interferenz . . . . .	36
2.6.2 Polarisation . . . . .	38
2.7 Lösungen der Übungsaufgaben . . . . .	40

<b>3</b>	<b>Quest 3: Verzaubernde Verschränkungen</b>	<b>43</b>
3.1	Zwei probabilistische Bits	43
3.1.1	Beide Bits messen	44
3.1.2	Lokale Operationen	45
3.1.3	Nur ein Bit messen	47
3.1.4	Der Zustand des anderen Bit	48
3.1.5	Die SWAP-Operation	50
3.1.6	Die kontrollierte-NOT-Operation	51
3.1.7	Produkt-Verteilungen	52
3.1.8	Korrelierte Verteilungen	54
3.2	Zwei Quantenbits	56
3.2.1	Zwei Qubits messen	58
3.2.2	Lokale Operationen	58
3.2.3	Parallele Operationen	60
3.2.4	Kontrollierte Operationen	63
3.2.5	Verschränkte Zustände	64
3.2.6	Verschränkung und Korrelationen	66
3.2.7	Die Macht von Verschränkung	67
3.3	Lösungen der Übungsaufgaben	71
<b>4</b>	<b>Quest 4: Quantenkompositionen</b>	<b>77</b>
4.1	Quantenschaltungen	77
4.1.1	Viele Quantenbits	77
4.1.2	Operationen	79
4.1.3	Die allgemeinsten Quantenoperationen	81
4.1.4	Regeln für Schaltungen	82
4.1.5	Alle Qubits messen	83
4.1.6	Einzelne Qubits messen	83
4.2	Quanten-Überraschungen	86
4.2.1	Unklonbarkeit	86
4.2.2	One-Time-Pad	87
4.2.3	Quanten-Teleportation	89
4.2.4	Ein Blick auf Quantennetzwerke	93
4.2.5	Die Unschärferelation	94
4.3	Lösungen der Übungsaufgaben	97
<b>5</b>	<b>Quest 5: Virtuose Algorithmen</b>	<b>101</b>
5.1	Mit Quantenorakeln sprechen	102
5.1.1	Umkehrbare Berechnungen	103
5.1.2	Bit-Orakel	104
5.1.3	Phasen-Orakel	106
5.2	Quantenalgorithmen	108
5.2.1	Der Algorithmus von Deutsch	108
5.2.2	Die Hadamard-Transformation und Interferenz	111
5.2.3	Der Deutsch-Jozsa-Algorithmus	114
5.2.4	Bernstein-Vazirani-Algorithmus	115
5.3	Suchen mit Grover	117
5.3.1	Winkelverstärkung	119
5.4	Deine Quanten-Reise	120
5.5	Lösungen der Übungsaufgaben	121

## Quest 1: Maestro der Wahrscheinlichkeit

Willkommen in der ersten Woche des *Quantum Quest* – du stehst am Anfang eines spannenden Abenteuers!

Die Quanteninformatik ist ein faszinierendes Thema, das unsere Fantasie anregt. Diese Faszination kommt hauptsächlich von den seltsamen Regeln, denen die Quantenwelt zu folgen scheint. Da die Regeln uns aber so fremd sind und der Zustandsraum eines Quantencomputers exponentiell groß ist, kannst du dich leicht verirren. Um das zu verhindern, musst du dich zunächst mit der Welt der Wahrscheinlichkeiten vertraut machen. Sobald du dann ein Maestro der Wahrscheinlichkeit bist, wird sich die Tür der Quantenwelt für dich öffnen!

Das Ziel der ersten Quest ist es, dir Wahrscheinlichkeiten und probabilistische Bits näher zu bringen: Was ist ein probabilistisches Bit, in was für Zuständen kann es sich befinden, was für Operationen können wir anwenden, und wie können wir Informationen extrahieren, indem wir ein solches Bit “messen”? In der zweiten Quest wird es dann um Quantenbits gehen, welche sich sehr ähnlich zu probabilistischen Bits verhalten.

### 1.1 Probabilistische Bits

Wir benutzen **Wahrscheinlichkeiten**, wann immer eines von mehreren möglichen Ereignissen eintreten kann, wir aber noch nicht wissen, welches Ereignis tatsächlich eintreten wird – je wahrscheinlicher das Ereignis, desto höher dessen Wahrscheinlichkeit. Ein Ereignis welches auf jeden Fall eintritt, hat eine Wahrscheinlichkeit von 1 (also 100%), während ein Ereignis, das niemals eintritt, eine Wahrscheinlichkeit von 0 hat.

Wir können uns beispielsweise einen Münzwurf vorstellen. Wenn du eine Münze wirfst und sie, ohne sie anzuschauen, mit deiner Hand abdeckst, können zwei mögliche Ereignisse eintreten: entweder die “Kopf”-Seite zeigt nach oben, oder die “Zahl”-Seite zeigt nach oben. Bei einer *fairen* Münze sind beide Fälle gleich wahrscheinlich, also weisen wir beiden eine Wahrscheinlichkeit von  $\frac{1}{2} = 0.50$ , also 50%, zu (in Abb. 1.1 stellen wir diesen Fall als 🍀 dar). Münzen können aber auch *unfair* (engl. *biased*) sein, also mit höherer Wahrscheinlichkeit auf einer der beiden Seiten landen. Je nachdem, wie unfair die Münze ist, man sich ein ganzes Spektrum an Möglichkeiten vorstellen: eine extrem unfaire Münze könnte immer mit der “Kopf”-Seite nach oben landen, eine andere dagegen vielleicht immer mit der “Zahl”-Seite (siehe 🍀 und 🍁 links und rechts in Abb. 1.1). Für die erste dieser beiden Münzen ist die Wahrscheinlichkeit des Ereignisses “Zahl” dabei 0 (da es immer mit der “Kopf”-Seite nach oben landet), während für die andere Münze das Ereignis “Zahl” die Wahrscheinlichkeit 1 hat.

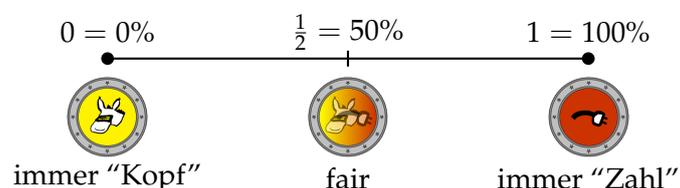


Abbildung 1.1: Ein probabilistisches Bit, das den Zustand einer zufälligen Münze beschreibt. Die “Kopf”-Seite ist mit einem Eselskopf beschriftet und die “Zahl”-Seite mit einem Eselschwanz. (Wieso das? Anstatt von “Kopf und Zahl” spricht man im Englischen von “heads or tails”...)

Da wir uns keine Gedanken über die genaue Konstruktion von Münzen von verschiedener Form und Größe machen wollen, ist es sinnvoll, die Information, die einen Münzwurf beschreibt, zu abstrahieren. Das erreichen wir, indem wir die Ereignisse “Kopf” und “Zahl” mit



den Werten 0 und 1 eines Bits assoziieren. Dann lässt sich ein Münzwurf mit zwei Wahrscheinlichkeiten beschreiben: Der Wahrscheinlichkeit  $p_0$ , dass das Ergebnis 0 (“Kopf”) ist, und der Wahrscheinlichkeit  $p_1$ , dass das Ergebnis 1 (“Zahl”) ist. Ein solches Bit, welches seine beiden möglichen Werte mit bestimmten Wahrscheinlichkeiten annimmt, wird als **probabilistisches Bit** bezeichnet. Beachte dabei, dass sich die beiden Wahrscheinlichkeiten  $p_0, p_1 \geq 0$  notwendigerweise zu 1 aufaddieren müssen:  $p_0 + p_1 = 1$ . Wenn die Münze, wie im Beispiel oben, fair ist, dann muss, wie oben beschrieben,  $p_0 = p_1 = \frac{1}{2}$  gelten.

Ist dir aufgefallen, dass es ausreichen würde, nur eine der beiden Wahrscheinlichkeiten  $p_0$  oder  $p_1$  anzugeben, da beide durch die jeweils andere bestimmt werden? Der Klarheit halber werden wir aber immer beide angeben und die Wahrscheinlichkeiten als **Vektor** schreiben:

$$p = \begin{pmatrix} p_0 \\ p_1 \end{pmatrix}. \quad (1.1)$$

Wir nennen diesen Vektor die **Wahrscheinlichkeitsverteilung** oder den **Zustand** des probabilistischen Bits. Diese Vektornotation ist nicht nur praktisch, um alle Wahrscheinlichkeiten in einer schönen Tabelle darzustellen, sondern sie erlaubt es uns auch, ein probabilistisches Bit geometrisch zu visualisieren. Außerdem wird es uns helfen, Parallelen zwischen probabilistischen Bits und Quantenbits zu sehen.

Wir nennen die beiden Zustände, bei denen das Ergebnis immer 0 (“Kopf”, ) oder immer 1 (“Zahl”, ) lautet, **deterministisch**. Dies entspricht den oben diskutierten “extrem unfairen” Münzen, bei denen das Ergebnis des Münzwurfs von vornherein eindeutig bestimmt ist. In Gl. (1.1) korrespondieren diese Zustände zu den Wahrscheinlichkeitsverteilungen mit  $p_0 = 1, p_1 = 0$  bzw.  $p_0 = 0, p_1 = 1$ . Da wir diese Zustände häufig benutzen, ist es sinnvoll, folgende Abkürzung einzuführen:

$$[0] = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad [1] = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (1.2)$$

Diese Notation mag am Anfang ein wenig verwirren, aber du kannst dir  $[0]$  und  $[1]$  als ) und ) oder als [Kopf] und [Zahl] vorstellen, wenn du möchtest.

Diese zwei Zustände bilden eine **Basis** aller Zustände. Das bedeutet, dass wir alle anderen Zustände als **Linearkombination** dieser beiden Zustände darstellen. Konkret:

$$\begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = p_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + p_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = p_0[0] + p_1[1]. \quad (1.3)$$

Da Zustände Vektoren sind, können wir sie in einem zweidimensionalen Koordinatensystem visualisieren. Die möglichen Zustände eines probabilistischen Bits entsprechen dann genau der Strecke zwischen den beiden Punkten  $[0]$  und  $[1]$ , die den deterministischen Zuständen des Bits entsprechen (siehe Abb. 1.2).

### Übungsaufgabe 1.1: Die blaue Strecke verstehen

Laut Abb. 1.2 bilden die möglichen Zustände eines probabilistischen Bits eine Strecke. Nimm dir etwas Zeit, um darüber nachzudenken, und versuche, folgende Fragen zu beantworten:

1. Warum liegen alle möglichen Zustände auf einer Strecke?
2. Warum hört die Strecke an den Koordinatenachsen auf und geht nicht weiter?
3. Welcher Punkt auf der Strecke entspricht einer fairen Münze?

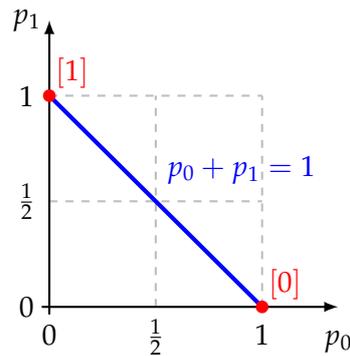


Abbildung 1.2: Die blaue Strecke entspricht den Zuständen eines probabilistischen Bits.

### 1.1.1 Multiplizieren von Wahrscheinlichkeiten

Wenn du zwei Münzen wirfst, wie hoch ist dann die Wahrscheinlichkeit, dass beide “Kopf” ergeben? Nimm an, dass die beiden Münzen durch probabilistischen Bits

$$a = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \quad (1.4)$$

beschrieben werden, wobei das Ergebnis 0 “Kopf” entspricht und das Ergebnis 1 “Zahl”. Die Wahrscheinlichkeit, dass die erste Münze die “Kopf”-Seite zeigt, ist also  $a_0$ , und die Wahrscheinlichkeit, dass die zweite Münze die “Kopf”-Seite zeigt, ist  $b_0$ . (Wir nehmen nicht an, dass die Münzen fair sind, diese Wahrscheinlichkeiten müssen also nicht 50% sein.) Um die Wahrscheinlichkeit, dass beide Münzen *gleichzeitig* “Kopf” zeigen, zu berechnen, müssen wir die beiden Wahrscheinlichkeiten miteinander *multiplizieren*:

$$p_{00} = a_0 b_0. \quad (1.5)$$

Bemerke, dass  $p_{00} \leq a_0$  und  $p_{00} \leq b_0$  da  $a_0 \leq 1$  und  $b_0 \leq 1$ . Das ist logisch, da “Kopf” für beide Münzen gleichzeitig nicht wahrscheinlicher sein kann als “Kopf” für eine der beiden Münzen alleine (meistens ist es sogar weniger wahrscheinlich). Nach dem gleichen Muster kann man die Wahrscheinlichkeiten aller anderer Kombinationen von Kopf und Zahl berechnen. Hier fassen wir die Fälle zusammen:

$$\begin{aligned} p_{00} &= a_0 b_0, & p_{01} &= a_0 b_1, \\ p_{10} &= a_1 b_0, & p_{11} &= a_1 b_1. \end{aligned} \quad (1.6)$$

Wir nennen zwei Ereignisse **unabhängig** (engl. *independent*), wenn sie aus verschiedenen Quellen stammen und das Eintreten des einen Ereignisses nichts über das andere aussagt. In solchen Situationen wird typischerweise das Wort “und” benutzt. Zum Beispiel: “Die erste Münze ist Kopf *und* die zweite ist Zahl”. Wir multiplizieren Wahrscheinlichkeiten, wenn wir wissen wollen, ob zwei unabhängige Ereignisse gleichzeitig eingetreten sind.

#### Übungsaufgabe 1.2: Wahrscheinlichkeiten multiplizieren

Alice sitzt gelangweilt im Mathematikunterricht und schaut auf ihre digitale Armbanduhr. Der Sekundenzeiger zeigt Werte von 00 bis 59. Nimm an, dass Alice zu einem zufälligen Zeitpunkt innerhalb der nächsten Minute auf ihre Uhr schaut.

1. Mit welcher Wahrscheinlichkeit sieht sie 00?
2. Mit welcher Wahrscheinlichkeit ist die letzte Ziffer 0?

3. Mit welcher Wahrscheinlichkeit ist die erste Ziffer 0?
4. Begründe, wieso die Werte beider Ziffern unabhängig voneinander sind. Überprüfe deine Antwort auf Frage 1, indem du deine Antworten auf Fragen 2 und 3 miteinander multiplizierst.

### 1.1.2 Wahrscheinlichkeiten addieren

Jetzt schauen wir uns ein etwas komplizierteres Problem an. Nimm wieder an, dass zwei Münzen  $a$  und  $b$  geworfen werden. Wie hoch ist nun die Wahrscheinlichkeit, dass beide das gleiche Ergebnis zeigen? Es gibt es zwei Möglichkeiten – entweder sind beide Münzen mit der “Kopf”-Seite nach oben gelandet, oder beide zeigen “Zahl”. Wir wissen schon von Gl. (1.6), dass die Wahrscheinlichkeiten dieser beiden einzelnen Ereignisse die folgenden sind:

$$p_{00} = a_0 b_0, \quad p_{11} = a_1 b_1.$$

Um die Wahrscheinlichkeit zu bestimmen, dass das eines der beiden Ereignisse eintritt, müssen wir die beiden Wahrscheinlichkeiten *addieren*:

$$p_{00} + p_{11} = a_0 b_0 + a_1 b_1. \quad (1.7)$$

Wir addieren Wahrscheinlichkeiten, wenn wir mehrere Ergebnisse eines Experimentes kombinieren wollen. Solche kombinierten Ereignisse werden typischerweise mit dem Wort “oder” beschrieben. Zum Beispiel: “Beide Münzen zeigen Kopf *oder* beide Münzen zeigen Zahl”. Aber aufgepasst: Das funktioniert nur, wenn sich die Münzen nicht gegenseitig beeinflussen!

#### Übungsaufgabe 1.3: Wahrscheinlichkeiten addieren

Bob sitzt ebenfalls gelangweilt im Mathematikunterricht. Ihm fällt auf, dass Alice auf ihre Uhr starrt – also wirft er auch einen Blick auf seine Uhr. Zu seiner Überraschung zeigt der Sekundenzeiger 44, was Bob unwahrscheinlich erscheint. Wie hoch ist die Wahrscheinlichkeit, dass beide Ziffern des Sekundenzeigers dieselben sind, wenn Bob zu einem zufälligen Zeitpunkt einer Minute auf seine Uhr schaut?

Nun hast du gelernt, wann Wahrscheinlichkeiten addiert oder multipliziert werden. Probiere doch mit diesem Wissen, deine erste Hausaufgabe zu lösen!

#### Hausaufgabe 1.1: Gegenteilige Münzen

Alice wirft zwei Münzen  $a$  und  $b$  mit den folgenden Wahrscheinlichkeitsverteilungen:

$$a = \begin{pmatrix} 2/3 \\ 1/3 \end{pmatrix}, \quad b = \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix}.$$

Mit welcher Wahrscheinlichkeit landen die beiden Münzen auf *unterschiedlichen* Seiten?

### 1.1.3 Wahrscheinlichkeiten und Berechnungen

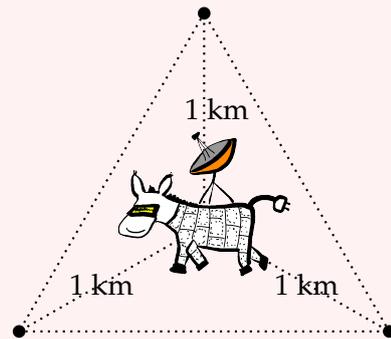
Sind probabilistische Bits für Berechnungen überhaupt von Nutzen? Auf den ersten Blick mag es so aussehen, als seien sie es nicht, da die Werte 0 und 1 eines normalen bits definitives Wissen darstellen, während ein probabilistisches bit ja nur *ungefähres* Wissen (also ein *Mangel* an Wissen) repräsentiert. Warum sollte ich also Speicherplatz auf meinem Computer verschwenden, um probabilistische bits zu speichern, die nur den Mangel an Information widerspiegeln, wenn ich doch einfach die Information speichern könnte, die ich habe, selbst wenn sie unvollständig ist. Der Vorteil von probabilistischen Bits ist, dass sie Teilwissen genauer darstellen können –

wenn du etwas nicht weißt, ist es besser das zuzugeben und zufällig zu raten, als so zu tun, als wüsstest du die Antwort sicher. Im folgenden Problem wird das noch einmal anschaulich dargestellt. Hier muss Alice' Eselsroboter eine Entscheidung treffen muss, ohne alle Informationen zu haben.

### Hausaufgabe 1.2: Alice' Esel

**Das Problem:** Alice will ihren Eselsroboter so programmieren, dass er alleine zu einer Ladestation laufen kann, um sich dort aufzuladen. Es gibt drei Ladestationen in der Nähe. Sie sind alle jeweils 1 km vom Roboter entfernt und bilden ein gleichseitiges Dreieck – mit dem Esel in der Mitte. Der Roboter hat noch genügend Akku, um 2,8 km zu laufen.

Alice wird das Programm, welches den Roboter die Ladestation aussuchen lässt, per WiFi auf ihn hochladen. Sie weiß aber, dass ihre böse Klassenkameradin Eve sie sabotieren will. Da Eve das WiFi überwacht, kann sie den Programmcode mitlesen. Damit Eve nicht sehen kann, wohin der Esel läuft, schaltet Alice das WiFi des Roboters aus, sobald das Programm hochgeladen ist. Solange der Esel dann läuft, kann Eve ihn also nur sabotieren, indem sie Ladestationen hackt und deaktiviert. Sie kann allerdings nur zwei der drei Ladestationen deaktivieren, bevor sie entdeckt wird. Da Eve nicht sehen kann, wohin der Roboter läuft, muss sie sich also nur auf Grund von Alice' Programm entscheiden, welche zwei Ladestationen sie deaktivieren will.

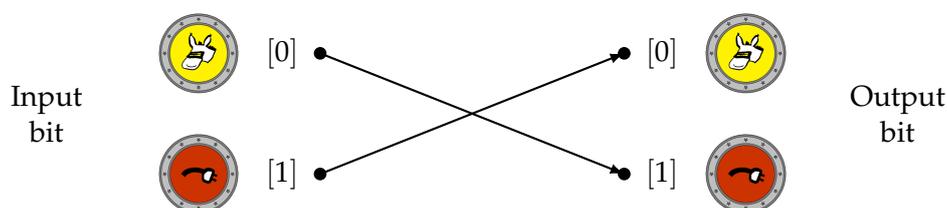


#### Fragen:

1. Wie viele Ladestationen kann der Esel besuchen, bevor seine Batterie leer ist?
2. Nimm an, dass Alice ihren Roboter so programmiert, dass er die Ladestationen in einer festgelegten Reihenfolge besucht. Kann Eve verhindern, dass der Roboter eine funktionierende Station erreicht? Bedenke, dass Eve Alice' Programmcode mitlesen kann. Sie weiß also, in welcher Reihenfolge der Esel die Stationen ablaufen wird.
3. Nimm an, dass Alice ihren Roboter so programmiert, dass dieser selbst zufällig entscheiden kann, wo er hinwill. (Eve kann zwar sehen, was Alice programmiert hat, kann aber nicht vorhersehen, welche Entscheidungen der Esel trifft, sobald er anfängt loszulaufen.) Was für eine Strategie sollte Alice dem Esel einprogrammieren, und wie sollte Eve auswählen, welche Stationen sie deaktiviert, um dem entgegen zu wirken? Mit welcher Wahrscheinlichkeit wird der Esel eine funktionierende Ladestation erreichen, falls sowohl Alice als auch Eve bestmögliche Strategien verwenden?

## 1.2 Operationen auf probabilistischen Bits

Da wir nun wissen, wie man Bits an Information durch Vektoren darstellt, können wir Operationen auf diesen als lineare Transformationen repräsentieren und Hilfsmittel aus der linearen Algebra nutzen. Zum Beispiel können wir uns die Operation anschauen, die "Kopf" und "Zahl" einer Münze vertauscht:



Wir bezeichnen diese Operation als NOT und schreiben mathematisch:

$$\text{NOT } \begin{array}{c} \text{☺} \\ \text{☹} \end{array} = \begin{array}{c} \text{☹} \\ \text{☺} \end{array}, \quad \text{NOT } \begin{array}{c} \text{☹} \\ \text{☺} \end{array} = \begin{array}{c} \text{☺} \\ \text{☹} \end{array}. \quad (1.8)$$

Mit den Konventionen aus Gl. (1.2) können wir also schreiben

$$\text{NOT } [0] = [1], \quad \text{NOT } [1] = [0]. \quad (1.9)$$

Beachte, wir schreiben NOT  $p$  als Abkürzung für NOT( $p$ ) – wobei beide Notationen einfach bedeuten, dass NOT auf einen Vektor wirkt.<sup>1</sup> Im Allgemeinen werden wir Operationen mit Großbuchstaben benennen um sie von Vektoren und Zahlen unterscheiden zu können.

Erinnere dich noch einmal an Gl. (1.2), die beschreibt, dass [0] und [1] jeweils die deterministischen Zustände 0 und 1 eines probabilistischen Bits beschreiben. Da die NOT Operation diese zwei Vektoren vertauscht, invertiert oder “negiert” sie den Wert des Bits. Genau aus diesem Grund haben wir sie “NOT” (engl. für “nicht”) benannt – sie entspricht der logischen Negation! Eine einfache Anwendung von NOT ist es, Daten in deinen Computer einzugeben. Wenn alle deine Bits zu Beginn auf 0 gesetzt sind, kannst du so einzelne auf 1 setzen, um Daten einzugeben – das ist häufig der erste Schritt einer Berechnung.

Wie sollten wir aber die NOT Operation auf einem probabilistischen Bit  $p = \begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$  definieren? Mit Wahrscheinlichkeit  $p_0$  ist das Bit 0 und wird zu einer 1 invertiert. Mit Wahrscheinlichkeit  $p_1$  ist das Bit 1 und wird zu einer 0 invertiert. Daher kann der Effekt der NOT Operation einfach so beschrieben werden:

$$\text{NOT } \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = \begin{pmatrix} p_1 \\ p_0 \end{pmatrix}. \quad (1.10)$$

Wenn man in diese Gleichung nun wieder  $p_0 = 1$  (und  $p_1 = 0$ ) oder  $p_0 = 0$  (und  $p_1 = 1$ ) einsetzt, erhalten wir die uns schon bekannte Gl. (1.9). Du kannst dir also die NOT Operation, und Gl. (1.10) wie folgt vorstellen: Wenn du dir ein probabilistisches Bit als einen ‘blinden’ Münzwurf vorstellst (du hast dir die Münze also noch nicht angeschaut), dann führst du die NOT Operation aus, indem du die Münze einmal umdrehst, immer noch ohne sie anzuschauen.

#### Übungsaufgabe 1.4: Die NOT Operation visualisieren

Wie wir in Abb. 1.2 gesehen haben, liegen alle möglichen Zustände eines probabilistischen Bits auf einer Strecke. Lass uns also versuchen zu verstehen, wie die NOT Operation diese Strecke transformiert oder verändert.

1. Such dir einen beliebigen<sup>a</sup> (engl. *arbitrary*) Punkt auf der Strecke mit Koordinaten  $(p_0, p_1)$ . Wo landet dieser Punkt nachdem du die NOT Operation auf ihn angewandt hast?
2. Was passiert mit den zwei Endpunkten der Strecke?
3. Gibt es einen Punkt auf der Strecke, der auf sich selbst abgebildet wird?

<sup>a</sup>Hier bedeutet ‘beliebig’, dass deine Berechnungen für *jeden möglichen* Punkt gelten muss. Am einfachsten ist es meist, die Werte  $p_0$  und  $p_1$  bis zum Ende als unbekannte Zahlen zu behandeln.



### 1.2.1 Erweitern durch Linearität

Wie sollten wir  $M \begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$  definieren, wenn  $M$  eine beliebige Operation auf einem Bit ist? Wie zuvor nehmen wir an, dass wir wissen, wie  $M$  sich auf die zwei möglichen Werte des Bits auswirkt.

<sup>1</sup>Man könnte auch NOT  $\cdot p$  schreiben, da diese Operation einer Matrix-Vektor-Multiplikation entspricht.

Dann schreiben wir  $M[0]$  als das Resultat der Operation, wenn das Bit 0 ist und  $M[1]$  wenn das Bit 1 ist. (Im Falle der NOT Operation ist das genau das, was wir in Gl. (1.9) gemacht haben.) Nun versuchen wir die gleichen Argumente wie oben zu nutzen: Mit Wahrscheinlichkeit  $p_0$  ist das Bit 0 und wir erhalten  $M[0]$  durch Anwenden der Operation  $M$ . Mit Wahrscheinlichkeit  $p_1$  ist das Bit 1 und wir erhalten stattdessen  $M[1]$ . Insgesamt sollten wir also definieren

$$M \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = p_0 M[0] + p_1 M[1], \quad (1.11)$$

wobei  $p_0 M[0]$  bedeutet, dass wir den Vektor  $M[0]$  mit der Wahrscheinlichkeit  $p_0$  multiplizieren.

### Übungsaufgabe 1.5: NOT auf probabilistischen Bits

Zeige, dass Gl. (1.11) zu Gl. (1.10) führt, wenn  $M$  die NOT Operation ist.

Durch Gl. (1.3) können wir Gl. (1.11) auch wie folgt umschreiben:

$$M(p_0 [0] + p_1 [1]) = p_0 M[0] + p_1 M[1]. \quad (1.12)$$

Beachte, dass der Unterschied der beiden Seiten der Gleichung in der Reihenfolge der Operationen liegt: auf der linken Seite wird erst die Linearkombination gebildet und dann  $M$  auf diese angewendet, während auf der rechten Seite zuerst  $M$  auf die Zustände wirkt und anschließend die Linearkombination gebildet wird. Diese Gleichung sieht der bekannten Regel  $a(b+c) = ab+ac$  sehr ähnlich (das "Distributivgesetz").

Wenn  $M$  eine Operation auf probabilistischen Bits ist, die Gl. (1.12) erfüllt, nennen wir  $M$  **linear**. Unsere Regeln aus Gl. (1.11) und (1.12) um  $M$  von Bits auf probabilistische Bits zu erweitern, nennt man **erweitern durch Linearität**. Häufig werden wir das gleiche Konzept für Quantenbits nutzen.

### 1.2.2 Zufällige Operationen

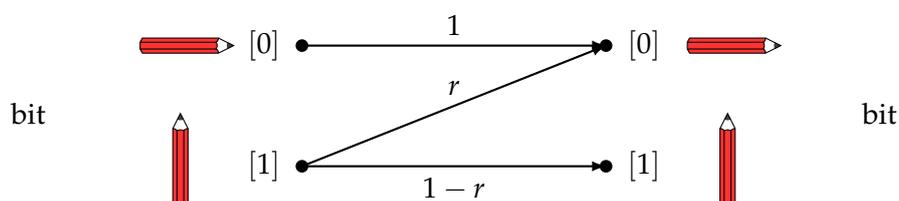
Wie du vielleicht bemerkt hast, haben wir in der Herleitung von Gl. (1.11) nicht angenommen, dass  $M[0]$  und  $M[1]$  wieder einer der deterministischen Zustände  $[0]$  oder  $[1]$  eines Bits waren (obwohl das für die NOT Operation der Fall war). Das bedeutet, dass Gl. (1.11) genauso gut funktioniert, wenn  $M[0]$  und  $M[1]$  probabilistische Bits sind! In diesem Fall nennen wir  $M$  eine **zufällige Operation**.

Eine der einfachsten zufälligen Operationen ist die folgende: Stell dir vor, du stellst  $[0]$  als einen horizontal liegenden Bleistift dar und  $[1]$  indem du den Stift vertikal auf dem Ende balancierst. Wenn du nun mit der Hand sanft auf den Tisch schlägst, fällt der Stift vielleicht um und ändert dabei den Zustand von  $[1]$  zu  $[0]$ . Lag der Stift jedoch bereits vorher, ändert sich der Zustand nicht. Also ist das schlagen des Tisches ein *probabilistisches Zurücksetzen* des Zustands des Stifts zu  $[0]$ ; je stärker du zuschlägst, desto höher ist die Wahrscheinlichkeit des Zurücksetzens.

Mathematisch beschreiben wir einen **probabilistischen Reset** (*reset* engl. für "zurücksetzen") als

$$R(r)[0] = [0] = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad R(r)[1] = r[0] + (1-r)[1] = \begin{pmatrix} r \\ 1-r \end{pmatrix}, \quad (1.13)$$

wobei  $r \in [0, 1]$  als die *Reset*-Wahrscheinlichkeit bezeichnet wird.



Durch Linearität können wir diese Operation auf alle Zustände erweitern:

$$\begin{aligned} R(r) \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} &= p_0 R(r) [0] + p_1 R(r) [1] \\ &= p_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + p_1 \begin{pmatrix} r \\ 1-r \end{pmatrix} = \begin{pmatrix} p_0 + p_1 r \\ p_1(1-r) \end{pmatrix}. \end{aligned}$$

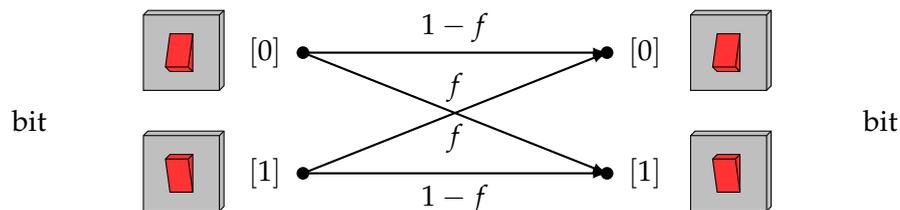
Beachte die beiden Spezialfälle dieser Gleichung  $R(0)$ , welcher den Zustand nicht verändert und  $R(1)$ , der jeden Zustand zu  $[0]$  zurücksetzt.



Ein anderes interessantes Beispiel einer zufälligen Operation ist der **probabilistische Flip**  $F(f)$ , der das Inputbit mit Wahrscheinlichkeit  $f$  invertiert oder flippt (engl. für "umdrehen") und mit Wahrscheinlichkeit  $1 - f$  nichts ändert:

$$F(f) [0] = (1 - f) [0] + f [1], \quad F(f) [1] = f [0] + (1 - f) [1], \quad (1.14)$$

wobei  $f \in [0, 1]$  die *Flip*-Wahrscheinlichkeit genannt wird. Eine anschauliche Vorstellung eines probabilistischen Flips bietet dir ein Lichtschalter, dessen Zustände  $[0]$  und  $[1]$  repräsentieren. Wirfst du nun ein Kissen gegen den Lichtschalter, so ändert sich der Zustand – also das Licht geht an oder aus – mit Wahrscheinlichkeit  $f$  und mit Wahrscheinlichkeit  $1 - f$  ändert sich nichts. Du kannst also die Funktion von  $F(f)$  wie folgt visualisieren:



Die folgende Übung wird dir helfen, dich mit dem probabilistischen Flip vertraut zu machen.

### Übungsaufgabe 1.6: Probabilistischer Flip

Sei  $F(f)$  die probabilistische Flip Operation aus Gl. (1.14).

1. Schreibe  $F(f) [0]$  und  $F(f) [1]$  als Vektoren.
2. Für welchen Wert von  $f$  entspricht  $F(f)$  der NOT Operation? Wie können wir mit  $F$  ein probabilistisches Bit aus dem  $[0]$  Zustand in einen beliebigen Zustand  $\begin{pmatrix} p \\ 1-p \end{pmatrix}$  versetzen?
3. Erweitere  $F(f)$  durch Linearität auf probabilistische Bits indem du  $F(f) \begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$  berechnest.
4. Sei  $\begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$  eine beliebige Wahrscheinlichkeitsverteilung. Zeige, dass  $F(1/2) \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$  gilt.

Die letzte Aufgabe der Übungsaufgabe zeigt, dass  $F(1/2)$  immer zur Gleichverteilung (engl. *uniform distribution*) führt, egal wie die eingehende Verteilung aussieht. Das ist nützlich, wenn man einen fairen Münzwurf simulieren will. Tatsächlich wirst du in der nächsten Hausaufgabe zeigen, dass man durch verändern der Flip-Wahrscheinlichkeit die Fairness der Münze *ändern* kann.

### Hausaufgabe 1.3: Schokoladenmünzen

Es ist der 15. Mai und Bob feiert Geburtstag! Da er Schokolade liebt, hat Alice sich entschlossen, ihm eine Schokoladenmünze zu machen. Damit die Münze ganz besonders ist, hat sie die Münze so geformt, dass sie genau mit der Wahrscheinlichkeit  $q = 5/15$ , die seinem Geburtstag entspricht, auf  landet, wenn man sie zuvor kreiseln lässt. Nach langem ausprobieren hat sie endlich genau die richtige Form gefunden. Aufgeregt lässt sie die Münze auf dem Tisch liegen, um eine Geburtstagskarte zu kaufen.

Als sie zurückkommt, muss sie leider feststellen, dass die Münze in der Sonne lag und eine Kante ein wenig angeschmolzen ist. Durch ausprobieren stellt sie fest, dass die neue Wahrscheinlichkeit von  jetzt  $p = 4/15$  ist. Leider hat sie keine Zeit mehr um eine neue Münze zu machen, also schreibt sie in die Geburtstagskarte, dass Bob die Münze nach dem Kreiseln mit Wahrscheinlichkeit  $f$  einfach umdrehen soll, damit  genau mit Wahrscheinlichkeit  $q$  eintritt. Hilf Alice herauszufinden, welchen Wert  $f$  haben sollte.

**Hinweis:** Die Werte  $p, q$  und  $f$  sollten folgende Gleichung erfüllen:  $F(f) \binom{p}{1-p} = \binom{q}{1-q}$ .

### Übungsaufgabe 1.7: Flip durch Reset und NOT

Wie kannst du  $F(f)$  aus den  $R(r)$  und NOT Operationen bauen?

## 1.3 Ein probabilistisches Bit messen

Wenn du eine Münze wirfst, sie aber sofort mit deiner Hand abdeckst, kannst du nicht wissen, auf welcher Seite sie gelandet ist. In dieser Situation lässt sich der Zustand der Münze durch die **Gleichverteilung** (engl. *uniform distribution*) beschreiben:

$$\text{chocolate coin} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} = \frac{1}{2} [0] + \frac{1}{2} [1]. \quad (1.15)$$

Wenn du dann aber deine Hand hebst und die Münze "Kopf" zeigt, wird dein Wissen über den Zustand aktualisiert zu

$$\text{chocolate coin} = [0] \quad (1.16)$$

da du ja jetzt sicher weißt, in welchem Zustand sich die Münze befindet. Wir bezeichnen den Prozess des Aufdeckens, bzw. die Hand in so einem Münzwurf zu heben, als **messen** (engl. *measurement*)<sup>2</sup>.

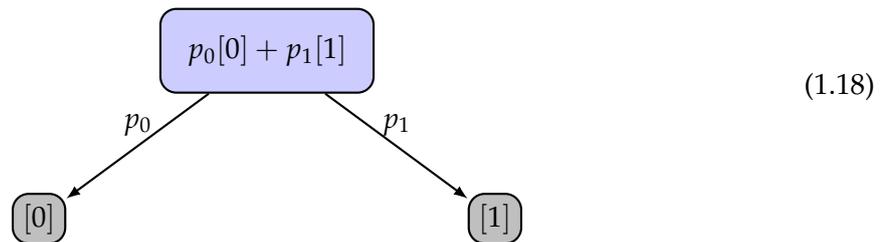
Bemerke, dass der Zustand nach dem Messen ein anderer ist als davor, siehe Gl. (1.15) und (1.16). Tatsächlich ist nach dem Messen der Zustand nicht mehr unklar. Stell dir jetzt vor, du bedeckst die Münze wieder mit deiner Hand, nachdem du ihren Zustand gerade gemessen hast. In welchem Zustand ist sie jetzt? Offensichtlich noch immer

$$\text{chocolate coin} = [0] \quad (1.17)$$

da wir ja bereits wissen, dass sie "Kopf" zeigt. Auch wenn du sie jetzt noch einmal misst (anschaut), wird sie wieder "Kopf" zeigen. Genauso verläuft das ganze, wenn du bei der ersten Messung einer zufälligen Münze "Zahl" als Ergebnis hattest: egal wie oft du die Messung wiederholst, das Ergebnis wird "Zahl" bleiben.

<sup>2</sup>Diesen Begriff haben wir aus der Welt der Quanteninformatik übernommen, wo ein ähnlicher Prozess existiert.

Im Allgemeinen resultiert das Messen eines zufälligen Bits mit der Verteilung  $\begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$  mit Wahrscheinlichkeit  $p_0$  im **Ergebnis** 0 ("Kopf") und mit Wahrscheinlichkeit  $p_1$  das Ergebnis 1 ("Zahl"):



Der Zustand des probabilistischen Bits ist nach dem Messen dann nicht mehr  $\begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$  sondern einer der beiden Basiszustände  $[0] = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  oder  $[1] = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ , in Abhängigkeit des Messergebnisses. Wenn du beispielsweise das Ergebnis 1 ("Zahl") erhältst, ist der Zustand danach  $[1]$ . Im Allgemeinen *verändert* messen den Zustand!

Es ist wichtig zu bemerken, dass sich durch das Messen eines probabilistischen Bits nicht die tatsächlichen Wahrscheinlichkeiten  $p_0$  oder  $p_1$  extrahieren lassen – du bekommst nur einen einzelnen Bit 0 oder 1 als Ergebnis. Außerdem ist die ursprüngliche Verteilung  $\begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$  nach dem Messen zerstört, man kann die Messung also nicht auf der ursprünglichen Verteilung wiederholen. Das ist tatsächlich sehr intuitiv: Wenn man eine Münze genau einmal wirft erhält man nur ein einzelnes zufälliges Ergebnis – man weiß aber nicht ob die Münze fair oder unfair ist.

Wenn wir aber die gleiche Münze sehr häufig werfen, erwarten wir, dass der Anteil der Würfe mit Ergebnis 1 in etwa  $p_1$  ist. Mit anderen Worten,

$$\frac{N_1}{N} \approx p_1, \quad (1.19)$$

wobei  $N$  die Anzahl aller Messungen und  $N_1$  die Anzahl der Messungen mit Ergebnis 1 ist. Je mehr Messungen wir durchführen, desto genauer sollte  $p_1$  angenähert werden<sup>3</sup>.

#### Hausaufgabe 1.4: Münzen werfen

1. Nimm dir eine Münze und markiere die Seiten mit 0 und 1. Wirf die Münze 30 mal und dokumentiere die Ergebnisse in einer Tabelle der folgenden Form:

Anzahl der Würfe $N$	1	2	3	4	5	6	7	8	9	...	30
$N$ -tes Ergebnis	1	0	1	0	0	0	1	1	1	...	1

( Die grauen Ergebnisse sind nur Beispiele, ersetze sie mit deinen Ergebnissen.)

2. Schätze mit Gl. (1.19) die Wahrscheinlichkeit, mit der deine Münze auf der mit 1 markierten Seite landet.
3. Es ist interessant zu sehen, wie sich diese Schätzung verändert, je höher die Zahl der Würfe  $N$  steigt. Um das anschaulich zu machen, erweitere deine Tabelle aus Aufgabe 1 um drei weitere Zeilen, sodass sie wie folgt aussieht:

Anzahl der Würfe $N$	1	2	3	4	5	6	7	8	...	30
$N$ -tes Ergebnis	1	0	1	0	0	0	1	1	...	1
Summe $N_1$	1	1	2	2	2	2	3	4	...	16
Anteil $N_1/N$	1	1/2	2/3	2/4	2/5	2/6	3/7	4/8	...	16/30
Wert des Bruchs	1.00	0.50	0.67	0.50	0.40	0.33	0.43	0.50	...	0.53

<sup>3</sup>Wie gut ist diese Annäherung? Man kann zeigen, dass der Fehler im Durchschnitt etwa Größenordnung  $1/\sqrt{N}$  hat, also schnell auf 0 zugeht, wenn wir häufig messen.

Die Zeilen haben dann folgende Bedeutung: (1) Anzahl  $N$  der Würfe bisher, (2) Ergebnis des  $N$ -ten Wurfs, (3) Summe der ersten  $N$  Würfe, (4) Schätzung der Wahrscheinlichkeit für Ergebnis 1 basiert auf den ersten  $N$  Messungen, (5) Dezimaldarstellung der Schätzung. Wenn du möchtest, kannst du *Excel* oder ein ähnliches Programm nutzen

4. Zeichne einen Graphen der letzte Zeile der Tabelle als eine Funktion der Anzahl der Würfe  $N$ .

## 1.4 Der QUIRKY Simulator

Die Gesetze der Wahrscheinlichkeit können ein wenig schwer zu verstehen sein. Zum Glück können wir versuchen, deren Verhalten mithilfe von Computern zuhause (oder dem in deiner Tasche)<sup>4</sup> zu **simulieren**.

In diesem Kurs verwenden wir einen Simulator, der QUIRKY heißt. QUIRKY ist das kleine Geschwisterchen von Quirk, einem Simulator mit mehr Funktionen, der von Craig Gidney bei Google entwickelt wurde. Da Craig seinen Quellcode mit einer *open-source*-Lizenz veröffentlicht hat, konnten wir den Simulator auf die Bedürfnisse dieses Kurses anpassen. Das Beste an QUIRKY ist, dass er direkt in deinem Browser funktioniert – du musst ihn nicht installieren! Besuche einfach:

<https://www.quantum-quest.org/quirky>

und klicke auf “Quest 1”. Probier es ruhig kurz aus – du kannst QUIRKY auch auf deinem Handy benutzen! Wenn du QUIRKY das erste Mal öffnest, sollte es in etwa wie in Abb. 1.3 aussehen.

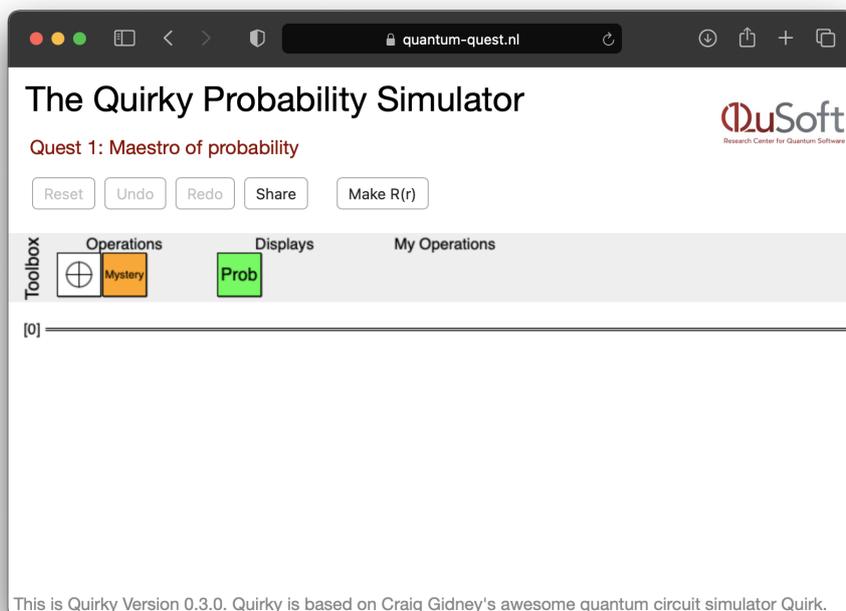


Abbildung 1.3:  
Erster Blick auf QUIRKY.

<sup>4</sup>Bis zu einem gewissen Maß gilt das auch für Quantencomputer – aber da greifen wir voraus...

### 1.4.1 Erste Schritte

Lass uns die Benutzeroberfläche von QUIRKY Schritt für Schritt anschauen. Oben findest du eine *Menüzeile* mit nützlichen Befehlen:



Die 'Reset'-Taste lässt dich QUIRKY zurücksetzen und von vorne beginnen. Die 'Undo'- und 'Redo'-Tasten machen Änderungen rückgängig oder stellen sie wieder her (*undo* engl. für "rückgängig machen", *redo* engl. für "wiederherstellen"). Beachte, dass du auch einen 'Reset' rückgängig machen kannst. Durch einen Klick auf 'Share' kannst du dein Programm mit anderen teilen. Über die 'Make R(r)'-Taste sprechen wir später.

Unter dem Menü gibt es die *Toolbox* mit den grundlegenden Operationen die wir schon kennen:



Die erste Box,  $\oplus$ , entspricht der NOT Operation aus §1.2, die den Zustand invertiert. Die anderen zwei Operationen betrachten wir gleich. Zum Glück musst du dir das nicht alles merken – wenn du mit der Maus über die Boxen fährst (oder sie auf deinem Handy antippst) siehst du ihre Beschreibung.

Unter der Toolbox siehst du das Herz von QUIRKY, das *probabilistische Bit*:



Die Doppellinie (oder auch 'Draht') entspricht einem Bit, das mit dem Zustand [0] initialisiert ist. Du kannst Operationen auf das Bit anwenden, indem du sie aus der Toolbox auf den Draht ziehst. Versuche mal, die folgende simple Berechnung in QUIRKY umzusetzen:<sup>5</sup>



Wie können wir nun das Ergebnis dieser Berechnung anzeigen? Da wir im Allgemeinen mit Wahrscheinlichkeiten arbeiten, suchen wir einen Weg, *Wahrscheinlichkeiten anzuzeigen*. Mit der grünen Box mit dem Namen **Prob** aus dem 'Display'-Bereich der Toolbox können wir genau das tun. Füg sie doch einfach mal zu der Berechnung hinzu und schau was passiert:



Es sieht so aus, als würde man bei einer Messung mit Wahrscheinlichkeit 100% das Ergebnis 1 erhalten (wenn du mit der Maus über die Box fährst, siehst du welcher Zustand oben und unten ist). Das entspricht natürlich unserer Erwartung. Der ursprüngliche [0] Zustand wird durch die NOT Operation zu einem [1] Zustand geflippt. Nach den Mess-Regeln aus Gl. (1.18) wird das Ergebnis also immer 1 sein.

<sup>5</sup>Falls du die digitale Version dieser Vorlesungsnotizen liest, kannst du einfach auf die Bilder klicken, um QUIRKY im Browser zu öffnen. Dann wird der ganze Schaltkreis automatisch kopiert! Falls das nicht funktioniert, öffne <https://www.quantum-quest.org/quirky> einfach selbst.

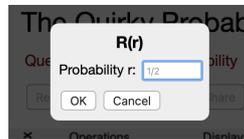
## Übungsaufgabe 1.8: Eine Operation entfernen

In QUIRKY kannst du Operationen auch wieder entfernen, indem du sie einfach vom Draht herunter zurück in die Toolbox ziehst. Entferne doch einfach mal die NOT Operation und überprüfe ob das Messergebnis jetzt immer 0 ist.

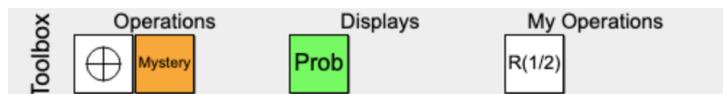
### 1.4.2 Eigene Operationen erstellen

Bisher hast du nur gelernt, die  $[0]$  und  $[1]$  Zustände mit QUIRKY zu erstellen. Um interessante Verteilungen zu erstellen, kannst du die Reset-Operation  $R(r)$  aus §1.2.2 benutzen. Da es unendlich viele dieser Operationen gibt (eine für jedes mögliche  $r$ ), können nicht alle in der Toolbox angezeigt werden. Stattdessen kannst du sie selbst in die Toolbox hinzufügen!

Lass uns das mal üben. Füge eine Operation hinzu, die mit Wahrscheinlichkeit  $r = \frac{1}{2} = 50\%$  einen Reset durchführt. Zuerst klickst du auf die 'Make  $R(r)$ '-Taste im Menü. Ein Popup-Fenster öffnet sich, in dem du  $r$  eingeben kannst:



Gib  $1/2$  ein und bestätige mit der 'Ok'-Taste. Glückwunsch! Du hast erfolgreich die  $R(1/2)$ -Operation zur Toolbox hinzugefügt, die jetzt so aussieht:



Um deine neue Operation zu testen, kannst du folgende Veränderung in QUIRKY machen:



Lass uns kurz überprüfen, dass das neue Ergebnis sinnvoll ist: Das Bit startet im  $[0] = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ -Zustand. Durch die NOT-Operation wird das Bit in den  $[1] = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ -Zustand geflippt. Nach Gl. (1.13) setzt die Operation  $R(1/2)$  Bits im Zustand  $[1]$  mit Wahrscheinlichkeit  $1/2$  zurück, ändert den Zustand also zu:

$$R(1/2)[1] = \frac{1}{2}[0] + \frac{1}{2}[1] = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} = \begin{pmatrix} 50\% \\ 50\% \end{pmatrix}.$$

Das ist genau das gleiche Ergebnis, das wir auch in QUIRKY gesehen haben.

In der folgenden Hausaufgabe wirst du QUIRKY nutzen um ein komplizierteres Experiment durchzuführen.

### Hausaufgabe 1.5: Zweimal Resetten

1. Baue die folgende Sequenz von Operationen in QUIRKY: Generiere zuerst den Zustand  $[1]$ , resette dann mit Wahrscheinlichkeit  $r = \frac{1}{4}$  und resette anschließend mit Wahrscheinlichkeit  $r = \frac{2}{3}$ . Nutze das Tool zum Anzeigen von Wahrscheinlichkeiten um die Wahrscheinlichkeiten der Ergebnisse herauszufinden.
2. Zeige, dass die Antwort aus QUIRKY korrekt ist.

### 1.4.3 Eine mysteriöse Operation



Bisher haben wir immer noch nicht über die mysteriöse orangene Box gesprochen. Lass uns diese Operation doch  $M$  nennen. Wie können wir herausfinden, was die Operation  $M$  macht? Als erstes können wir versuchen herauszufinden, was das Ergebnis von  $M[0]$  ist, also was passiert, wenn man  $M$  auf ein Bit im Zustand  $[0]$  anwendet. In QUIRKY entspricht das folgender Berechnung:



Wie können wir jetzt  $M[0]$  ablesen? Wir müssen uns an dieser Stelle kurz daran erinnern, dass man bei zufälligen Bits in der Natur *nicht* einfach die Wahrscheinlichkeiten ablesen kann. Stattdessen muss man, wie in §1.3 beschrieben, viele Messungen durchführen (z.B. viele Münzwürfe durchführen) und die Wahrscheinlichkeiten vom Ergebnis abschätzen. Der Vorteil von einem Simulator wie QUIRKY ist, dass diese Regeln für uns nicht gelten – wir können mit der Wahrscheinlichkeitsanzeige den Zustand einfach anzeigen lassen:



Also wissen wir jetzt

$$M[0] = 0.2[0] + 0.8[1].$$

Jetzt bist du an der Reihe!

#### Hausaufgabe 1.6: Zeit für ein Mysterium

1. Bestimme den Zustand  $M[1]$
2. Bestimmen  $M[0]$  und  $M[1]$  die zufällige Operation  $M$  vollständig?  
Falls ja, schreibe eine Formel für  $M\left(\frac{1}{2}\right)$  auf und überprüfe sie mit QUIRKY. Falls nicht, erkläre wieso.

In den kommenden Wochen werden wir den Sprung von gewöhnlichen Bits zu Quantenbits wagen und lernen, wie man mit diesen auf immer raffinierte Weise rechnet. QUIRKY wird uns dabei als zuverlässiges Werkzeug dienen und im Laufe der Zeit mit immer mehr neuen Funktionen ausgestattet werden. Benutze es gerne, um die Theorie, die du lernen wirst zu verstehen und um dir bei den Hausaufgaben zu helfen.

## 1.5 Lösungen der Übungsaufgaben

### Lösung Übungsaufgabe 1.1

1. Da immer eines der Ereignisse eintritt, muss die Summe der Wahrscheinlichkeiten notwendigerweise sich auf 1 addieren. Das bedeutet, dass  $p_0 + p_1 = 1$  gilt. Durch umformen erhält man  $p_1 = 1 - p_0$ , was einer Geradengleichung mit Steigung  $-1$  entspricht.
2. Wenn die Strecke über die Koordinatenachsen hinaus gehen würde, müsste eine der Wahrscheinlichkeiten negativ werden. Da Wahrscheinlichkeiten aber nicht negativ sein können, müssen die Bedingungen  $p_0 \geq 0$  und  $p_1 \geq 0$  gelten, also die Strecke an den Koordinatenachsen enden.
3. Das ist der Mittelpunkt, an dem  $p_0 = p_1 = \frac{1}{2}$  gilt.

### Lösung Übungsaufgabe 1.2

1. Der Sekundenzeiger kann 60 verschiedene Werte anzeigen. Die Wahrscheinlichkeit einen bestimmten zu sehen ist  $\frac{1}{60}$ .
2. Die letzte Ziffer kann 10 verschiedene Werte haben. Die Wahrscheinlichkeit einen bestimmten zu sehen ist  $\frac{1}{10}$ .
3. Die erste Ziffer kann 6 verschiedene Werte haben. Die Wahrscheinlichkeit einen bestimmten zu sehen ist  $\frac{1}{6}$ .
4. Wenn du nur die erste Ziffer siehst, kann die letzte Ziffer immer noch alle 10 möglichen Werte mit gleicher Wahrscheinlichkeit einnehmen. Das gleiche gilt, wenn du nur die letzte Ziffer siehst, dann kann die erste immer noch alle 6 möglichen Werte mit gleicher Wahrscheinlichkeit einnehmen. Also sind die Werte der zwei Ziffern unabhängig. Du kannst die Wahrscheinlichkeit mit folgender Gleichung verifizieren:

$$\frac{1}{6} \cdot \frac{1}{10} = \frac{1}{60}.$$

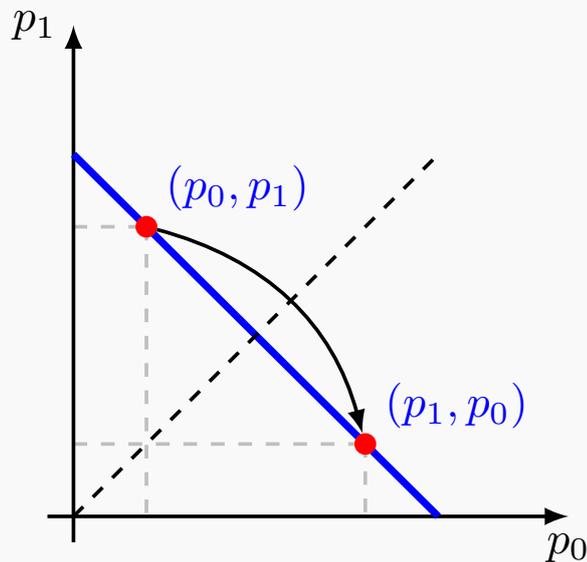
### Lösung Übungsaufgabe 1.3

Es gibt sechs verschiedene Fälle, in denen beide Ziffern gleich sind (von **00** bis **55**). Jeder einzelne Fall tritt mit Wahrscheinlichkeit  $\frac{1}{60}$  ein. Wir können diese sechs Ereignisse in eines zusammenfassen, sodass dessen Wahrscheinlichkeit die Summe der Einzelwahrscheinlichkeiten ist:

$$\underbrace{\frac{1}{60} + \frac{1}{60} + \frac{1}{60} + \frac{1}{60} + \frac{1}{60} + \frac{1}{60}}_{6 \text{ Terme}} = \frac{6}{60} = \frac{1}{10}.$$

### Lösung Übungsaufgabe 1.4

1. In Gl. (1.10) haben wir gesehen, dass der Punkt  $(p_0, p_1)$  durch die NOT-Operation auf den Punkt  $(p_1, p_0)$  abgebildet wird. Mit anderen Worten, die Koordinaten des Punktes werden getauscht. Hier ist ein Beispiel, wie das grafisch aussieht:



Du kannst dir die NOT-Operation also grafisch als "Spiegeln an der gestrichelten Diagonale" vorstellen.

2. Nach Gl. (1.2) korrespondieren die zwei Endpunkte der Achse mit den zwei deterministischen Zuständen  $[0]$  und  $[1]$ . In Gl. (1.9) haben wir gesehen, dass die NOT-Operation die beiden miteinander vertauscht.
3. Ein Punkt mit den Koordinaten  $(p_0, p_1)$  bleibt unverändert von der NOT-Operation, wenn  $(p_0, p_1) = (p_1, p_0)$  also  $p_0 = p_1$ . Da  $p_0 + p_1 = 1$  gelten muss, gilt das für  $p_0 = p_1 = 1/2$ , was dem Punkt  $(1/2, 1/2)$  entspricht. Dies ist der einzige Punkt der unverändert bleibt.

### Lösung Übungsaufgabe 1.5

Wir nutzen Gl. (1.9) und (1.11),

$$\text{NOT} \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = p_0 \text{NOT} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + p_1 \text{NOT} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = p_0 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + p_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} p_1 \\ p_0 \end{pmatrix},$$

was dasselbe ist wie Gl. (1.10).

## Lösung Übungsaufgabe 1.6

1. Aus der Definition von  $F(f)$  in Gl. (1.14),

$$F(f) [0] = (1-f) \binom{1}{0} + f \binom{0}{1} = \binom{1-f}{f},$$

$$F(f) [1] = f \binom{1}{0} + (1-f) \binom{0}{1} = \binom{f}{1-f}.$$

2.  $F(f)$  flippt das Bit mit Sicherheit wenn  $f = 1$ , also ist NOT =  $F(1)$ . Um  $[0]$  zu einem beliebigem Zustand  $\binom{p}{1-p}$  zu verändern, müssen wir  $f = 1 - p$  wählen. Tatsächlich sieht man aus der ersten Gleichung oben, dass

$$F(1-p) [0] = \binom{1-(1-p)}{1-p} = \binom{p}{1-p}.$$

3. Nach Gl. (1.11),

$$F(f) \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = p_0 \binom{1-f}{f} + p_1 \binom{f}{1-f} = \begin{pmatrix} p_0(1-f) + p_1 f \\ p_0 f + p_1(1-f) \end{pmatrix}.$$

4. Wenn man  $f = 1/2$  in die vorige Gleichung einsetzt, erhält man

$$F(1/2) \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = \begin{pmatrix} p_0/2 + p_1/2 \\ p_0/2 + p_1/2 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} p_0 + p_1 \\ p_0 + p_1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

## Lösung Übungsaufgabe 1.7

Wir unterscheiden zwei Fälle:

- $\frac{1}{2} \leq f \leq 1$ : In diesem Fall gilt  $0 \leq \frac{1-f}{f} \leq 1$ . Wir behaupten, dass die Flip-Operation  $F(f)$  gebaut werden kann, indem zuerst  $R(\frac{1-f}{f})$ , dann NOT und dann  $R(1-f)$  angewendet wird. Tatsächlich gilt:

$$R(1-f) \text{ NOT } R(\frac{1-f}{f}) [0] = R(1-f) \text{ NOT } [0] = R(1-f) [1] = (1-f) [0] + f [1]$$

und

$$\begin{aligned} R(1-f) \text{ NOT } R(\frac{1-f}{f}) [1] &= R(1-f) \text{ NOT } \left( \frac{1-f}{f} [0] + (1 - \frac{1-f}{f}) [1] \right) \\ &= R(1-f) \left( (1 - \frac{1-f}{f}) [0] + \frac{1-f}{f} [1] \right) \\ &= (1 - \frac{1-f}{f}) [0] + \frac{1-f}{f} ((1-f) [0] + f [1]) \\ &= f [0] + (1-f) [1]. \end{aligned}$$

- $0 \leq f \leq \frac{1}{2}$ : Dieser Fall kann auf den ersten reduziert werden, da  $F(f)$  anzuwenden das gleiche ist wie zunächst  $F(1-f)$  und anschließend NOT.



## Quest 2: Das Qubit bezwingen

Da du nun Wahrscheinlichkeiten und probabilistische Bits beherrschst, bist du bereit, etwas über Quantenbits zu lernen. Quantenbits sind den probabilistischen Bits sehr ähnlich, tatsächlich musst du nur Wahrscheinlichkeiten durch Quanten-Amplituden ersetzen. Deine dieswöchige Quest wird dir alles wichtige über die Zustände eines Qubit, erlaubte Operationen auf einem Qubit und das Messen eines Qubits beibringen. Außerdem wirst du die neue *Quanten*-variante von QUIRKY ausprobieren können.

### 2.1 Quantenbits

Bits sind die Einheit für Information in modernen Computern. Um ein Bit zu bauen, braucht man ein physisches Objekt, das sich in einem von zwei zuverlässig unterscheidbaren Zuständen befindet, wie zum Beispiel eine Münze mit zwei Seiten oder ein Kondensator, der elektrische Ladung bei zwei verschiedenen Spannungsniveaus speichert.<sup>6</sup> Das Verhalten dieser Objekte (und auch der entsprechenden Bits) kann durch physikalische Theorien wie Mechanik (für Münzen) oder Elektromagnetismus (für Kondensatoren) beschrieben werden.

Für sehr kleine<sup>7</sup> Objekte gelten diese Gesetze nicht mehr und wir müssen eine grundlegendere Theorie namens **Quantenmechanik** nutzen. Ein Elektron hat zum Beispiel eine Eigenschaft, die 'Spin' (von engl. für "Drehung" oder "Drall") genannt wird, welche (wie eine Münze) zwei Werte – hoch oder runter – annehmen, und damit zum Speichern von Bits genutzt werden kann. Aber anders als eine Münze, kann der Spin eines Elektrons auch in einer sogenannten "Superposition" beider Zustände sein! In gewisser Hinsicht ähnelt das einem probabilistischen Bit, das in einem Zwischenzustand zwischen 0 und 1 sein kann.

Es gibt aber einen feinen Unterschied zwischen Wahrscheinlichkeiten und "Superpositionen" (siehe §2.6.1 über Interferenz). Wie wir sehen werden, führen die Gesetze der Quantenmechanik zu einem viel grundlegenden Begriff von Information als einem Bit – einem **Quantenbit** oder **Qubit**.

Wir werden Qubits mit einem simplen mathematischen Modell definieren, ohne uns Gedanken über Interpretation deren seltsamen Verhaltens zu machen und stattdessen die Frage stellen: "Wofür können wir das nutzen?". Genauso wenig werden wir uns darüber Gedanken machen, wie man Qubits physisch bauen kann, bzw. aus welchen physischen Objekten sie bestehen. Falls dich das aber interessiert, in §2.6.2 reden wir kurz darüber, wieso polarisiertes Licht genutzt werden kann, um Qubits zu bauen.

#### 2.1.1 Wahrscheinlichkeiten versus Amplituden

Quantenbits sind probabilistischen Bits sehr ähnlich, es gibt nur zwei große Unterschiede:

1. Wahrscheinlichkeiten werden durch Amplituden ersetzt (welche auch negativ sein können),
2. Amplituden werden während dem Messen quadriert (Wahrscheinlichkeiten dagegen nicht).

Auf diese Unterschiede werden wir in Kürze mit mehr Detail eingehen, aber zunächst schauen wir uns einmal die möglichen Zustände eines Qubits an. Erinnerst du dich, wie wir die zwei Seiten einer Münze genutzt haben, um zwei deterministische Zustände eines

<sup>6</sup>So werden Bits auch auf deinem Computer, Handy usw. gespeichert.

<sup>7</sup>Mit "sehr klein" meinen wir hier *wirklich sehr sehr klein!* Die Anzahl der Elektronen, die man nebeneinander legen müsste um eine Linie der Länge 1 cm zu bilden ist vergleichbar mit der Anzahl an Papierblättern, die man stapeln müsste um den Mond zu erreichen!



probabilistischen Bits zu beschreiben (siehe Abb. 1.1)? In der Quanteninformatik werden diese zwei Zustände meistens  $|0\rangle$  und  $|1\rangle$  genannt, um sie von den klassischen Bits  $[0]$  und  $[1]$  zu unterscheiden. Genau wie bei probabilistischen Bits kann der Zustand eines Qubits  $|\psi\rangle$  als Linearkombination, oder auch **Superposition**, dieser zwei deterministischen Zustände geschrieben werden:

$$|\psi\rangle = \psi_0 |0\rangle + \psi_1 |1\rangle. \quad (2.1)$$

Hier ist der griechische Buchstabe  $\psi$  (ausgesprochen: "psi") der Name des Zustands (genau wie wir das probabilistische Bit  $p$  genannt haben). Die Klammern  $|\cdot\rangle$  bilden dabei eine sogenannte "ket"-form (vom engl. *bracket* für "Klammer"), die zeigt, dass wir mit einem Quantenzustand rechnen. Zum Vergleich kannst du dir in Gl. (1.3) noch einmal ansehen, dass ein beliebiges probabilistisches Bit  $p$  wie folgt geschrieben werden kann:

$$p = p_0[0] + p_1[1]. \quad (2.2)$$

Beachte, dass Gl. (2.1) identisch aussieht, außer, dass die Wahrscheinlichkeiten  $p_0$  und  $p_1$  durch die **Amplituden**  $\psi_0$  und  $\psi_1$ , sowie die klassische Notation  $[0]$  und  $[1]$  durch die Quanten Notation  $|0\rangle$  und  $|1\rangle$  ersetzt wurden! Es gibt allerdings einen wichtigen Unterschied: Während für die Wahrscheinlichkeiten aus Gl. (2.2) folgendes gilt,

$$p_0, p_1 \geq 0 \quad p_0 + p_1 = 1, \quad (2.3)$$

gilt für die Amplituden

$$\psi_0^2 + \psi_1^2 = 1. \quad (2.4)$$

Daraus folgt insbesondere  $\psi_0^2 \leq 1$  und  $\psi_1^2 \leq 1$  und damit  $\psi_0, \psi_1 \in [-1, 1]$ . Im Gegensatz dazu folgt aus den Bedingungen für Wahrscheinlichkeiten in Gl. (2.3), dass  $p_0, p_1 \in [0, 1]$ . Der entscheidende Unterschied ist dabei, dass Amplituden negativ sein dürfen, Wahrscheinlichkeiten dagegen nicht!

Genau wie bei probabilistischen Bits, ist es sinnvoll Zustände als Vektoren darzustellen. Analog zu Gl. (1.2) stellen wir die deterministischen Qubit Zustände  $|0\rangle$  und  $|1\rangle$  durch die zwei Basisvektoren dar:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Damit ist ein allgemeiner Quantenzustand  $|\psi\rangle$  aus Gl. (2.1) dargestellt als

$$|\psi\rangle = \psi_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \psi_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix}.$$



### 2.1.2 Ein Qubit als ein Kreis

Bemerke, dass Gl. (2.4) für Quantenamplituden ähnlich der Gleichung  $x^2 + y^2 = 1$  des Einheitskreises ist. Lass uns versuchen, diese Korrelation genauer zu analysieren, sie wird uns nämlich helfen, Qubits zu visualisieren und sie auf einer intuitiven Ebene zu verstehen.

Es ist sinnvoll, ein Qubit zu parametrisieren indem wir

$$\psi_0 = \cos \theta, \quad \psi_1 = \sin \theta$$

setzen, wobei  $\theta \in [0, 2\pi)$  ein Winkel ist. Tatsächlich erlauben wir jede reale Zahl für  $\theta$  (solange wir daran denken, dass zwei Winkel identisch sind, wenn sie die Distanz  $2\pi$  haben). Da  $\cos^2 \theta + \sin^2 \theta = 1$  gilt, ist Gl. (2.4) automatisch erfüllt. Bei dieser Wahl für  $\psi_0$  und  $\psi_1$  sieht ein allgemeiner Qubit Zustand so aus:

$$|\psi(\theta)\rangle = \cos \theta |0\rangle + \sin \theta |1\rangle = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}. \quad (2.5)$$

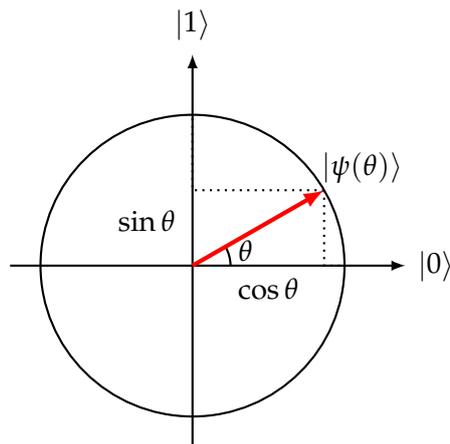


Abbildung 2.1: Der Qubit Zustand  $|\psi(\theta)\rangle$  als Punkt auf dem Einheitskreis.

Diesen Zustand können wir uns als zweidimensionalen Vektor vorstellen, der am Ursprung beginnt und Winkel  $\theta$  zur horizontalen Achse  $|0\rangle$  hat (siehe Abb. 2.1). Insbesondere gilt  $|0\rangle = |\psi(0)\rangle$  und  $|1\rangle = |\psi(\frac{\pi}{2})\rangle$ . Damit ergibt die Menge aller Qubit Zustände den Einheitskreis, mit Zentrum im Ursprung. Im Vergleich dazu hatten wir in Abb. 1.2 dargestellt, dass die Zustände eines probabilistischen Bits einer Strecke zwischen  $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$  und  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$  auf den Koordinatenachsen entsprechen. Beide Mengen sind in Abb. 2.2 abgebildet.

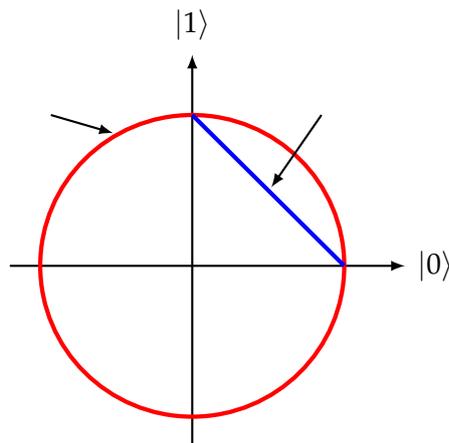


Abbildung 2.2: Der Zustandsraum eines probabilistischen Bits (blau) sowie eines Qubits (rot).

### Übungsaufgabe 2.1: Zustände auf dem Kreis

Betrachte folgende zwei Zustände eines Qubits:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Wo liegen diese Zustände auf dem Einheitskreis? Welchem Winkel  $\theta$  entsprechen sie jeweils?

## 2.2 Ein Qubit messen

Wir wissen nun, dass jeder Zustand eines Qubits die Form  $|\psi(\theta)\rangle$  hat. Sagen wir, du bekommst den Zustand  $|\psi(\theta)\rangle$  in die Hände und würdest gerne herausfinden, welchen Wert  $\theta$  hat. Leider

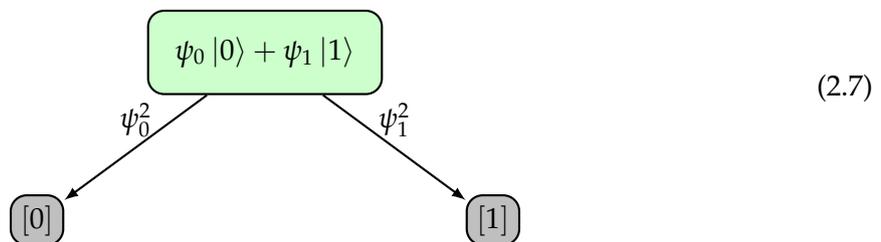


lässt dich die Quantenmechanik nicht an diesen Wert heran! Das scheint ein riesengroßes Problem zu sein – was bringt ein Quantencomputer, wenn man nicht an die Antwort kommt? Nun, nicht so schnell! Erwinnere dich an Gl. (1.18), das Gleiche gilt auch für probabilistische Bits – wenn du ein probabilistisches Bit mit Verteilung  $\begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$  misst, findest du den Wert von  $p_0$  oder  $p_1$  nicht heraus. Du bekommst nur ein einzelnes Bit an Information: 0 mit Wahrscheinlichkeit  $p_0$  und 1 mit Wahrscheinlichkeit  $p_1$ .

Die **Quantenmessung** ist da sehr ähnlich und wird durch die sogenannte *Born Regel*<sup>8</sup> beschrieben. Wenn du einen Qubit im Zustand  $\begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix} = \psi_0 |0\rangle + \psi_1 |1\rangle$  misst, erhältst du auch nur ein einzelnes Bit an Information: 0 oder 1 mit Wahrscheinlichkeiten

$$p_0 = \psi_0^2, \quad p_1 = \psi_1^2. \quad (2.6)$$

Das Quadrieren mag überraschend wirken, aber bedenke, dass  $p_0 + p_1 = \psi_0^2 + \psi_1^2 = 1$  gilt. Also ist das quadrieren notwendig, damit  $\begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$  eine gültige Wahrscheinlichkeitsverteilung ist, also ergibt die Regel Sinn! Nach dem Messen ist der Zustand verschwunden und du hast nur noch ein einzelnes Bit mit Information über das Ergebnis der Messung. Mit anderen Worten, das Messen wandelt ein Qubit in ein klassisches Bit um, dessen Wert zufällig durch Gl. (2.6) bestimmt wird:



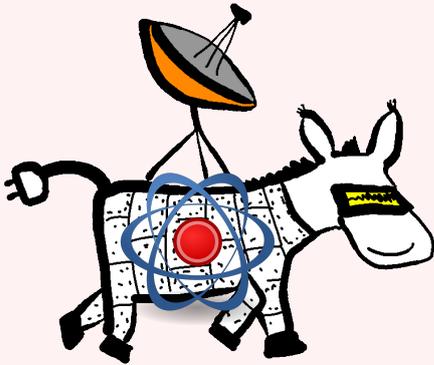
Wie du in Gl. (1.18) und (2.7) siehst, sind die Regeln für das Messen von probabilistischen Bits und Qubits sehr ähnlich. In beiden Fällen geht der ursprüngliche Zustand durch das Messen verloren und man erhält nur ein einzelnes Bit, dessen Wert zufällig vom ursprünglichen Zustand abhängt. (Insbesondere erhält man immer das gleiche Ergebnis, wenn man einen Zustand mehrmals hintereinander misst – mehrfach messen liefert also keine neuen Informationen.) Der einzige Unterschied ist, dass man für Qubits die Amplituden quadrieren muss, um die Wahrscheinlichkeitsverteilung zu erhalten, wie in Gl. (2.6), während man die Wahrscheinlichkeiten von probabilistischen Bits direkt erhält. Das scheint zwar ein kleiner Unterschied zu sein, dieser hat jedoch bedeutende Auswirkungen auf die möglichen Zustände, diese können jetzt auch negativ sein, während probabilistische Bits immer Positiv sind (siehe Abb. 2.2).

Es gibt einen noch feineren Unterschied. Es kann nämlich niemand das Ergebnis einer Quantenmessung vorhersehen. Der Unterschied ist fein, denn das gleiche scheint auch für probabilistische Bits zu gelten. Worin liegt der Unterschied? Kurz gesagt, probabilistische Bits scheinen zufällig, weil wir nicht wissen, in welchem Zustand sie sind (bzw. welche Verteilung sie haben), während Qubits selbst dann zufällig sind, wenn wir alles über ihren Zustand wissen, was es zu wissen gibt. Stell dir zum Beispiel vor, du wirfst eine faire Münze und verdeckst sie sofort danach mit deiner Hand. Den Zustand würdest du normalerweise als Gleichverteilung beschreiben (siehe Gl. (1.15)). Wenn du die Münze beim Flug aber mit einer Hochgeschwindigkeitskamera filmst, könntest du wahrscheinlich relativ gut vorhersagen, auf welcher Seite sie gelandet ist, basierend auf den Filmaufnahmen. Die Gesetze der Mechanik erlauben dir dabei, basierend auf der Rotation, Geschwindigkeit und Position der Münze vorherzusagen, wie sie landet. In diesem Sinne basiert der Zufall eines probabilistischen Bits auf unserem Unwissen. Bei Qubits dagegen ist der Zufall fundamentaler Bestandteil. Egal was wir im Vorhinein an Informationen über das Qubit gesammelt haben, es ist im Allgemeinen *unmöglich* das

<sup>8</sup>Benannt nach Max Born, einem deutschen Physiker

Ergebnis einer Quantenmessung perfekt vorherzusagen. Das bedeutet im Umkehrschluss, dass Quantenmessungen eine gute Quelle für Zufall sind!

### Hausaufgabe 2.1: Ein zufälligen Bit auf Quantenbasis erzeugen



**Das Problem:** Der Akku von Alice' Eselsroboter ist schon wieder fast leer und muss den Weg zu einer Ladestation finden. Blöderweise hat Eve es geschafft, den Zufallszahlengenerator des Esels zu hacken. Aber da Eve damit auf einem Hackerforum angegeben hat, weiß auch Alice davon. Um deren bösen Plan aufzuhalten, hat Alice einen Mini-Quantencomputer mit einem Qubit im Roboter verbaut. Sie will die fundamentale Unvorhersehbarkeit einer Quantenmessung nutzen, um Zufallsbits zu generieren, die Eve nicht vorhersehen kann.

**Fragen:** Alice kann das Qubit in jeden Zustand  $|\psi(\theta)\rangle$  versetzen und will ein zufälliges Bit generieren, indem sie das Qubit misst.

1. Mit welcher Wahrscheinlichkeit erhält sie das Ergebnis 0 beim messen? Mit welcher Wahrscheinlichkeit das Ergebnis 1?
2. Alice will den Winkel  $\theta$  so einstellen, dass beide Wahrscheinlichkeiten  $1/2$  sind. Welchen Winkel  $\theta$  sollte sie wählen? (Es könnte mehrere Lösungen geben!)

### 2.3 Qubits mit QUIRKY simulieren

Die Gesetze der Quanteninformatik sind seltsam und die meisten von uns haben keinen Quantencomputer, auf dem man herumexperimentieren könnte. Glücklicherweise hat QUIRKY seit letzter Woche neue Funktionen bekommen und kann nun ein *Quanten*-Bit simulieren!<sup>9</sup> Besuche zunächst wieder:

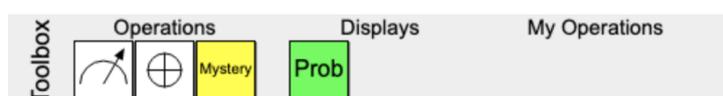
<https://www.quantum-quest.org/quirky>

und tippe auf "Quest 2". Dein Fenster wird ähnlich wie in Abb. 2.3 aussehen.

Der wichtigste Unterschied zu letzter Woche ist, dass der 'Draht' nun einem *Qubit* entspricht, welches in Zustand  $|0\rangle$  startet.

$|0\rangle$  \_\_\_\_\_

Genau wie letzte Woche enthält die *Toolbox* wieder Operationen, die du auf den Draht ziehen kannst:



Die erste Operation, , lässt uns Qubits messen. Lass uns doch direkt mal den ersten einfachen Schaltkreis in QUIRKY bauen:

<sup>9</sup>Warum brauchen wir überhaupt Quantencomputer, wenn wir sie doch mit normalen simulieren können? Nun, Simulatoren wie QUIRKY funktionieren gut, solange wir nur auf einer Handvoll Qubits rechnen, sobald wir aber mehr brauchen funktionieren sie nicht mehr. Warum das so ist, werden wir in Quest 4 (§4, Seite 78) herausfinden.

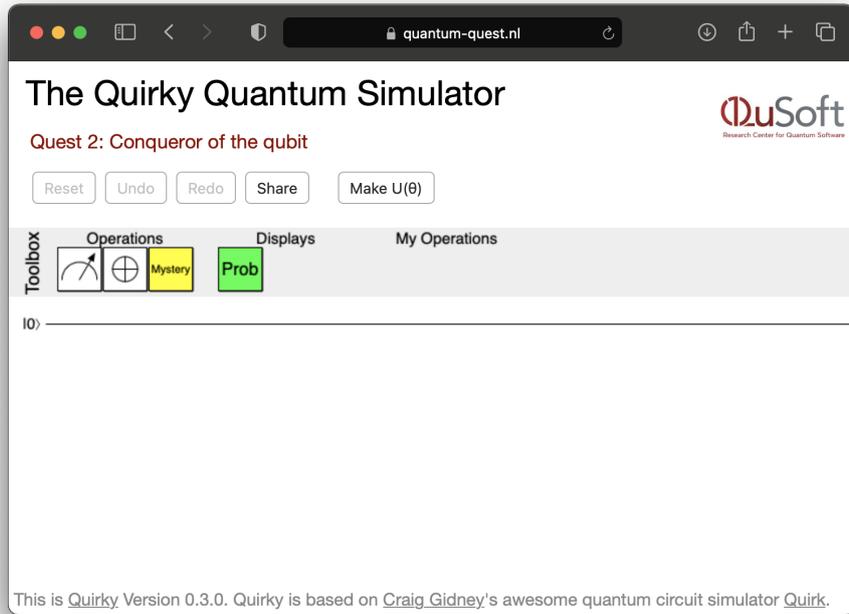


Abbildung 2.3:



Du wirst merken, dass sich die Linie nach dem Messen in eine Doppellinie verwandelt. Das liegt daran, dass in QUIRKY einzelne Linien Qubits entsprechen und Doppellinien eben 'klassischen' Bits. Tatsächlich wissen wir aus §2.2, dass das Messen eines Qubits zu einem Ergebnis von entweder 0 oder 1 mit gewissen Wahrscheinlichkeiten führt, also einem probabilistischen Bit.

Um die Wahrscheinlichkeiten der beiden Möglichkeiten darzustellen, können wir wieder die Wahrscheinlichkeitsanzeige **Prob** benutzen, die wir letzte Woche kennengelernt haben. Füg sie doch einfach mal hinzu:



Es sieht so aus, als wäre das Ergebnis der Messung mit 100% 'Null' (indem du mit der Maus über die Box fährst, siehst du mehr Details). Das entspricht natürlich genau unseren Erwartungen. Wenn wir  $|0\rangle$  messen, sollte das Ergebnis laut den Regeln aus Gl. (1.18) immer 'Null' sein.

Im weiteren Verlauf dieses Kapitels werden wir noch die übrigen Operationen aus der Toolbox besprechen.



## 2.4 Operationen auf einem Qubit

Ein Qubit im Ursprungszustand zu messen ist relativ langweilig, wir wissen ja in welchem Zustand es sich befindet. Aber was für Operationen können wir auf ein Qubit anwenden um den Zustand zu verändern? Wir wollen also aus dem Zustand  $|0\rangle$  durch eine Operation den Zustand  $|\psi(\theta)\rangle$  erzeugen. Wie auch immer die Operation aussieht, sie sollte auf jeden Fall wieder in einem gültigen Zustand resultieren, also den Zustandsraum auf sich selbst abbilden. In Abb. 2.1 hatten wir gesehen, dass der Zustandsraum eines Qubits dem Einheitskreis entspricht, also müssen wir Operationen finden, die diesen Kreis auf sich selbst abbilden.

Lass uns zu Beginn einmal die **NOT-Operation** anschauen, die wir wie in Gl. (1.9) für probabilistische Bits definieren können:

$$\text{NOT } |0\rangle = |1\rangle, \quad \text{NOT } |1\rangle = |0\rangle.$$

Wie können wir NOT auf beliebige Qubit Zustände erweitern? Genau wie für probabilistische Bits in §1.2.1 nutzen wir das Konzept der Linearität. Wenn die Operation  $M$  auf  $|0\rangle$  und  $|1\rangle$  definiert ist, können wir sie auf einem beliebigen Zustand definieren als

$$M(\psi_0 |0\rangle + \psi_1 |1\rangle) = \psi_0 M|0\rangle + \psi_1 M|1\rangle. \quad (2.8)$$

Wir können Gl. (2.8) auch in der Vektornotation ausschreiben:

$$M \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix} = M \left( \psi_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \psi_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) = \psi_0 M \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \psi_1 M \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (2.9)$$

Wie wir schon bemerkt hatten, wird eine Operation, die diese Eigenschaft erfüllt, **linear** genannt und eine Operation auf diese Weise zu erweitern nennt man **erweitern durch Linearität**. Wenn wir also wissen, dass  $M$  linear ist und wie  $M$  sich auf  $|0\rangle$  und  $|1\rangle$  auswirkt, können wir herleiten wie sich  $M$  auf beliebige Zustände auswirkt!

In Gl. (2.8) haben wir nur die Vektoren  $|0\rangle$  und  $|1\rangle$  betrachtet. Im Allgemeinen gilt sogar

$$M(a|\psi\rangle + b|\phi\rangle) = aM|\psi\rangle + bM|\phi\rangle \quad (2.10)$$

für beliebige Vektoren  $|\psi\rangle, |\phi\rangle$  und Zahlen  $a, b$ . Verstehst du, wie Gl. (2.10) aus Gl. (2.8) folgt?

Die Gesetze der Quantenmechanik garantieren, dass jede lineare Operation  $M$  eine Qubit Operation ist – solange sie den Qubit Zustandsraum auf sich selbst abbildet! Das soll heißen, dass jeder Zustand eines Qubits (Punkt auf dem Kreis) auf einen anderen Zustand (Punkt auf dem Kreis) abgebildet wird.

Im Falle der NOT-Operation erhält man nach erweitern durch Linearität

$$\text{NOT}(\psi_0 |0\rangle + \psi_1 |1\rangle) = \psi_0 |1\rangle + \psi_1 |0\rangle, \quad \text{oder} \quad \text{NOT} \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix} = \begin{pmatrix} \psi_1 \\ \psi_0 \end{pmatrix}. \quad (2.11)$$

Bemerke, dass Gl. (2.8), (2.9) und (2.11) genau wie Gl. (1.10) und (1.12) aussehen – nur das  $\psi_0$  und  $\psi_1$  jetzt auch negativ sein können. Auf Abb. 2.2 bezogen, stellt die NOT-Operation eine *Spiegelung* an der 45-Grad-Achse dar (das gilt auch für probabilistische Bits). In Abb. 2.4 ist die Operation grafisch dargestellt. Offensichtlich bildet die NOT-Operation den Zustandsraum (Einheitskreis) auf sich selbst ab, ist also eine gültige Quanten-Operation.

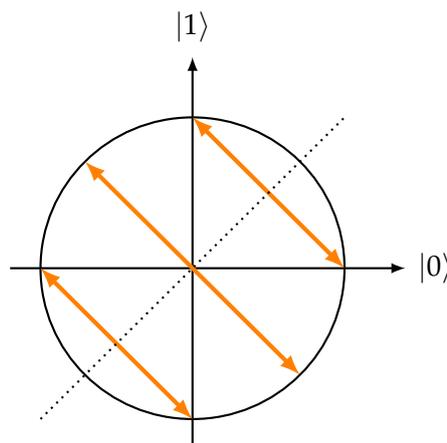


Abbildung 2.4: Die NOT-Operation auf Qubits, definiert in Gl. (2.11), entspricht einer Spiegelung an der 45 Grad (oder  $\pi/4$ ) Achse (gestrichelt).

In QUIRKY sieht die NOT-Operation auf Qubits genau wie die klassische Version aus, also so:  $\oplus$ . Probier mal, die folgende Schaltung zu bauen:



Jetzt scheint das Messergebnis mit einer Wahrscheinlichkeit von 100% 'Eins' zu sein. Tatsächlich wird also der ursprüngliche  $|0\rangle$  Zustand durch die NOT-Operation zu  $|1\rangle$  verändert, sodass das Ergebnis nach den Regeln der Messung aus Gl. (1.18) immer 'Eins' sein muss.

Wir können auch andere Qubit-Operationen definieren, indem wir Spiegelungen an anderen Achsen betrachten. Beispielsweise entspricht die **Z-Operation**, definiert durch

$$Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle, \quad (2.12)$$

einer Spiegelung an der horizontalen  $|0\rangle$ -Achse. Wenn wir Z durch Linearität erweitern, erhalten wir tatsächlich

$$Z \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix} = \begin{pmatrix} \psi_0 \\ -\psi_1 \end{pmatrix},$$

was offensichtlich Qubit-Zustände auf Qubit-Zustände abbildet.

### Hausaufgabe 2.2: Die Z-Operation

Betrachte folgende zwei Qubit Zustände:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

1. Berechne  $Z|+\rangle$  und  $Z|-\rangle$ .
2. Stelle die Z-Operation grafisch auf dem Einheitskreis dar, wie in Abb. 2.4.

### Übungsaufgabe 2.2: Linearität genügt nicht

Betrachte die MAD-Operation, indem du  $MAD|0\rangle = |0\rangle$  und  $MAD|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  durch Linearität erweiterst. Suche einen Qubit Zustand  $|\psi\rangle$ , sodass  $MAD|\psi\rangle$  kein gültiger Qubit-Zustand ist. Zeige also, dass MAD keine gültige Operation auf Qubits ist!



## 2.4.1 Rotationen

Bisher wissen wir nur, wie man die Zustände  $|0\rangle$  und  $|1\rangle$  mit QUIRKY baut. Quanteninformatik wäre aber nicht besonders interessant, wenn das unsere einzigen Optionen wären! Um spannendere Zustände zu generieren, müssen wir uns andere Operationen ausdenken.

Eine naheliegende Operation ist, den Kreis um einen festgelegten Winkel zu *drehen* (rotieren). Lass uns die *Rotation* um den Winkel  $\theta$  einfach  $U(\theta)$  nennen. Du kannst immer annehmen, dass der Winkel in  $[0, 2\pi)$  liegt. Da  $|0\rangle = |\psi(0)\rangle$  und  $|1\rangle = |\psi(\frac{\pi}{2})\rangle$  gilt, wirkt die Operation wie folgt auf die Basisvektoren (siehe Abb. 2.5):

$$U(\theta)|0\rangle = |\psi(\theta)\rangle, \quad U(\theta)|1\rangle = |\psi(\theta + \frac{\pi}{2})\rangle. \quad (2.13)$$

Wir können diese Definition also in Vektornotation so schreiben:

$$U(\theta) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}, \quad U(\theta) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix}, \quad (2.14)$$

wobei wir ausgenutzt haben, dass  $\cos(\theta + \frac{\pi}{2}) = -\sin \theta$  und  $\sin(\theta + \frac{\pi}{2}) = \cos \theta$ .

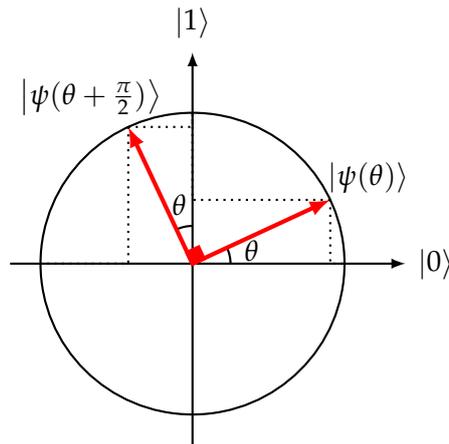


Abbildung 2.5: Die Zustände  $|0\rangle$  und  $|1\rangle$  um den Winkel  $\theta$  gedreht, siehe Gl. (2.13).

Genau wie zuvor auch, können wir die Linearität ausnutzen, um  $U(\theta)$  auf allgemeine Qubit-Zustände zu erweitern. In der folgenden Übung wirst du zeigen, dass die daraus folgende Operation  $U(\theta)$  tatsächlich einer Rotation von Qubit-Zuständen entspricht. Konkret bedeutet das, dass  $U(\theta)$  den Zustandsraum auf sich selbst abbildet, also eine gültige Operation ist.

### Übungsaufgabe 2.3: Qubit-Rotationen

1. Berechne  $U(\alpha) \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix}$  mit Gl. (2.8) und (2.13).
2. Nutze die Definition von  $|\psi(\theta)\rangle$  aus Gl. (2.5) um zu prüfen, dass für alle Winkel  $\alpha$  und  $\beta$  folgendes gilt:

$$U(\alpha) |\psi(\beta)\rangle = |\psi(\alpha + \beta)\rangle. \quad (2.15)$$

Das bedeutet, dass  $U(\theta)$  eine Rotation auf beliebigen Qubit-Zuständen entspricht.

**Hinweis:** Die trigonometrischen Regeln für Winkelsummen und -differenzen könnten hilfreich sein:

$$\sin(\alpha \pm \beta) = \sin \alpha \cos \beta \pm \cos \alpha \sin \beta, \quad \cos(\alpha \pm \beta) = \cos \alpha \cos \beta \mp \sin \alpha \sin \beta. \quad (2.16)$$

Beachte dass eine Rotation um 90 Grad (also  $\pi/2$ ) nicht einer Spiegelung entspricht. Zwar bilden sowohl die NOT-Operation als auch  $U(\pi/2)$  den Zustand  $|0\rangle$  auf  $|1\rangle$  ab, aber auf  $|1\rangle$  haben sie unterschiedliche Wirkungen:

$$\text{NOT } |1\rangle = |0\rangle, \quad U(\pi/2) |1\rangle = -|0\rangle.$$

Wie können wir eine solche Rotation in QUIRKY ausführen? Da es unendlich viele Rotationen  $U(\theta)$  gibt, können nicht alle in der Toolbox enthalten sein. Stattdessen kannst du die Rotationen, die du benötigst, selbst in die Toolbox hinzufügen! Probier doch einfach mal, eine Rotation um  $30^\circ$  hinzuzufügen. Tippe zunächst auf 'Make  $U(\theta)$ ' im Menü. Ein neues Fenster mit einem Eingabefeld für den Winkel sollte sich öffnen:



Gib  $\pi/6$ , was  $30^\circ$  entspricht, ein und bestätige mit der OK-Taste. Glückwunsch! Du hast erfolgreich die  $U(\pi/6)$ -Rotation in die Toolbox hinzugefügt, welche jetzt so aussieht:



Die neue Rotation können wir mit folgendem Schaltkreis erst einmal in QIRKY testen:



Lass uns noch schnell überprüfen, ob diese Ausgabe Sinn ergibt. Wir haben mit dem  $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  Zustand gestartet. Nach Gl. (2.14) bildet  $U(\theta)$  den Zustand  $|0\rangle$  auf  $|\psi(\theta)\rangle = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}$  ab. Für unseren Fall gilt  $\theta = \pi/6$  und damit

$$|\psi(\pi/6)\rangle = \begin{pmatrix} \cos(\pi/6) \\ \sin(\pi/6) \end{pmatrix} = \begin{pmatrix} \sqrt{3}/2 \\ 1/2 \end{pmatrix}.$$

Nach den Regeln der Quantenmessung aus Gl. (2.7) können wir schließen, dass die Wahrscheinlichkeit für das Ergebnis 1 wie folgt ist

$$p_1 = \left(\frac{1}{2}\right)^2 = \frac{1}{4} = 25\%,$$

was genau der Ausgabe von QIRKY entspricht. In der folgenden Hausaufgabe wirst du QIRKY nutzen um den Effekt von  $U(\theta)$  auf den anderen Basisvektor  $|1\rangle$  herauszufinden.

### Hausaufgabe 2.3: Die $30^\circ$ -Rotation testen

1. Baue folgende Abfolge von Operationen in QIRKY: Bereite zunächst den Zustand  $|1\rangle$  vor, rotiere dann um den Winkel  $\pi/6$  und messe schlussendlich das Qubit.
2. Nutze QIRKY's Wahrscheinlichkeitsanzeige um die Ergebnisse der Messung anzuzeigen. Zeige, dass die Ausgabe von QIRKY korrekt ist.
3. Verändere deinen Schaltkreis so, dass man mit Wahrscheinlichkeit 42% das Messergebnis 1 erhält.



## 2.4.2 Zusammensetzen von Quanten-Operationen

Um eine neue Quanten-Operation zu erhalten, können wir zwei vorhandene Operationen  $M$  und  $N$  einfach zusammensetzen. Angenommen, der ursprüngliche Zustand ist  $|\psi\rangle$ , dann erhält man durch Anwenden von  $M$  einfach  $M(|\psi\rangle) = M|\psi\rangle$ . Danach  $N$  anzuwenden resultiert dann im Zustand  $N(M|\psi\rangle)$ . Wir benennen diese **zusammengesetzte** (engl. *composite*) Operation  $NM$ , sodass

$$NM|\psi\rangle = N(M|\psi\rangle).$$

Achte darauf, dass du nicht die Reihenfolge der Operationen vertauscht. Wenn die zusammengesetzte Operation  $NM$  ist, wird zunächst  $M$  und danach  $N$  angewandt! Das liegt daran, dass  $M$  direkt neben dem  $|\psi\rangle$  steht, sich also zuerst auf den Zustand auswirken muss.

### Übungsaufgabe 2.4: Linearität einer zusammengesetzten Operation (optional)

Zeige dass  $NM$  auch linear ist.

**Hinweis:** Nutze Gl. (2.10).

Genauso können wir auch drei oder mehr Qubit-Operationen zusammensetzen. Dann schreiben wir  $ONM$  und so weiter. Zum Beispiel kann man neue Operationen durch das Zusammensetzen von Rotationen und Spiegelungen bilden, darüber sprechen wir später in §2.4.3.

Eine interessante Bemerkung ist, dass alle bisherigen Qubit-Operationen **invertierbar** sind. Das bedeutet, dass für jede Operation  $M$  eine andere Operation, die wir  $M^{-1}$  nennen, sodass zuerst  $M$  und danach  $M^{-1}$  anwenden, oder umgekehrt, den Zustand nicht verändert.<sup>10</sup> In Formeln können wir schreiben

$$M^{-1}M = MM^{-1} = I, \quad (2.17)$$

wobei  $I$  der **Identität** entspricht, die die "triviale" Eigenschaft

$$I|0\rangle = |0\rangle, \quad I|1\rangle = |1\rangle \quad (2.18)$$

erfüllt. (Wir hätten  $I$  auch als  $U(0)$  definieren können, der Rotation um  $0^\circ$ .) Also gilt  $I|\psi\rangle = |\psi\rangle$  für die durch Linearität erweiterte Operation.

Als Beispiel können wir uns das Inverse der Operation  $U$  anschauen. Geometrisch ergibt es Sinn, dass erst um  $\beta$  und anschließend um  $-\beta$  rotieren den Zustand unverändert lässt. Um das formal zu zeigen, müssen wir nur zweimal Gl. (2.15) anwenden:

$$U(-\beta)U(\beta)|\psi(\alpha)\rangle = U(-\beta)|\psi(\alpha + \beta)\rangle = |\psi(\alpha + \beta - \beta)\rangle = |\psi(\alpha)\rangle$$

und genauso wenn wir zuerst um  $-\beta$  und anschließend um  $\beta$  rotieren. Das bedeutet, dass die inverse Operation von  $U(\beta)$  einfach  $U(-\beta)$  ist:

$$U(\beta)^{-1} = U(-\beta).$$

Da die NOT-Operation einer Spiegelung entspricht, ist es klar, dass ein Zustand nach zweimaligen anwenden unverändert ist. Aus Gl. (1.9) folgt tatsächlich

$$\text{NOT NOT } |0\rangle = \text{NOT } |1\rangle = |0\rangle \quad \text{und} \quad \text{NOT NOT } |1\rangle = \text{NOT } |0\rangle = |1\rangle.$$

Aus Linearität folgt, dass  $\text{NOT NOT } |\psi\rangle = |\psi\rangle$  für jeden Zustand  $|\psi\rangle$  gilt, also ist NOT nicht nur invertierbar, sondern sogar *selbstinvers*, also  $\text{NOT}^{-1} = \text{NOT}$ .

### Übungsaufgabe 2.5: Das Inverse einer zusammengesetzten Operation

Zeige dass  $NM$  invertierbar ist, wenn  $M$  und  $N$  invertierbar sind. Beschreibe das Inverse der zusammengesetzten Operation  $(NM)^{-1}$  durch die einzelnen Inverse  $N^{-1}$  und  $M^{-1}$ .

Tatsächlich lässt sich zeigen, dass jede Operation, die den Zustandsraum eines Qubits auf sich selbst abbildet notwendigerweise invertierbar ist. Das ist der Fall für Rotationen  $U(\theta)$ , wie wir schon gesehen haben, und wird auch für Spiegelungen  $V(\theta)$  gelten, wie du gleich sehen wirst. Bei probabilistischen Bits hatten wir dagegen gesehen, dass beispielsweise der zufällige Flip  $F(1/2)$  jeden Zustand auf die Gleichverteilung  $(\frac{1}{2})$  abgebildet hat (siehe Übungsaufgabe 1.6) und damit nicht invertierbar ist.



### 2.4.3 Spiegelungen

Jede Qubit-Operation ist entweder eine Rotation oder eine Spiegelung (Reflektion). Mit Rotationen aus Gl. (2.13) haben wir uns schon ein wenig auseinandergesetzt. Spiegelungen dagegen kennen wir erst zwei: Z und NOT, siehe Gl. (2.11) und (2.12). Aber wie sieht die allgemeinste Spiegelung aus?

Man kann beispielsweise jede Spiegelung bilden, indem man eine bestimmte (sagen wir, die NOT-Spiegelung) betrachtet, und sie mit einer passenden Rotation kombiniert, sodass die Spiegelachse sich korrekt ändert. In der folgenden Übung wirst du zeigen, wie man die Z-Spiegelung aus der NOT-Spiegelung auf zwei Wege erhalten kann.

#### Hausaufgabe 2.4: Z aus NOT

Seien Z, NOT und  $U(\theta)$  die Qubit-Operationen definiert in Gl. (2.11) bis (2.13).

1. Finde einen Winkel  $\theta$ , für den  $Z = U(\theta) \text{ NOT } U(-\theta)$  gilt.
2. Finde einen Winkel  $\theta$ , für den  $Z = \text{NOT } U(\theta)$  gilt.

Kannst du die Abfolge der Transformationen auf dem Einheitskreis visualisieren?

**Hinweis:** Schau dir Abb. 2.4 und die Zeichnung, die du für die Hausaufgabe 2.2 erstellt hast an.

Es stellt sich heraus, dass man *jede* Spiegelung auf eine ähnliche Art und Weise bauen kann. Die allgemeinste Spiegelung hat die Form

$$V(\theta) = \text{NOT } U(\theta) = U(-\theta) \text{ NOT}. \quad (2.19)$$

Eine sehr nützliche Operation ist zum Beispiel die **Hadamard** (nach Jacques Hadamard) Transformation, die sich wie folgt auf die Basiszustände auswirkt (siehe Abb. 2.6):

$$H |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle, \quad H |1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle. \quad (2.20)$$

Diese Operation ist ein Spezialfall der allgemeinen Spiegelung:

$$H = V(\pi/4). \quad (2.21)$$

Zusammengefasst ist jede Qubit-Operation entweder eine Rotation  $U(\theta)$  oder eine Spiegelung  $V(\theta)$ , wobei  $\theta$  ein Winkel ist.

## 2.5 Quantenzustände unterscheiden

Alice schaut einen Wettbewerb mit Eselsrobotern und schreibt auf, ob ihr Lieblingsroboter gewinnt: eine 1 falls er gewinnt, ansonsten eine 0. Diese Information könnte sie auch in einem Qubit kodieren: sie versetzt das Qubit in den Zustand  $|\psi(\theta_0)\rangle$ , falls der Roboter verliert und in Zustand  $|\psi(\theta_1)\rangle$  falls er gewinnt. Das erreicht sie, indem sie entweder  $U(\theta_0)$  oder  $U(\theta_1)$  auf einen Qubit im Zustand  $|0\rangle$  anwendet, wie in Gl. (2.13). Jetzt nehmen wir an, Alice gibt das Qubit an Bob weiter. Kann Bob nur anhand des Qubits raten, welchen Bitwert (0 oder 1) Alice in dem Qubit kodiert hat? Wäre es besser, wenn Bob zuerst eine Rotation oder Spiegelung ausführen könnte? In der folgenden Übungsaufgabe kannst du diese Prinzipien einmal ausprobieren.

<sup>10</sup>Diese Notation und Gl. (2.17) erinnert dich vielleicht an folgendes: Ist  $x$  eine Zahl größer 0, dann ist  $x^{-1} = \frac{1}{x}$  deren Inverses, also gilt  $xx^{-1} = x^{-1}x = 1$ .

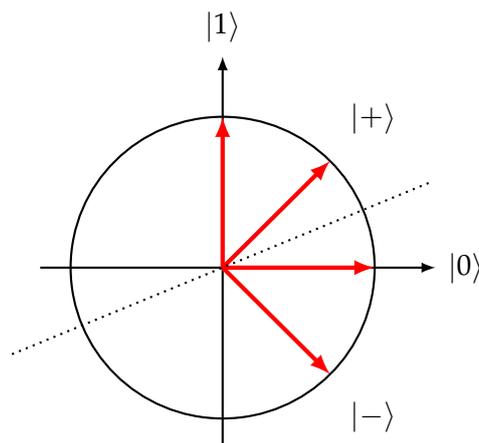


Abbildung 2.6: Die Hadamard-Operation  $H$  auf einem Qubit entspricht einer Spiegelung an der  $45/2$  Grad (oder  $\pi/8$ ) Achse (gestrichelt). Außerdem sind die Zustände  $|0\rangle$ ,  $|1\rangle$ ,  $|+\rangle$ , und  $|-\rangle$  aus Gl. (2.20) dargestellt.

### Übungsaufgabe 2.6: Plus und Minus

Stell dir vor, jemand gibt dir ein Qubit in einem der folgenden Zustände:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Du willst erraten, in welchem der beiden Zustände es sich befindet. Du darfst eine Rotation anwenden und danach messen. Um welchen Winkel solltest du rotieren und mit welcher Wahrscheinlichkeit rätst du korrekt?

Wenn du verschiedene Bitwerte durch verschiedene Quantenzustände darstellen willst, musst du aufpassen, dass du nicht  $|\psi(\theta)\rangle$  und  $|\psi(\theta + \pi)\rangle$  nutzt, da diese nicht unterschieden werden können.

### Übungsaufgabe 2.7: Ununterscheidbare Zustände

Zeige, dass die beiden Zustände  $|\psi(\theta)\rangle$  und  $|\psi(\theta + \pi)\rangle = -|\psi(\theta)\rangle$  nicht unterschieden werden können. Also, dass egal welche Qubit-Operationen du anwendest, bevor du misst, die Messergebnisse immer gleich wahrscheinlich sein werden.

Es ist recht aufschlussreich, die beiden Übungsaufgaben 2.6 und 2.7 zu vergleichen. Wenn sich zwei Zustände durch ein allgemeines Minuszeichen unterscheiden, sind sie ununterscheidbar, wie in Übungsaufgabe 2.7. Die beiden Vektoren  $\pm |\psi(\theta)\rangle$  beschreiben, praktisch gesehen, den *selben* Zustand. Im Gegensatz dazu sind 'relative' Minuszeichen wichtig und können sogar zu perfekt unterscheidbaren Zuständen führen!

In der folgenden Hausaufgabe kannst du herausfinden, wie man zwei *beliebige* Quantenzustände optimal unterscheidet.

### Hausaufgabe 2.5: Zwei Zustände unterscheiden

Seien  $\theta, \theta'$  zwei Winkel. Nimm der Einfachheit halber an, dass  $-\frac{\pi}{2} \leq \theta \leq \theta' \leq \frac{\pi}{2}$ . Angenommen, Eve gibt dir ein einzelnes Qubit, welches sich jeweils mit 50% Wahrscheinlichkeit in Zustand  $|\psi(\theta)\rangle$  oder  $|\psi(\theta')\rangle$  befindet. (Sie könnte beispielsweise eine faire Münze werfen, um zu entscheiden, in welchem Zustand sich das Qubit befinden soll.) Deine Aufgabe ist es nun, herauszufinden, welchen Zustand das Qubit hat. In ein paar Schritten wirst du die

optimale Strategie finden:

1. Wende zunächst die Rotation  $U(\phi)$  um einen Winkel  $\phi$  an. Welche zwei möglichen Zustände erhältst du dann?
2. Führe als nächstes eine Quantenmessung durch und interpretiere das Ergebnis wie folgt: Falls das Ergebnis 0 ist, rätst du, dass das Qubit in Zustand  $|\psi(\theta)\rangle$  war, ansonsten rätst du den Zustand  $|\psi(\theta')\rangle$ . Mit welcher Wahrscheinlichkeit identifizierst du den Zustand korrekt? Schreibe eine Formel mit den Variablen  $\theta, \theta'$  und  $\phi$ .

**Hinweis:** Berechne zunächst die Erfolgswahrscheinlichkeit, angenommen du hast den ersten Zustand bekommen. Dann die Erfolgswahrscheinlichkeit, angenommen du hast den zweiten Zustand bekommen. Und dann erinnere dich daran, dass beide mit Wahrscheinlichkeit 50% auftreten.

3. Du kannst den Rotationswinkel  $\phi$  immer noch clever wählen. Was ist die Erfolgswahrscheinlichkeit als Funktion von  $\theta$  und  $\theta'$ , wenn du  $\phi$  optimal wählst?

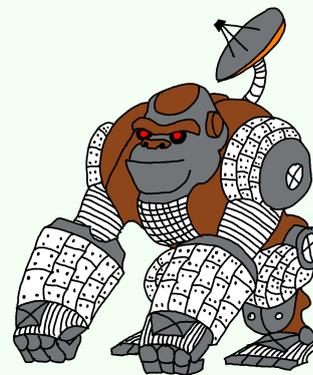
**Hinweis:** Versuche die trigonometrischen Identitäten aus Gl. (2.16) zu verwenden. Insbesondere kannst du mit diesen zeigen, dass

$$\sin^2 \alpha = \frac{1}{2}(1 - \cos(2\alpha)), \quad \cos^2 \alpha = \frac{1}{2}(1 + \cos(2\alpha)). \quad (2.22)$$

Wenn du nicht mehr weiter weißt, kannst du auch [Wolfram Alpha](#) nutzen.

### Übungsaufgabe 2.8: Arm- und Beinbruch (herausfordernd)

Alice und Bob erkunden gerne die Wildnis rund um ihre Stadt. Dafür haben sie zwei große Gorilla-Roboter gebaut, die das unwegsame Gelände navigieren können und sie dabei bequem auf dem Rücken tragen. Aber heute ist kein guter Tag für Bob, sein Roboter ist von einer Klippe gefallen! Zum Glück überlebt Bob den Sturz mit nur ein paar blauen Flecken, aber dem Roboter geht es nicht so gut: ein Arm, ein Bein und das Kommunikationsmodul sind kaputt gegangen. Bob hat leider keine Ersatzteile für die Arme und Beine mitgebracht, schafft es aber zumindest das Kommunikationsmodul kurzzeitig zu reparieren. Dummerweise kann er nur ein einzelnes Qubit senden, bevor es ganz kaputt geht. Bob würde gerne Alice Bescheid geben, welches Bein (links oder rechts) und welcher Arm (links oder rechts) kaputt ist, damit sie das entsprechende Bauteil von ihrem Roboter zu ihm herunter schicken kann. Sie kann ihm aber nur ein Gliedmaß geben, da beide Roboter noch nach Hause laufen können müssen (sie können zum Glück auf drei Beinen laufen). Die Situation ist aber noch komplizierter, da Alice nicht ihr ganzes Werkzeug mitgebracht hat. Bob weiß, dass sie entweder das Werkzeug zum abmontieren von Beinen oder Armen dabei hat – leider kann er sich nicht daran erinnern welches von beiden!



Es gibt vier mögliche Kombinationen, in denen die Arme und Beine gebrochen sein können – du kannst annehmen dass alle mit gleicher Wahrscheinlichkeit  $1/4$  auftreten. Weiterhin gibt es zwei mögliche Werkzeuge, die Alice dabei haben könnte und du kannst annehmen, dass beide mit Wahrscheinlichkeit  $1/2$  auftreten.

### Fragen:

1. Wenn Bob nur ein Bit an Alice senden kann, wie sollte er den Wert auswählen, je nachdem welche der vier Möglichkeiten eingetroffen ist? Wie sollte Alice die Nachricht interpretieren und entscheiden welches Gliedmaß, linkes oder rechtes, sie ihm senden sollte? (Denk daran, dass Alice entweder nur Beine oder nur Arme senden kann und Bob nicht weiß welches von beiden der Fall ist.) Wenn beide die optimale Strategie nutzen, mit welcher Wahrscheinlichkeit interpretiert Alice die Nachricht richtig und sendet das richtige Körperteil an Bob?
2. Was, wenn Bob stattdessen ein Qubit senden kann? Je nach seiner Situation kann er einen von vier Zuständen wählen und Alice kann abhängig von ihrer Situation eine von zwei Rotationen anwenden, bevor sie das Qubit misst. Was ist ihre gemeinsame beste Strategie und welche Erfolgswahrscheinlichkeit hat sie?

Du kannst annehmen, dass Alice und Bob wissen, wie sie ihre Nachrichten interpretieren müssen, da sie im Voraus darüber gesprochen haben, was sie in dieser Notsituation machen sollten.

### 2.5.1 Noch eine mysteriöse Operation

Wir haben noch immer nicht über die gelbe Box in QUIRKY geredet. Im Gegensatz zu der mysteriösen Operation von letzter Woche, welche auf klassischen Bits funktioniert hat, arbeitet diese auf Qubits. Lass uns die mysteriöse Quanten-Operation einfach  $M$  nennen. Wie können wir herausfinden, was in der Box passiert? Als ersten Schritt können wir herausfinden, was  $M|0\rangle$  ist. In QUIRKY können wir diesen Zustand wie folgt generieren:



Man nennt das Problem, einen unbekanntem Quantenzustand herausfinden zu wollen, auch **Quantentomographie**, da man einen unbekanntem Quantenzustand 'von außen' durch verschiedene Messungen bestimmen möchte. Es handelt sich dabei um eine grundlegende Problemstellung, mit denen sich experimentell Forschende täglich auseinandersetzen müssen, wenn sie prüfen wollen, ob das Qubit sich in dem Zustand befindet, in den sie es versetzen wollten!

Wir können schon eine Menge Informationen bekommen, wenn wir eine Messung auf dem unbekanntem Zustand ausführen. Um das klar sehen zu können, schreiben wir

$$M|0\rangle = \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix}.$$

Wenn wir eine Quantenmessung durchführen, erhalten wir laut Gl. (2.6) das Ergebnis 1 mit Wahrscheinlichkeit  $\psi_1^2$ . Das bedeutet, wenn wir das obige Experiment sehr häufig wiederholen, erwarten wir, dass der Anteil der Messungen mit Ergebnis 1 in etwa  $\psi_1^2$  ist. Dieses Verhalten ist komplett analog zu dem mehrfachen Werfen der Münze und anschließendem zählen der Ergebnisse, das wir letzte Woche in §1.3 besprochen haben. Das liefert uns ein Verfahren zur Schätzung von  $\psi_1^2$ . In QUIRKY können wir einfach die Wahrscheinlichkeitenanzeige nach der Messung nutzen, um die Wahrscheinlichkeit vom Ergebnis 1 herauszufinden:



Wir stellen also fest, dass  $\psi_1^2 \approx 11.7\%$ . Da  $M|0\rangle$  ein Einheitsvektor ist, können wir außerdem schlussfolgern, dass  $\psi_0^2 = 1 - \psi_1^2 \approx 88.3\%$ . Jedoch können Amplituden auch negativ sein, also

bestimmt das  $\psi_0$  und  $\psi_1$  nur bis auf das Vorzeichen! Erwinnere dich nun aus Übungsaufgabe 2.7 daran, dass  $|\psi\rangle$  und  $-|\psi\rangle$  ununterscheidbar sind, also können wir  $|\psi\rangle = M|0\rangle$  höchstens bis auf ein allgemeines Vorzeichen bestimmen. Daher bleiben zwei Möglichkeiten:

$$\pm \begin{pmatrix} \sqrt{88.3\%} \\ \sqrt{11.7\%} \end{pmatrix}, \quad \pm \begin{pmatrix} \sqrt{88.3\%} \\ -\sqrt{11.7\%} \end{pmatrix}$$

Beachte, dass diese Situation der aus Übungsaufgabe 2.6 sehr ähnelt, wo wir zwischen  $|+\rangle$  und  $|-\rangle$  unterscheiden mussten. In der letzten Hausaufgabe für diese Woche wirst du genauer herausfinden, wie die mysteriöse Box von Innen aussieht.

### Hausaufgabe 2.6: Zeit für ein Mysterium

1. Wie kannst du zwischen den beiden Optionen unterscheiden? Nutze QUIRKY um den Quantenzustand  $M|0\rangle$  bis auf allgemeines Vorzeichen festzustellen.
2. Bestimme genauso auch den Quantenzustand  $M|1\rangle$ .
3. *Bonusfrage:* Bestimmen die beiden Antworten die Operation  $M$  vollständig? Falls ja, schreibe eine Formel für  $M$  auf. Falls nicht, wie kannst du  $M$  herausfinden?

## 2.6 Exkurs zur Physik (optional)

Unser Hauptfokus der *Quantum Quest* liegt auf der *Mathematik* der Quanteninformatik. Da aber überall auf der Welt schon kleine Quantencomputer in Laboren gebaut werden, ist es auch sinnvoll, etwas über die physikalischen Grundlagen der Quanteninformatik zu wissen. Mithilfe welcher physikalischen Effekte funktionieren Quantencomputer überhaupt?

### 2.6.1 Interferenz

Einer der wichtigsten physikalischen Effekte, den sich Quantencomputer zunutze machen, ist die **Interferenz** – die Interaktion zwischen zwei sich überlappenden Wellen oder Oszillationen. Du kannst Interferenzen zum Beispiel in der Form der Wellen von zwei aneinander vorbeifahrenden Booten beobachten, oder indem du zwei Steine gleichzeitig in einen See wirfst und dir die Wellenmuster anschaust. Wenn sich die Wellen gegenseitig verstärken, nennt man das *konstruktiv*, wenn sie sich dagegen gegenseitig aufheben sind sie *destruktiv* (siehe Abb. 2.7).

Interferenz tritt auch in anderen Umgebungen auf, beispielsweise bei Schallwellen. Vielleicht kennst du ja destruktive Interferenz schon von Noise-Cancelling Kopfhörern. Diese funktionieren, indem sie störende Hintergrundgeräusche aufnehmen und mit umgedrehter Oszillationsrichtung wieder abspielen. Überlagern sich diese zwei Schallwellen dann, so heben sie sich gegenseitig auf:  $1 - 1 = 0$ . Würden die Kopfhörer die Oszillationsrichtung nicht umkehren, würdest du die Geräusche viel lauter hören:  $1 + 1 = 2$ . Dann wären deine Kopfhörer praktisch Hörgeräte!

Mit Qubits können wir im Gegensatz zu probabilistischen Bits *beide* Arten von Interferenz nutzen – konstruktive und destruktive – während probabilistische Berechnungen nur konstruktive Interferenz nutzen können. Mathematisch können wir das mit der Flip-Operation  $F(1/2)$  und der Hadamard-Operation  $H$  aus Gl. (1.14) und (2.20) zeigen:

$$\begin{aligned} F(1/2)[0] &= \frac{1}{2}[0] + \frac{1}{2}[1], & H|0\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, \\ F(1/2)[1] &= \frac{1}{2}[0] + \frac{1}{2}[1], & H|1\rangle &= \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle. \end{aligned} \quad (2.23)$$

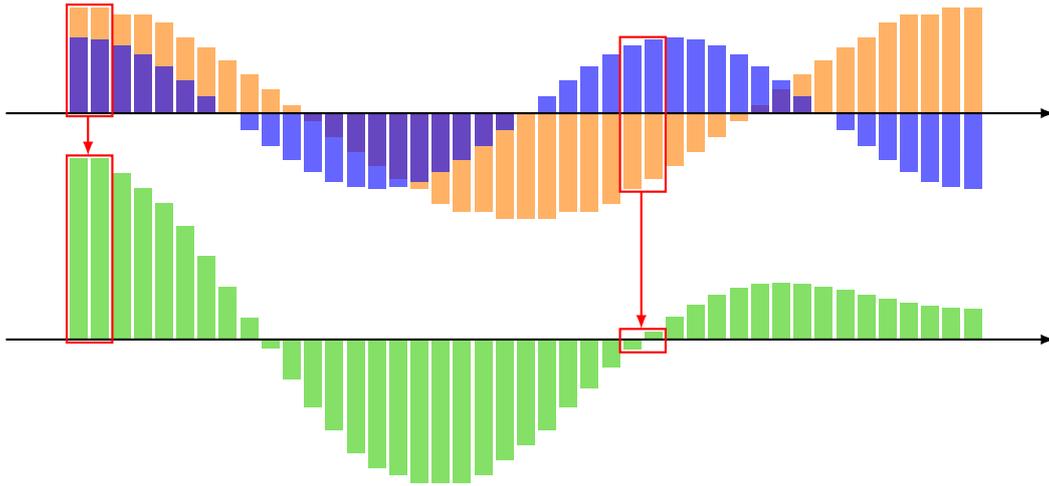


Abbildung 2.7: Interferenz zweier Wellen: die Amplituden der blauen und orangenen Wellen addieren sich an jeder Stelle zu der Amplitude der grünen auf. Wenn beide Wellen das gleiche Vorzeichen haben, ist die Interferenz *konstruktiv* und resultiert in einer noch höheren Amplitude. Haben die Wellen dagegen gegensätzliche Vorzeichen, ist die Interferenz *destruktiv* und die Amplitude schrumpft.

Bis auf die Quadratwurzeln sind beide Operationen fast identisch. Allerdings unterscheidet sich  $F(1/2) [1]$  durch das Plus von  $H [1]$ , welches ein Minus hat. Das scheint nur ein kleiner Unterschied zu sein, aber dieser hat große Folgen.

Lass uns doch einmal anschauen wie sich diese zwei Operationen auf die Gleichverteilung  $\frac{1}{2} [0] + \frac{1}{2} [1]$  und deren quanten-analog, den Plus-Zustand  $|+\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle$  auswirken. Die Flip-Operation  $F(1/2)$  hat folgenden Effekt auf die Gleichverteilung:

$$\begin{aligned}
 F(1/2) \left( \frac{1}{2} [0] + \frac{1}{2} [1] \right) &= \frac{1}{2} F(1/2) [0] + \frac{1}{2} F(1/2) [1] \\
 &= \frac{1}{2} \left( \frac{1}{2} [0] + \frac{1}{2} [1] \right) + \frac{1}{2} \left( \frac{1}{2} [0] + \frac{1}{2} [1] \right) \\
 &= \left( \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \right) [0] + \left( \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \right) [1] \\
 &= \frac{1}{2} [0] + \frac{1}{2} [1],
 \end{aligned}$$

wobei wir die Linearität und Gl. (2.23) angewandt, sowie die Wahrscheinlichkeiten der Zustände  $[0]$  und  $[1]$  zusammengefasst haben. Beachte, dass sich die Wahrscheinlichkeiten für  $[1]$  der beiden Terme zu  $1/2$  verstärken. Das entspricht unserer Intuition: wenn man eine uniform zufällige Münze uniform zufällig flippt, bleibt sie uniform zufällig.

Jetzt betrachten wir aber den Effekt der Hadamard-Operation auf den Plus-Zustand  $|+\rangle$ :

$$\begin{aligned}
 H \left( \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) &= \frac{1}{\sqrt{2}} H |0\rangle + \frac{1}{\sqrt{2}} H |1\rangle \\
 &= \frac{1}{\sqrt{2}} \left( \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) + \frac{1}{\sqrt{2}} \left( \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle \right) \\
 &= \left( \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} \right) |0\rangle + \left( \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} \right) |1\rangle \\
 &= |0\rangle.
 \end{aligned}$$

Die Berechnung ist sehr ähnlich, das Ergebnis aber ein vollkommen anderes – die Amplituden von  $|1\rangle$  heben sich vollständig auf und übrig bleibt nur  $|0\rangle$ . Eine solche destruktive Interferenz

ist mit probabilistischen Bits nicht möglich, da die Wahrscheinlichkeiten immer positiv sind – sie können sich nur verstärken, nicht aufheben.

Zwar sind probabilistische und Quanten-Bits sehr ähnlich, allerdings zeigt das Beispiel, dass sie sich dank destruktiver Interferenz sehr unterschiedlich verhalten. Viele der Quantenüberraschungen, die du in den nächsten Wochen kennenlernen wirst, sind auf die eine oder andere Art auf dieses Phänomen zurückzuführen. Die Existenz von destruktiven Interferenzen ist ein Grund, warum Quantencomputer einen Vorteil gegenüber klassischen Computern haben – sie ermöglicht es dem Quantencomputer die richtige Antwort auszugeben, während falsche Antworten sich durch destruktive Interferenz aufheben. Wie wir in Quests 4 und 5 mit mehr Detail besprechen werden, hilft dabei häufig das Hadamard-Gatter, welches eine wichtige Rolle in Quantenalgorithmen spielt.

## 2.6.2 Polarisation

Da wir uns nun schon ausgiebig mit mathematischen Konzepten von Quantenzuständen und -Operationen beschäftigt haben, können wir uns ja mal die physikalischen Äquivalente anschauen.

Eine der einfachsten physikalischen Repräsentationen eines Qubits ist durch **Polarisation** von Licht. Licht ist eine elektromagnetische Welle, die sich geradlinig im Raum ausbreitet. Diese Welle oszilliert (schwingt) senkrecht zur Ausbreitungsrichtung. Beachte, dass es viele verschiedenen Oszillationsrichtungen geben kann – eine sich vorwärts bewegende Welle kann von links nach rechts oder oben nach unten oszillieren. Diese verschiedenen Zustände können wir als Basiszustände eines Qubits interpretieren:

$$|\leftrightarrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |\updownarrow\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Noch allgemeiner können wir eine Welle, die im Winkel  $\theta$  zur horizontalen Achse schwingt durch den folgenden Zustand darstellen:

$$|\psi(\theta)\rangle = \cos\theta |\leftrightarrow\rangle + \sin\theta |\updownarrow\rangle = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$$

So entspricht diagonal polarisiertes Licht, welches um einen  $45^\circ$  Winkel zwischen der horizontalen und vertikalen Achse oszilliert, dem Zustand  $|\psi(\pi/4)\rangle = |+\rangle$ . Beachte, dass der Winkel der Oszillation der Welle in dieser Darstellung mit dem Winkel des Vektors aus Abb. 2.1 in §2.1.2, der einen Quantenzustand auf dem Einheitskreis darstellt, entspricht.

Um einen dieser Zustände herzustellen, schicken wir einfach einen Lichtstrahl durch einen Polarisationsfilter, wie er in Sonnenbrillen oder 3D Brillen benutzt wird. Ein Polarisationsfilter lässt dabei nur manche Wellen durch – jene, deren Oszillationsrichtung der des Polarisationsfilters entsprechen. Für den Zustand  $|\psi(\theta)\rangle$  können wir also einfach den Filter um den Winkel  $\theta$  von der horizontalen Achse weg drehen. Beispielsweise sind in Abb. 2.8 die Zustände  $|0\rangle$ ,  $|1\rangle$  und  $|+\rangle$  dargestellt.

Eine interessante Eigenschaft der Darstellung von Quantenzuständen durch polarisiertes Licht ist, dass die Zustände  $|\psi(\theta)\rangle$  und  $|\psi(\theta + \pi)\rangle$  auf gleiche Weise hergestellt werden – durch Drehen des Filters um einen Winkel  $\theta$ . Also müssen diese beiden Zustände identisch sein! Damit liefert uns Polarisation eine Intuition, wieso  $|\psi\rangle$  und  $-|\psi\rangle$  ununterscheidbar sein sollten (siehe Übungsaufgabe 2.7).

Eine weitere schöne Eigenschaft der Darstellung von Qubits durch polarisiertes Licht ist, dass wir Messungen leicht visualisieren können. Angenommen, wir wollen den Zustand  $|\psi(\theta)\rangle$  messen und die Wahrscheinlichkeit des Ergebnisses 0 herausfinden. Wenn wir den Zustand als Lichtstrahl gegeben haben, der um den Winkel  $\theta$  polarisiert ist, können wir diesen einfach durch einen horizontalen Polarisationsfilter schicken und messen, wie viel Licht herauskommt

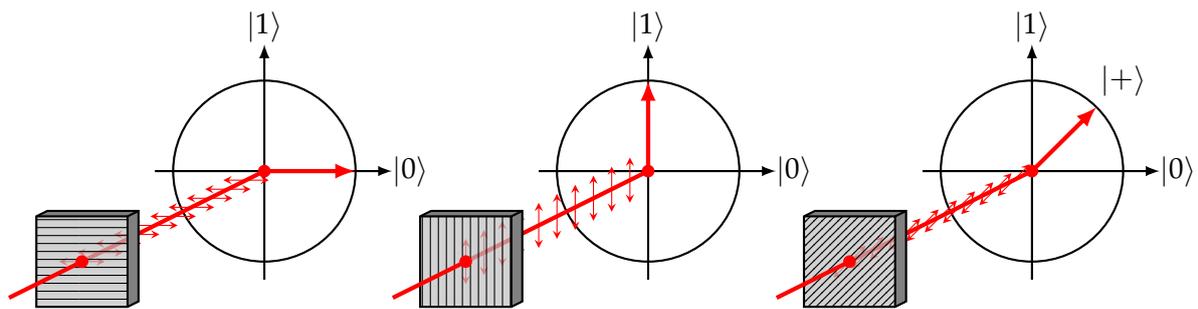


Abbildung 2.8: Horizontal, vertikal und diagonal polarisiertes Licht können die Quantenzustände  $|0\rangle$ ,  $|1\rangle$  und  $|+\rangle$  darstellen.

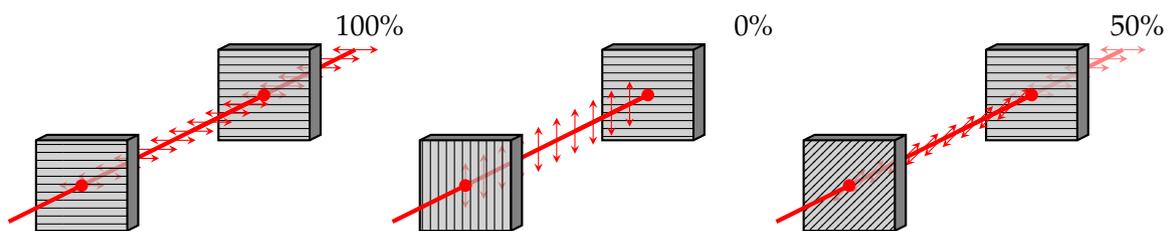


Abbildung 2.9: Man kann die Menge des horizontal polarisierten Lichts messen, indem man es durch einen horizontalen Polarisationsfilter schickt und anschließend die Helligkeit misst. Für jeweils horizontal, vertikal und diagonal polarisiertes Licht sind die Messergebnisse dabei 100%, 0% und 50% der Helligkeit, was den Wahrscheinlichkeiten des Messergebnisses 0 beim Messen der Zustände  $|0\rangle$ ,  $|1\rangle$  und  $|+\rangle$  entspricht.

– wenn sich die Helligkeit auf 70% reduziert, so ist die Wahrscheinlichkeit des Ergebnisses 0 genau 70%. Insbesondere wird ein horizontal polarisierter Lichtstrahl komplett herauskommen, während ein vertikal polarisierter Lichtstrahl nicht hindurchkommt. Einen diagonal polarisierten Lichtstrahl durch einen horizontalen Polarisationsfilter zu schicken, wird in einem 50% dunkleren Lichtstrahl resultieren (siehe Abb. 2.9).

### Übungsaufgabe 2.9: Experimentieren mit Polarisationsfiltern

Falls du eine polarisierte Sonnenbrille Zuhause hast, kannst du sie einmal aufsetzen und auf einen PC- oder Handybildschirm schauen. Normalerweise strahlen Bildschirme polarisiertes Licht aus (wobei die Polarisationsrichtung vom Gerät abhängt). Wenn du also deinen Kopf zur Seite kippst, sollte der Bildschirm heller oder dunkler erscheinen. Kannst du erklären, warum das so ist?

Polarisiertes Licht ist nur eine Methode unter vielen, mit denen man Qubits in Laboren darstellen kann. Ein anderes Beispiel ist die *location* (engl. für "Position") eines lichttransportierenden Teilchens namens Photon – da ein Photon sich nach den Gesetzen der Quantenmechanik verhält, kann es gleichzeitig an zwei Positionen gleichzeitig sein – in *superposition* (engl. für "Überlagerung"). Wenn wir diese Positionen 0 und 1 nennen, entspricht der Zustand des Photons einem Qubit. Es gibt auch noch eine Menge anderer Möglichkeiten: In supraleitenden Schaltungen kann Strom in beide Richtungen gleichzeitig fließen, ein Elektron kann sich in zwei Orbitalen eines Atoms gleichzeitig befinden und so weiter. Kurz gesagt kann sich jedes quantenmechanische System, das sich in zwei unterscheidbaren Zuständen befinden kann, auch in deren Überlagerung befinden, also als physikalische Repräsentation eines Qubits genutzt werden.

## 2.7 Lösungen der Übungsaufgaben

### Lösung Übungsaufgabe 2.1

Beachte, dass

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = |\psi(\pi/4)\rangle, \quad |-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = |\psi(-\pi/4)\rangle.$$

Also müssen die Winkel  $\theta = \pm\pi/4$  sein und die Zustände jeweils 45 Grad über und unter  $|0\rangle$  sein.

### Lösung Übungsaufgabe 2.2

Sei  $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  ein gültiger Quantenzustand. Da  $MAD$  aus erweitern durch Linearität entsteht, impliziert Gl. (2.8), dass

$$\begin{aligned} MAD|\psi\rangle &= MAD\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}MAD|0\rangle + \frac{1}{\sqrt{2}}MAD|1\rangle \\ &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{2}(|0\rangle + |1\rangle) = \left(\frac{1}{\sqrt{2}} + \frac{1}{2}\right)|0\rangle + \frac{1}{2}|1\rangle. \end{aligned}$$

Aber

$$\left(\frac{1}{\sqrt{2}} + \frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = 1 + \frac{1}{\sqrt{2}} \neq 1,$$

also ist  $MAD|\psi\rangle$  kein gültiger Quantenzustand.

### Lösung Übungsaufgabe 2.3

1.  $U(\alpha)$  wirkt sich wie folgt auf einen beliebigen Zustand aus:

$$\begin{aligned} U(\alpha) \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix} &= U(\alpha) \left( \psi_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \psi_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) = \psi_0 \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} + \psi_1 \begin{pmatrix} -\sin \alpha \\ \cos \alpha \end{pmatrix} \\ &= \begin{pmatrix} \psi_0 \cos \alpha - \psi_1 \sin \alpha \\ \psi_0 \sin \alpha + \psi_1 \cos \alpha \end{pmatrix}. \end{aligned}$$

2. Da  $|\psi(\beta)\rangle = \begin{pmatrix} \cos \beta \\ \sin \beta \end{pmatrix}$ ,

$$\begin{aligned} U(\alpha) |\psi(\beta)\rangle &= U(\alpha) \begin{pmatrix} \cos \beta \\ \sin \beta \end{pmatrix} = \begin{pmatrix} \cos \beta \cos \alpha - \sin \beta \sin \alpha \\ \cos \beta \sin \alpha + \sin \beta \cos \alpha \end{pmatrix} = \begin{pmatrix} \cos(\alpha + \beta) \\ \sin(\alpha + \beta) \end{pmatrix} \\ &= |\psi(\alpha + \beta)\rangle. \end{aligned}$$

### Lösung Übungsaufgabe 2.4

Nimm einen beliebigen Zustand  $\psi_0 |0\rangle + \psi_1 |1\rangle$ , nutze zunächst Linearität von  $M$  und anschließend die Linearität von  $N$ :

$$\begin{aligned} NM(\psi_0 |0\rangle + \psi_1 |1\rangle) &= N(M(\psi_0 |0\rangle + \psi_1 |1\rangle)) \\ &= N(\psi_0 M|0\rangle + \psi_1 M|1\rangle) = \psi_0 NM|0\rangle + \psi_1 NM|1\rangle. \end{aligned}$$

Im letzten Schritt haben wir Gl. (2.10) genutzt.

### Lösung Übungsaufgabe 2.5

Es gilt  $(NM)^{-1} = M^{-1}N^{-1}$ , da für jedes  $|\psi\rangle$  gilt

$$M^{-1}N^{-1}NM|\psi\rangle = M^{-1}(N^{-1}N(M|\psi\rangle)) = M^{-1}(M|\psi\rangle) = M^{-1}M|\psi\rangle = |\psi\rangle$$

und

$$NMM^{-1}N^{-1}|\psi\rangle = N(MM^{-1}(N^{-1}|\psi\rangle)) = N(N^{-1}|\psi\rangle) = NN^{-1}|\psi\rangle = |\psi\rangle.$$

### Lösung Übungsaufgabe 2.6

Wende  $U(-\pi/4)$  an und messe. Du kannst die Zustände mit Sicherheit erraten!

### Lösung Übungsaufgabe 2.7

Wir haben bereits gesehen, dass jede Kombination  $M$  an Rotationen und Spiegelungen linear ist. Damit folgt aus  $M|\psi(\theta)\rangle = |\psi(\theta')\rangle = \begin{pmatrix} \cos\theta' \\ \sin\theta' \end{pmatrix}$ , dass  $M(-|\psi(\theta)\rangle) = -|\psi(\theta')\rangle = \begin{pmatrix} -\cos\theta' \\ -\sin\theta' \end{pmatrix}$ . Nach Gl. (2.6) sind die Wahrscheinlichkeiten  $p_0$  und  $p_1$  der Messergebnisse für beide Zustände gleich.

### Lösung Übungsaufgabe 2.9

Wenn du deinen Kopf neigst, änderst du den Winkel zwischen dem Polarisationsfilter in deiner Sonnenbrille und der Richtung in welcher das Licht des Bildschirms polarisiert ist. Da die Menge des Lichts, das durch den Filter kommt, von diesem Winkel abhängt, ändert sich die Helligkeit. Genauso wird ein ändern des Winkels  $\theta$  die Wahrscheinlichkeit des Messergebnisses 0 beim Messen von  $|\psi(\theta)\rangle$  ändern.

## Lösung Übungsaufgabe 2.8

Idealerweise würde Bob gerne 2 Bit senden wollen, um zu zeigen welcher Arm und welches Bein defekt ist. Aber Alice interessiert sich nur für eins dieser Bits, da sie eh nur für eins das richtige Werkzeug dabei hat.

1. Lass uns die zwei möglichen Zustände für jedes der beiden Bits als  $L$  (links) und  $R$  (rechts) nennen. Eine mögliche Strategie für Bob ist der "Mehrheitsentscheid" der beiden Bits. Er könnte also die Zustände wie folgt abbilden:  $LL \mapsto L$ ,  $RR \mapsto R$ . Die beiden anderen Zustände kann er beliebig kodieren, zum Beispiel so:  $LR \mapsto L$ ,  $RL \mapsto R$ . Alice' Strategie besteht dann einfach, das entsprechende Gliedmaß an ihn zu senden (links wenn sie  $L$  erhält,  $R$  ansonsten). Das funktioniert mit Wahrscheinlichkeit

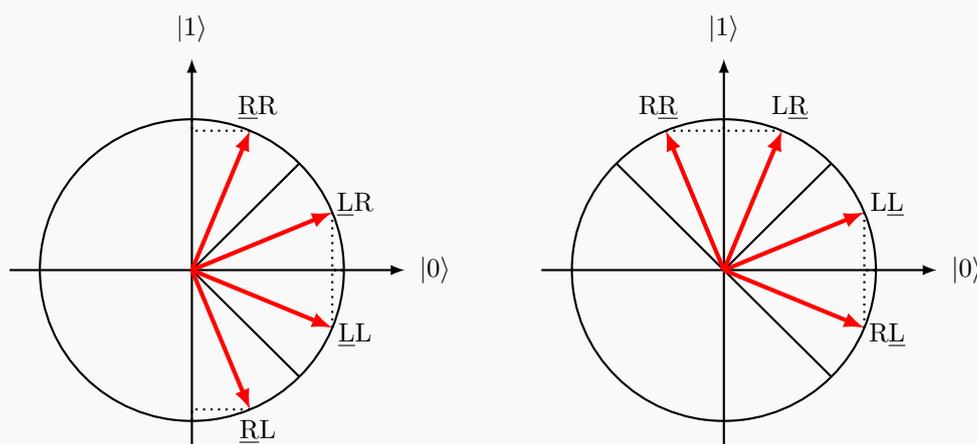
$$\frac{1}{4} \left( 1 + 1 + \frac{1}{2} + \frac{1}{2} \right) = \frac{3}{4} = 0.75, \quad (2.24)$$

wobei die vier Terme in den Klammern den Wahrscheinlichkeiten entsprechen, dass Alice jeweils das richtige Körperteil schickt für jede der vier Situationen.

2. Bob kann die folgenden Quantenzustände senden, je nachdem welche Gliedmaßen defekt sind (LR bedeutet linkes Bein und rechter Arm usw.)

$$\begin{aligned} |LL\rangle &= \cos(\pi/8) |0\rangle - \sin(\pi/8) |1\rangle \\ |LR\rangle &= \cos(\pi/8) |0\rangle + \sin(\pi/8) |1\rangle \\ |RR\rangle &= \cos(3\pi/8) |0\rangle + \sin(3\pi/8) |1\rangle \\ |RL\rangle &= \cos(3\pi/8) |0\rangle - \sin(3\pi/8) |1\rangle \end{aligned}$$

Um aus diesem Zustand das Bit über das Bein zu extrahieren, misst sie einfach. Wenn sie das Bit Information über den Arm haben möchte, wendet sie vor dem Messen noch  $U(\pi/4)$  an. (Beachte, dass sie nicht beide Bits an Information extrahieren kann, da der ursprüngliche Zustand nach dem Messen zerstört ist.) Diese Strategie wird mit Wahrscheinlichkeit  $\cos^2(\pi/8) = \frac{1}{2} + \frac{1}{2\sqrt{2}} \approx 0.85$  erfolgreich sein. Das ist besser als im vorigen Szenario!



## Quest 3: Verzaubernde Verschränkungen

In den letzten zwei Wochen haben wir darüber gesprochen, wie sich ein einzelnes probabilistisches Bit und ein einzelnes Qubit verhält. Diese Woche wirst du lernen, was passiert, wenn du zwei zur Verfügung hast. Zunächst werden wir lernen, wie sich zwei klassische Bits verhalten, welche Zustände sie haben können und wie sie *korrelieren*. Anschließend werden wir uns analog zwei Qubits anschauen und herausfinden, was es heißt, wenn diese *verschränkt* (engl. *entangled*) sind.

### 3.1 Zwei probabilistische Bits

Zwei nebeneinander liegende Münzen können sich in folgenden vier möglichen Zuständen befinden:



Diese entsprechen den vier möglichen Bitstrings<sup>11</sup>: 00, 01, 10, 11.

Der Zustand eines Paares von probabilistischen Bits ist durch die Wahrscheinlichkeitsverteilung über diese vier deterministische Zustände gegeben. Mit anderen Worten, der Zustand wird durch die vier Zahlen  $p_{00}, p_{01}, p_{10}, p_{11} \geq 0$  definiert, wobei  $p_{00} + p_{01} + p_{10} + p_{11} = 1$ . Genau wie im Falle eines einzelnen Bits können wir diesen Zustand in Vektorschreibweise so schreiben:

$$\begin{pmatrix} p_{00} \\ p_{01} \\ p_{10} \\ p_{11} \end{pmatrix}. \quad (3.1)$$

Diese Darstellung ist zwar akkurat, kann aber schnell verwirrend werden, weil man aufpassen muss welche Wahrscheinlichkeit wo steht (kommt  $p_{10}$  zuerst oder doch  $p_{01}$  ?!) und es wird nur komplizierter, je mehr Bits man hat. Viel einfacher ist es, eine Notation zu verwenden, die die Wahrscheinlichkeiten direkt den entsprechenden Bitstrings zuweist. Dafür erweitern wir die Schreibweise aus Gl. (1.3) von einem probabilistischen Bit auf zwei und beschreiben den obigen Zustand so:

$$p_{00}[00] + p_{01}[01] + p_{10}[10] + p_{11}[11]. \quad (3.2)$$

Wenn du möchtest, kannst du diese Gleichung mit den folgenden verallgemeinerungen von Gl. (1.2) auf Gl. (3.1) zurückführen:

$$[00] = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad [01] = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad [10] = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad [11] = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \quad (3.3)$$

Ein Vorteil dieser Notation ist, dass wir im Falle von mehreren Bits Einträge die 0 sind einfach ignorieren können. So können wir zum Beispiel schreiben

$$\frac{1}{2}[00] + \frac{1}{2}[11]$$

<sup>11</sup>Mit einem 'String' bezeichnen wir eine Zeichenkette, in diesem Fall bestehend aus Bitwerten. Wir werden mehrere Münzen oder Qubits hauptsächlich durch solche Strings darstellen.

anstelle von der klobigeren Notation

$$\frac{1}{2}[00] + 0 [01] + 0 [10] + \frac{1}{2}[11].$$

Mit dieser Notation ist es außerdem viel einfacher Messungen und Operationen auf vielen probabilistischen Bits darzustellen. Mit QUIRKY können wir ein bisschen mit zwei probabilistischen Bits herumspielen, denn das Tool hat wieder neue Fähigkeiten bekommen. Gehe zunächst zu

<https://www.quantum-quest.org/quirky>

und klicke auf “Quest 3” und wähle “Two Bits” aus. QUIRKY sollte dann in etwa wie in Abb. 3.1 aussehen. Wie du siehst gibt es jetzt *zwei* Drähte, die zwei Bits entsprechen, welche im Zustand  $[00]$  starten. Vielleicht überrascht es dich, dass das *untere* Bit das erste ist und das *obere* das zweite. Außerdem gibt es eine neue Operation in der Toolbox:  (aber dafür fehlt die Mystery-Box). Die Funktion dieser Box besprechen wir später in diesem Kapitel.

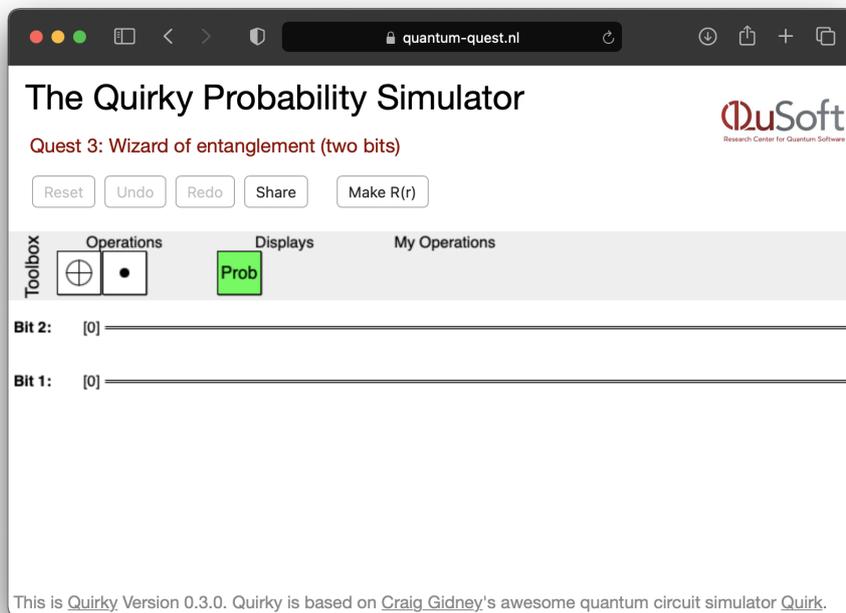
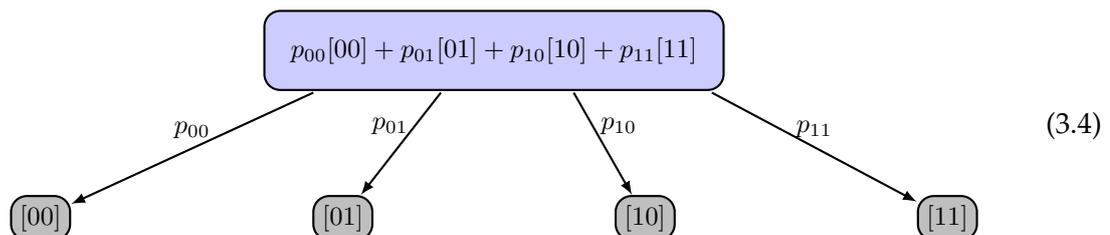


Abbildung 3.1: QUIRKY für Quest 3.

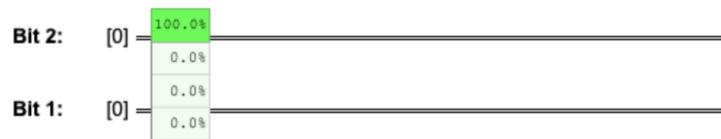
### 3.1.1 Beide Bits messen

Zwei Bits zu messen (oder “anzuschauen”) funktioniert genauso wie in Gl. (1.18) bei einem Bit. Du erhältst einen der vier möglichen Ergebnisse (00, 01, 10, 11) mit der entsprechenden Wahrscheinlichkeit:



Nach der Messung befinden sich die Bits nicht mehr in einer Verteilung über die vier möglichen Zustände sondern in einer Konfiguration, die der gemessenen entspricht. Um diesen

Unterschied darzustellen, verwenden wir hellblau für probabilistische Bits und grau für deterministische (also z.B. gemessene). Wie können wir beide Bits in QUIRKY messen? Wir nutzen einfach wieder die Wahrscheinlichkeitsanzeige:



Beachte, dass die Wahrscheinlichkeitsanzeige automatisch mit *beiden* Drähten verbindet, also die Wahrscheinlichkeiten für *beide* Bits anzeigt. Die Reihenfolge der Wahrscheinlichkeiten entspricht der aus der 4-Vektor Notation in Gl. (3.1). Du musst dir die Reihenfolge aber nicht merken, fahre einfach mit dem Mauszeiger über die Tabelle, um dich an sie zu erinnern (Dankeschön Craig!).

Zum Beispiel erhalten wir beim Messen zweier probabilistischen Bits im Zustand

$$\frac{1}{2}[00] + \frac{1}{2}[11], \quad (3.5)$$

die beiden Ergebnisse 00 oder 11 jeweils mit 50% Wahrscheinlichkeit. Dir fällt vielleicht auf, dass dieser Zustand besonders ist – wenn das Messergebnis des ersten Bits 0 ist, wissen wir automatisch, dass das zweite Bit auch 0 sein muss und genauso für das Ergebnis 1. Da die Messergebnisse beider Bits immer identisch sind, nennen wir die beiden Bits in Zustand Gl. (3.5) **perfekt korreliert**. Wie man so einen Zustand herstellt wirst du gleich lernen.

### 3.1.2 Lokale Operationen

Auf zwei oder mehr probabilistischen Bits kann man eine Vielzahl verschiedener Operationen ausführen. Insbesondere kann man entweder auf allen gleichzeitig eine **globale Operation** oder auf einem einzelnen Bit eine **lokale Operation** ausführen. Zunächst betrachten wir nur lokale Operationen.

Erinnere dich an die NOT-Operation aus §1.2, die ein einzelnes Bit flippt. Was passiert, wenn wir zwei Bits haben, aber NOT nur auf eines davon anwenden? In diesem Fall sollte das erste Bit geflippt werden, während das zweite unverändert bleibt. Das bedeutet, dass die *lokale* NOT-Operation auf dem ersten Bit, die wir  $NOT_1$  nennen, sich wie folgt verhält:

$$NOT_1 [00] = [10], \quad NOT_1 [01] = [11], \quad NOT_1 [10] = [00], \quad NOT_1 [11] = [01]. \quad (3.6)$$

Analog verhält sich die NOT-Operation auf dem zweiten Bit, also  $NOT_2$ , so:

$$NOT_2 [00] = [01], \quad NOT_2 [01] = [00], \quad NOT_2 [10] = [11], \quad NOT_2 [11] = [10]. \quad (3.7)$$

Bisher haben wir die lokale NOT-Operation auf deterministischen Bits beschrieben. Wie sollten wir diese Definition auf probabilistische Bits erweitern? Erwinnere dich daran, dass wir in §1.2.1 gezeigt haben, dass jede vollständig auf deterministischen Bits definierte Operation *durch Linearität* auf probabilistische Bits erweitert werden kann. Zum Beispiel verhält sich  $NOT_2$  auf zwei probabilistischen so:

$$\begin{aligned} NOT_2 (p_{00}[00] + p_{01}[01] + p_{10}[10] + p_{11}[11]) \\ &= p_{00}[01] + p_{01}[00] + p_{10}[11] + p_{11}[10] \\ &= p_{01}[00] + p_{00}[01] + p_{11}[10] + p_{10}[11], \end{aligned}$$

wobei wir im ersten Schritt Gl. (3.7) genutzt haben und anschließend die Terme nach den Zuständen sortiert haben. Das kannst du natürlich auch in der 4-Vektor Notation schreiben, dann ist das Ergebnis aber vielleicht weniger intuitiv:

$$\text{NOT}_2 \begin{pmatrix} p_{00} \\ p_{01} \\ p_{10} \\ p_{11} \end{pmatrix} = \begin{pmatrix} p_{01} \\ p_{00} \\ p_{11} \\ p_{10} \end{pmatrix}. \quad (3.8)$$

### Übungsaufgabe 3.1: NOT<sub>1</sub> in der 4-Vektor Notation (optional)

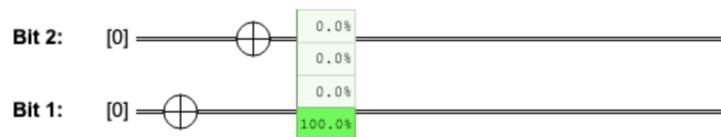
Schreibe die Auswirkung von NOT<sub>1</sub> auf zwei probabilistischen Bits analog zu Gl. (3.8) in der 4-Vektor Notation.

Eine Ein-Bit-Operation kannst du in QUIRKY anwenden, indem du das zugehörige Symbol aus der Toolbox auf den entsprechenden Draht ziehst. Der folgende Schaltkreis generiert zum Beispiel den Zustand [10] und zeigt die Wahrscheinlichkeiten, wenn man beide Bits misst.



Das ergibt Sinn, da der untere Draht in QUIRKY dem ersten Bit entspricht.

Genauso können wir zuerst das eine und anschließend das andere Bit flippen um den Zustand [11] zu erhalten:



Offensichtlich ist die Reihenfolge, in der wir die Operationen anwenden irrelevant. Also können wir sie auch *parallel* anwenden.



Genauso können wir auch zufällige Operationen auf eines der beiden Bits anwenden. Nehmen wir mal an, wir wenden die Operation  $R(r)$ , die das Bit mit Wahrscheinlichkeit  $r$  auf 0 setzt (Gl. (1.13)), auf das erste Bit an. Da  $R(r)[0] = [0]$  gilt, erhalten wir

$$R(r)_1[00] = [00], \quad R(r)_1[01] = [01]. \quad (3.9)$$

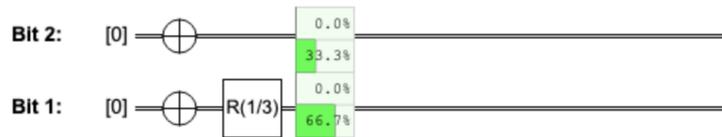
Außerdem gilt  $R(r)[1] = r[0] + (1-r)[1]$ , woraus folgt, dass

$$R(r)_1[10] = r[00] + (1-r)[10], \quad R(r)_1[11] = r[01] + (1-r)[11]. \quad (3.10)$$

Wenn wir also den Zustand [11] haben und auf das erste Bit  $R(1/3)$  anwenden, erhalten wir

$$R(1/3)_1[11] = \frac{1}{3}[01] + \frac{2}{3}[11], \quad (3.11)$$

wie QUIRKY bestätigt:



In der Hausaufgabe schaust du dir das folgende Beispiel an, welches sich noch spannender verhält:



### Hausaufgabe 3.1: $R(r)$ auf dem zweiten Bit

1. Schreibe analog zu Gl. (3.9) und (3.10) die Formeln für  $R(r)_2$  auf.
2. Erkläre, warum das Ergebnis von QUIRKY in Gl. (3.12) richtig ist.

#### 3.1.3 Nur ein Bit messen

Wenn du zwei probabilistische Bits gegeben hast und nur eines der Beiden messen willst, welche Wahrscheinlichkeiten haben dann die zwei möglichen Ergebnisse? Unsere Notation ist in diesem Fall sehr praktisch. Schauen wir uns doch nochmal den allgemeinen Zwei-Bit-Zustand an:

$$p_{00}[00] + p_{01}[01] + p_{10}[10] + p_{11}[11].$$

Um die Wahrscheinlichkeit für das Messergebnis 0 eines Bits zu erhalten, müssen wir nur die Summe über die Wahrscheinlichkeiten aller Zustände bilden, in denen das Bit den richtigen Wert hat.

Zum Beispiel ist die Wahrscheinlichkeit, das Ergebnis 1 beim messen des *ersten* Bits zu erhalten

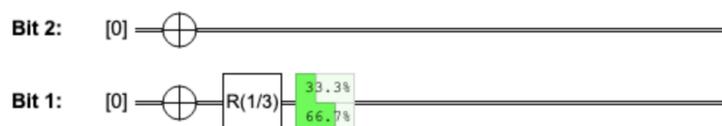
$$p_{10} + p_{11}, \tag{3.13}$$

die den Wahrscheinlichkeiten für die Zustände [10] und [11] aus Gl. (3.4) entsprechen, da das die beiden Bitstrings sind, die mit 1 beginnen. Genauso ist die Wahrscheinlichkeit beim messen des *zweiten* Qubits 0 zu erhalten

$$p_{00} + p_{10},$$

die den Wahrscheinlichkeiten für die Zustände [00] und [10] entsprechen, welches die Bitstrings sind, die auf 0 enden. Wenn man die vier Wahrscheinlichkeiten in einer  $2 \times 2$  Form wie in Abb. 3.2 darstellt wird die Berechnung noch einfacher.

Wir können auch QUIRKY benutzen um die Wahrscheinlichkeiten einer Ein-Bit-Messung anzuzeigen. Verkleinere einfach die Wahrscheinlichkeitsanzeige auf das gewünschte Bit, zum Beispiel so:



Tatsächlich können wir beide Bits einzeln messen und uns beide Messergebnisse gleichzeitig anzeigen lassen:

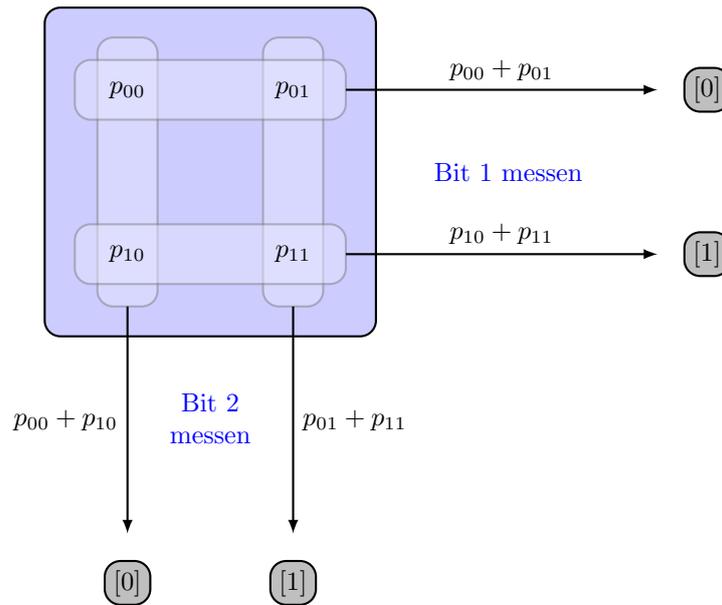
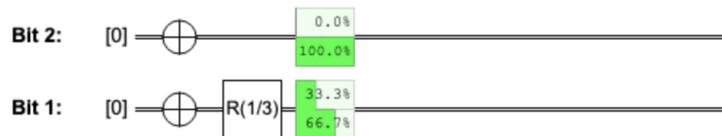


Abbildung 3.2: Wahrscheinlichkeiten der Messergebnisse wenn nur eines der beiden probabilistischen Bits gemessen wird.



Das Ergebnis ist sehr intuitiv. Da die beiden Bits nie “korrelieren” ist klar, dass das erste Bit im Zustand  $\frac{1}{3}[0] + \frac{2}{3}[1]$  und das zweite im Zustand  $[1]$  sein muss.

### 3.1.4 Der Zustand des anderen Bit

Nachdem ein Bit gemessen wurde, befindet sich dieses in dem entsprechenden deterministischen Zustand (genau, wie bei der Messung eines einzelnen Bits in Gl. (1.18)). Aber was ist mit dem anderen Bit, das nicht gemessen wurde? Im Allgemeinen befindet sich dieses nicht in einem deterministischen Zustand. Wenn zum Beispiel im Zustand

$$\frac{1}{2}[10] + \frac{1}{2}[11] \tag{3.14}$$

das erste Bit gemessen wird, erhält man mit Wahrscheinlichkeit  $1/2 + 1/2 = 1$  das Ergebnis 1. Mit anderen Worten, das erste Bit im Zustand aus Gl. (3.14) ist deterministisch, die Wahrscheinlichkeiten beschreiben also nur das zweite Bit. Du kannst dir diesen Zustand also als “Kombination” der beiden getrennten Zustände  $[1]$  und  $\frac{1}{2}[0] + \frac{1}{2}[1]$  vorstellen (in §3.1.7 werden wir uns näher mit dem Kombinieren von probabilistischen Bits beschäftigen). Daher ist es nur logisch, dass das zweite Bit sich nach der Messung in einer Gleichverteilung befindet, also im Zustand

$$\frac{1}{2}[0] + \frac{1}{2}[1].$$

Im Allgemeinen starten wir mit zwei probabilistischen Bits in einem beliebigen Zustand

$$p_{00}[00] + p_{01}[01] + p_{10}[10] + p_{11}[11] \tag{3.15}$$

und messen das erste Bit. Dann wird der Zustand des anderen Bits vom Ergebnis der Messung abhängen. Wenn das Messergebnis beispielsweise 1 war, betrachten wir wieder alle Terme aus

Gl. (3.15), in denen das erste Bit den Zustand 1 hat:

$$p_{10}[0] + p_{11}[1].$$

Dann ignorieren wir das erste Bit, da wir bereits wissen, dass es den Wert 1 hat:

$$p_{10}[0] + p_{11}[1].$$

Und schließlich, da die Summe dieser Wahrscheinlichkeiten nicht zwingend 1 ergibt, teilen wir sie durch ihre Summe  $p_{10} + p_{11}$ :

$$\frac{p_{10}}{p_{10} + p_{11}}[0] + \frac{p_{11}}{p_{10} + p_{11}}[1]. \quad (3.16)$$

Das ist die Wahrscheinlichkeitsverteilung des zweiten Bits, wenn die Messung des ersten Bits 1 ergeben hat (siehe rechts in Abb. 3.3). Die übrigen Fälle sind auch in Abb. 3.3 zusammengefasst.

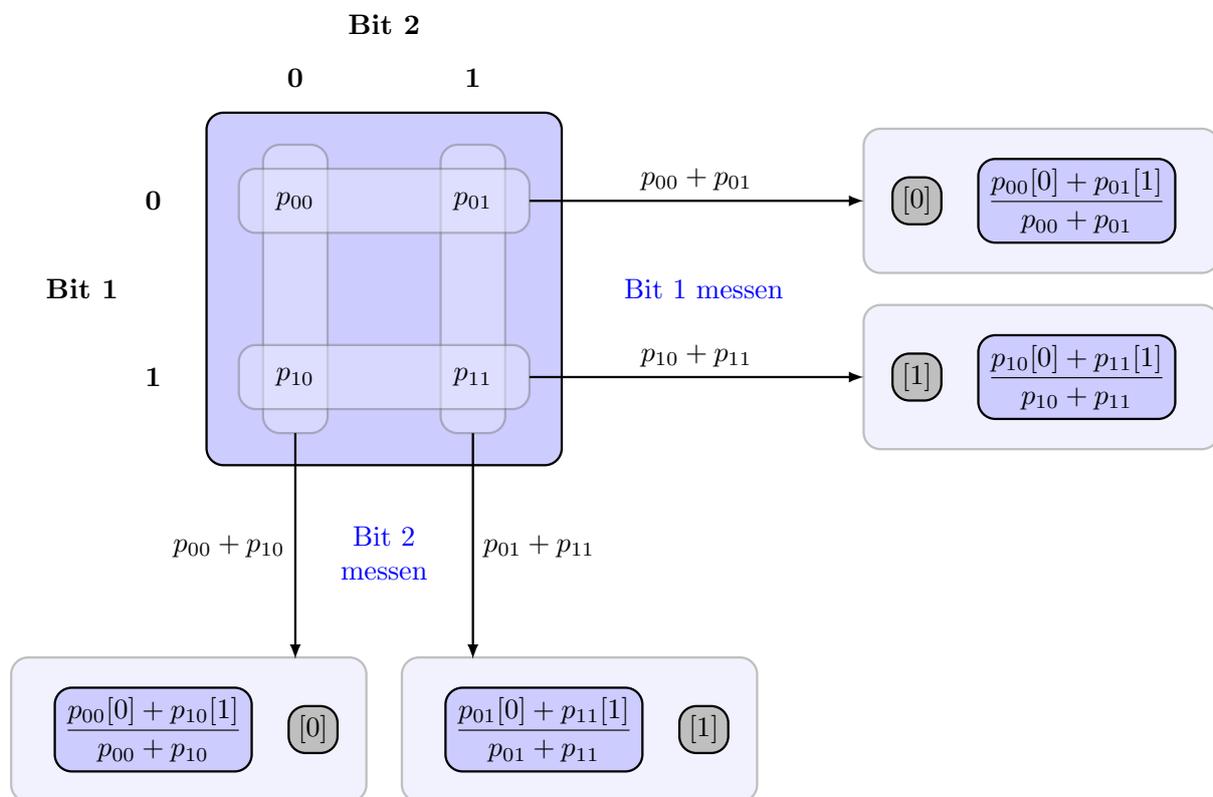


Abbildung 3.3: Wahrscheinlichkeiten der Ergebnisse und die korrespondierenden Zustände des übrigen Bits, nachdem eines von zwei probabilistischen Bits gemessen wurde. Nach der Messung ist das gemessene Bit deterministisch (grau) während das andere probabilistisch bleibt (hellblau).

Um diese Regeln besser zu verstehen, zeigen wir, dass es keinen Unterschied macht, ob man beide Bits gleichzeitig misst oder zunächst das eine und anschließend das andere. So sollte die Wahrscheinlichkeit, 1 vom ersten Bit und 0 vom zweiten Bit zu erhalten einfach  $p_{10}$  sein. Tatsächlich erhalten wir das Ergebnis 1 beim messen des ersten Bits laut Gl. (3.13) mit Wahrscheinlichkeit  $p_{10} + p_{11}$  und mit diesem Ergebnis muss laut Gl. (3.16) die Messung des zweiten Bits 0 mit Wahrscheinlichkeit  $p_{10}/(p_{10} + p_{11})$  ergeben. Damit ist die Gesamtwahrscheinlichkeit für den Fall, dass wir zunächst 1 vom ersten und anschließend 0 vom zweiten Bit erhalten

$$(p_{10} + p_{11}) \times \frac{p_{10}}{p_{10} + p_{11}} = p_{10},$$

was genau dem erwarteten Ergebnis aus Gl. (3.4) entspricht. Die anderen Fälle können analog überprüft werden.

### Übungsaufgabe 3.2: Alice' Münze erraten

**Problem:** Alice ist im Besitz von drei Münzen  $u, q, r$  mit den folgenden Verteilungen:

$$u = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}, \quad q = \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix}, \quad r = \begin{pmatrix} 1/3 \\ 2/3 \end{pmatrix}.$$

Sie führt nun die folgende Sequenz von Würfeln durch:

1. Sie wirft die Münze  $u$ .
2. Je nach Ergebnis wirft sie eine der anderen beiden Münzen:
  - (0) wenn  $u$  das Ergebnis 0 hatte, wirft sie  $q$ ;
  - (1) wenn  $u$  das Ergebnis 1 hatte, wirft sie  $r$ .
3. Alice erzählt Bob das Ergebnis (0 oder 1) des zweiten Münzwurfs (aber nicht, ob es das Ergebnis von  $q$  oder  $r$  ist).

Diese Situation besteht aus *zwei* probabilistischen Bits: Alice' erster Münzwurf und Alice' zweiter Münzwurf (der auch Bob's probabilistischem Bit entspricht).

#### Fragen:

1. Was ist die Wahrscheinlichkeitsverteilung der beiden Würfe von Alice?
2. Was ist die Verteilung, wenn Bob sein probabilistisches Bit misst?
3. Wenn Bob sein Bit misst und als Ergebnis 0 erhält, ist es dann wahrscheinlicher, dass erste Münze 0 oder 1 war? Was wenn er 1 misst?

### 3.1.5 Die SWAP-Operation

Wir wissen bereits, wie man einzelne probabilistischen Bits manipuliert und wollen natürlich auf mehreren probabilistischen Bits gleichzeitig Operationen anwenden. Eine der einfachsten Zwei-Bit Operationen ist die **SWAP-Operation**, die zwei Bits vertauscht:

$$\begin{aligned} \text{SWAP}[00] &= [00], \\ \text{SWAP}[01] &= [10], \\ \text{SWAP}[10] &= [01], \\ \text{SWAP}[11] &= [11]. \end{aligned}$$

SWAP vertauscht also sozusagen die beiden Zustände  $[01]$  und  $[10]$  miteinander und lässt die anderen beiden unberührt. Die obenstehenden Gleichungen können also so zusammengefasst werden:

$$\text{SWAP}[a, b] = [b, a], \quad (3.17)$$

für alle  $a, b \in \{0, 1\}$ , wobei das Komma die die beiden Bits voneinander trennt. Wie sonst auch, können wir SWAP auch durch Linearität auf probabilistische Bits erweitern:

$$\begin{aligned} \text{SWAP}(p_{00}[00] + p_{01}[01] + p_{10}[10] + p_{11}[11]) \\ &= p_{00}[00] + p_{01}[10] + p_{10}[01] + p_{11}[11] \\ &= p_{00}[00] + p_{10}[01] + p_{01}[10] + p_{11}[11]. \end{aligned}$$

### Übungsaufgabe 3.3: SWAP in der 4-Vektorschreibweise (optional)

Schreibe die SWAP-Operation auf zwei probabilistischen Bits in der 4-Vektorschreibweise.

#### 3.1.6 Die kontrollierte-NOT-Operation

NOT und SWAP kann man nutzen, um einzelne Bits und deren Reihenfolge zu ändern. Die Bits interagieren dabei aber nicht miteinander. Um sinnvolle Algorithmen zu Bauen brauchen wir eine Operation, die ein Bit in Abhängigkeit eines anderen verändert. Die CNOT-Operation, oder auch **kontrollierte-NOT** (engl. *controlled-NOT*), ist die einfachste Operation, die genau das macht. Sie flippt das **Zielbit** (engl. *target*) genau dann, wenn das **Steuerbit** (engl. *control*) auf 1 gesetzt ist.

Wenn das erste Bit das Steuer- und das zweite das Zielbit ist, schreiben wir  $CNOT_{1 \rightarrow 2}$ . In diesem Fall gilt:

$$\begin{aligned}CNOT_{1 \rightarrow 2} [00] &= [00], \\CNOT_{1 \rightarrow 2} [01] &= [01], \\CNOT_{1 \rightarrow 2} [10] &= [11], \\CNOT_{1 \rightarrow 2} [11] &= [10].\end{aligned}\tag{3.18}$$

Das entspricht also einem Vertauschen der beiden strings [10] und [11] während die anderen beiden unverändert bleiben.

Man kann sich  $CNOT_{1 \rightarrow 2}$  auch als Addition vorstellen: die beiden Bits werden aufaddiert (modulo 2) und das Ergebnis wird im zweiten Bit gespeichert. Das lässt sich so zusammenfassen:

$$CNOT_{1 \rightarrow 2} [a, b] = [a, a \oplus b],\tag{3.19}$$

für jedes  $a, b \in \{0, 1\}$  wobei " $\oplus$ " die *Addition modulo 2* darstellt und das Komma die beiden Bits trennt. Die Addition modulo 2 funktioniert wie folgt:

$$0 \oplus 0 = 1 \oplus 1 = 0, \quad 0 \oplus 1 = 1 \oplus 0 = 1.\tag{3.20}$$

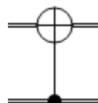
Manchmal nennt man " $\oplus$ " auch **XOR** oder **exklusives Oder** (engl. *exclusive or*), also "entweder-oder", da das Ergebnis nur dann 1 ist, wenn genau eins der beiden Bits 1 ist. Wie üblich kann man  $CNOT_{1 \rightarrow 2}$  durch Linearität von deterministischen Bits auf probabilistische erweitern.

Wir werden auch kontrollierte-NOT Operationen sehen, bei denen das *zweite* Bit das Steuer- und das *erste* das Zielbit ist. Analog schreiben wir diese Operation  $CNOT_{2 \rightarrow 1}$ .

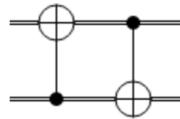
### Übungsaufgabe 3.4: Steuer- und Zielbit vertauschen

1. Schreibe die Operation  $CNOT_{2 \rightarrow 1}$  in einer Formel wie in Gl. (3.19).
2. Wie kann man  $CNOT_{2 \rightarrow 1}$  durch SWAP und  $CNOT_{1 \rightarrow 2}$  implementieren?

Du kannst eine kontrollierte-NOT-Operation in QUIRKY hinzufügen, indem du zunächst die  Box auf dein **Steuerbit** ziehst. Danach ziehst du die  Box auf den Draht, der dem **Zielbit** entsprechen soll. Wenn alles klappt, verbinden sich die beiden Boxen und du hast erfolgreich eine kontrollierte-NOT-Operation erstellt. Hast du zum Beispiel das untere Bit als Steuer- und das obere als Zielbit ausgewählt, solltest du folgendes Ergebnis erhalten:



Da das untere (!) Bit das erste, und das obere das zweite Bit ist, entspricht diese Abbildung der  $\text{CNOT}_{1 \rightarrow 2}$  Operation. Hier siehst du ein komplexeres Beispiel, in dem zunächst  $\text{CNOT}_{1 \rightarrow 2}$  und anschließend  $\text{CNOT}_{2 \rightarrow 1}$  angewendet wird:



### Hausaufgabe 3.2: SWAP aus CNOT's

**Problem:** Es ist beinahe Mitternacht, aber Bob bastelt immer noch an dem Prototypen seines probabilistischen-Bit Computers herum, den er am nächsten Tag in der Schule vorstellen will. Er ist so beschäftigt damit, seinen Zufallszahlengenerator zu kalibrieren, dass er vergisst seinen Papageien Ziggy zu füttern. Um auf sich aufmerksam zu machen, stößt Ziggy Bob's Kaffeetasse um und der ganze Kaffee fließt über Bob's selbstgebastelte Tastatur, mit der er Operationen auslösen kann. Entsetzt stellt Bob fest, dass die SWAP-Taste nicht mehr funktioniert! Zum Glück ist es der CNOT-Taste besser ergangen, sie funktioniert noch immer.



**Fragen:** Wie kann Bob die SWAP-Operation nur durch CNOT-Operationen implementieren? (Wenn du zeigen willst, dass zwei Operationen identisch sind, musst du das dank Linearität nur für die Basiszustände zeigen.)

**Hinweis:** Du solltest drei CNOT-Operationen benötigen.



### 3.1.7 Produkt-Verteilungen

Lass uns nun noch näher anschauen, wie man Zwei-Bit-Systeme aus zwei einzelnen Bits erhält. Nehmen wir einmal an, wir haben die beiden probabilistischen Bits  $q = q_0[0] + q_1[1]$  und  $r = r_0[0] + r_1[1]$  gegeben. Wie können wir diese beiden zu einem Zwei-Bit-System zusammenführen? Bisher haben wir uns diese Frage zwar noch nicht explizit gestellt, allerdings haben wir in §1.1.1 schon gesehen, dass die Wahrscheinlichkeit, dass zwei Ereignisse gleichzeitig auftreten, durch das Produkt ihrer Einzelwahrscheinlichkeiten gegeben ist. So ist die Wahrscheinlichkeit, dass die Bits  $q$  und  $r$  im Zustand  $[00]$  sind  $q_0r_0$ . Genauso ist die Wahrscheinlichkeit, dass sie im Zustand  $[01]$  sind  $q_0r_1$ . Wenn wir also uns alle vier Wahrscheinlichkeiten anschauen, erhalten wir

$$q_0r_0[00] + q_0r_1[01] + q_1r_0[10] + q_1r_1[11]. \quad (3.21)$$

Mit anderen Worten sind die vier Wahrscheinlichkeiten  $p_{ab}$  des kombinierten Zustands

$$p_{00}[00] + p_{01}[01] + p_{10}[10] + p_{11}[11]$$

durch die folgende Gleichung gegeben:

$$p_{ab} = q_a r_b. \quad (3.22)$$

Du kannst einmal kontrollieren, dass die Wahrscheinlichkeiten der Messergebnisse wenn du das erste Bit misst durch  $q$  und die Wahrscheinlichkeiten fürs zweite durch  $r$  gegeben sind (dafür kannst du die Regel aus Abb. 3.2 und  $r_0 + r_1 = 1$  sowie  $q_0 + q_1 = 1$  nutzen). Der Zwei-Bit Zustand hat allerdings eine besondere Eigenschaft: die beiden Bits sind **unabhängig**

(engl. *independent*) voneinander. Das bedeutet, dass du beim messen eines der beiden Bits keine Information über das andere Bit erhältst. In der folgenden Übung kannst du das mal selbst ausprobieren:

### Übungsaufgabe 3.5: Unabhängige Bits (optional)

Nimm an, dass wir das erste Bit aus dem Zustand Gl. (3.21) messen und das Ergebnis als  $a \in \{0, 1\}$  bezeichnen. Zeige, dass der Zustand des zweiten Bit  $r$  ist, unabhängig vom Ergebnis  $a$  des ersten Bits. Mit anderen Worten, die beiden Bits zu kombinieren und das erste Bit zu messen hat den Zustand des zweiten nicht beeinflusst (was es auch nicht sollte)!

Für diese besondere Struktur führen wir eine Notation ein. Und zwar benutzen wir das Symbol " $\otimes$ ", wenn wir zwei probabilistische Bits zu einem einzelnen System aus zwei Bits kombinieren wollen:

$$q \otimes r = (q_0[0] + q_1[1]) \otimes (r_0[0] + r_1[1]) \quad (3.23)$$

Das Symbol " $\otimes$ " heißt **Tensorprodukt** oder auch *Kronecker-Produkt*. Wie können wir also diese seltsame Notation in eine Verteilung von zwei Bits wie in Gl. (3.21) umwandeln? Zunächst stellen wir fest, dass die Operation " $\otimes$ " auf deterministischen Bits einfach ein aneinanderreihen (auch "konkatenerieren") der Zeichenkette ist. Zum Beispiel gilt

$$[0] \otimes [1] = [01]. \quad (3.24)$$

Das ergibt Sinn, da ein Bit im Zustand  $[0]$  und ein anderes Bit im Zustand  $[1]$  zu haben das gleiche ist, wie zwei Bits im Zustand  $[01]$ . Wie immer können wir die uns gut bekannte Linearität um Hilfe bitten, wenn wir diese Regel auf probabilistische Bits erweitern wollen. Durch Linearität können wir beide Terme aus Gl. (3.23) erweitern und anschließend die Konkatenationsregel aus Gl. (3.24) anwenden:

$$\begin{aligned} & (q_0[0] + q_1[1]) \otimes (r_0[0] + r_1[1]) \\ &= q_0 r_0 ([0] \otimes [0]) + q_0 r_1 ([0] \otimes [1]) + q_1 r_0 ([1] \otimes [0]) + q_1 r_1 ([1] \otimes [1]) \\ &= q_0 r_0 [00] + q_0 r_1 [01] + q_1 r_0 [10] + q_1 r_1 [11]. \end{aligned}$$

Beachte, dass wir wieder die Verteilung aus Gl. (3.21) erhalten haben. Mit anderen Worten, wir haben gezeigt, dass die beiden Gleichungen (3.23) und (3.21) identisch sind:

$$(q_0[0] + q_1[1]) \otimes (r_0[0] + r_1[1]) = q_0 r_0 [00] + q_0 r_1 [01] + q_1 r_0 [10] + q_1 r_1 [11]. \quad (3.25)$$

Das bedeutet, dass wir die " $\otimes$ "-Operation tatsächlich konsistent zu unseren vorherigen Überlegungen ist, dass die Wahrscheinlichkeitsverteilung eines Zwei-Bit-Systems durch multiplizieren der Einzelwahrscheinlichkeiten gebildet wird, siehe Gl. (3.22).

Beachte, dass Gl. (3.25) dem Distributivgesetz für Addition und Multiplikation stark ähnelt:

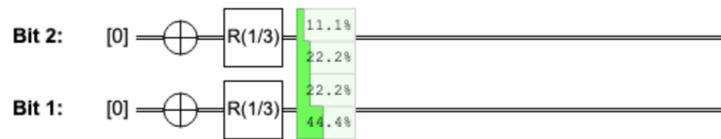
$$(a + b)(c + d) = ac + ad + bc + bd.$$

Der einzige Unterschied ist, dass wir anstelle von Zahlen nun Vektoren und anstelle der Multiplikation die Konkatenationsregel  $[a] \otimes [b] = [a, b]$  nutzen. Wichtig ist jedoch, dass die Reihenfolge der Argumente bei der Konkatenation wichtig ist, während bei der Multiplikation in den meisten Fällen das Kommutativgesetz gilt. Im Allgemeinen ist also  $[a, b] \neq [b, a]$ , da  $[a]$  und  $[b]$  zu konkatenieren nicht das gleiche ist wie  $[b]$  und  $[a]$  zu konkatenieren. Du kannst das übrigens mithilfe der Vektornotation überprüfen, indem du das Tensorprodukt wie folgt schreibst:

$$\begin{pmatrix} q_0 \\ q_1 \end{pmatrix} \otimes \begin{pmatrix} r_0 \\ r_1 \end{pmatrix} = \begin{pmatrix} q_0 \begin{pmatrix} r_0 \\ r_1 \end{pmatrix} \\ q_1 \begin{pmatrix} r_0 \\ r_1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} q_0 r_0 \\ q_0 r_1 \\ q_1 r_0 \\ q_1 r_1 \end{pmatrix},$$

wobei der zweite Ausdruck ein sogenannter Blockvektor ist, dessen beide Einträge jeweils selbst Vektoren sind.

Das Tensorprodukt liefert einen schnellen Weg, um zu verstehen, was im Schaltkreis (3.12) passiert, den wir hier noch einmal wiederholen:



Beachte, dass beide Zustände unabhängig voneinander in den Zustand  $\frac{1}{3}[0] + \frac{2}{3}[1]$  gebracht werden. Daher ist der kombinierte Zustand beider Bits

$$\left(\frac{1}{3}[0] + \frac{2}{3}[1]\right) \otimes \left(\frac{1}{3}[0] + \frac{2}{3}[1]\right) = \frac{1}{9}[00] + \frac{2}{9}[01] + \frac{2}{9}[10] + \frac{4}{9}[11] = \begin{pmatrix} 1/9 \\ 2/9 \\ 2/9 \\ 4/9 \end{pmatrix} \approx \begin{pmatrix} 11.1\% \\ 22.2\% \\ 22.2\% \\ 44.4\% \end{pmatrix},$$

was mit dem Ergebnis von QUIRKY übereinstimmt.

### Hausaufgabe 3.3: Das Tensorprodukt

Finde zwei probabilistische Bits  $q$  und  $r$ , sodass folgende Gleichung gilt:

$$q \otimes r = 0.48[00] + 0.32[01] + 0.12[10] + 0.08[11].$$

Das Tensorprodukt erlaubt uns auch kompaktere Schreibweisen für lokale Operationen zu nutzen. So gilt wenn  $M$  eine Operation auf einem Bit ist, dass

$$M_1([a] \otimes [b]) = M[a] \otimes [b], \quad M_2([a] \otimes [b]) = [a] \otimes M[b]. \quad (3.26)$$

Das stimmt mit den Gleichungen aus §3.1.2 überein.

Da die oben beschriebenen Zwei-Bit-Verteilungen durch ein Produkt von zwei Ein-Bit-Zuständen  $p = q \otimes r$  entstanden sind, heißen sie **Produktzustände** oder **Produkt-Verteilungen**. Wie du in Übungsaufgabe 3.5 gesehen hast, modellieren Produktzustände Situationen, in denen zwei Bits unabhängig voneinander sind, zum Beispiel wenn zwei Münzen geworfen werden. Aber, ist jeder Zwei-Bit-Zustand ein Produktzustand? Interessanterweise werden wir im nächsten Abschnitt sehen, dass dies nicht der Fall ist.



### 3.1.8 Korrelierte Verteilungen

Manche Zwei-Bit-Verteilungen sind *nicht* Produktzustände – sie können nicht wie in Gl. (3.21) oder Gl. (3.23) geschrieben werden, egal welche Werte du für  $q_a$  und  $r_b$  wählst. Wir werden eine Verteilung **korreliert** nennen, wenn sie keine Produktverteilung ist. Ein Beispiel einer korrelierten Verteilung ist

$$\frac{1}{2}[00] + \frac{1}{2}[11]. \quad (3.27)$$

Wir können zeigen, dass dieser Zustand kein Produktzustand ist, indem wir annehmen, wir haben eines der beiden Bits gemessen. Das Ergebnis  $a$  dieser Messung ist komplett zufällig, also entweder 0 oder 1 mit jeweils 50%. Aber sobald wir  $a$  kennen, ist der Zustand des anderen Bits deterministisch – es zu messen würde mit 100% das gleiche Ergebnis liefern. Also ist der Zustand des Bits  $b = a$ , hängt also vom Messergebnis  $a$  des gemessenen Bits abhängig. In Übungsaufgabe 3.5 haben wir gesehen, dass das bei Produktzuständen nicht passieren kann.

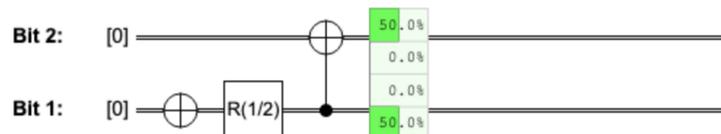
Damit haben wir bewiesen, dass Gl. (3.27) einen korrelierten Zustand beschreibt. Tatsächlich sind die beiden Bits sogar *perfekt* korreliert, da die Messergebnisse zwar zufällig, aber immer identisch sind ( $a = b$ ). Aufgrund dieser Eigenschaft nennen wir Gl. (3.27) ein Paar **perfekt korrelierter Zufallsbits**.

Korrelierte Verteilungen entstehen auf natürliche Weise durch eine Art von Interaktion. Nimm zum Beispiel an, du wirfst eine Münze und schreibst das Ergebnis auf einen Zettel, den du in einem Briefumschlag verschließt und einer anderen Person gibst. Aus der Sicht der anderen Person (die zwar deine Vorgehensweise, nicht aber das Ergebnis des Münzwurfs kennt) lässt sich der Zustand deiner Münze (Bit 1) und des Papiers im Umschlag (Bit 2) durch

$$\frac{1}{2} \begin{array}{c} \text{Kopf} \\ \text{Zahl} \end{array} \otimes \begin{array}{c} \text{Kopf} \\ \text{Zahl} \end{array} + \frac{1}{2} \begin{array}{c} \text{Zahl} \\ \text{Kopf} \end{array} \otimes \begin{array}{c} \text{Zahl} \\ \text{Kopf} \end{array}$$

beschreiben, was nichts anderes als eine unterhaltsame Schreibweise des Zustands in Gl. (3.27) ist.

Wie können wir also in QUIRKY korrelierte Zustände herstellen? Lokale Operationen alleine reichen nicht aus, da diese nur zu Produktzuständen führen. Wir können aber die kontrollierte-NOT-Operation nutzen, um die zwei Bits im folgenden Schaltkreis interagieren zu lassen:



Warum funktioniert das?

### Übungsaufgabe 3.6: Perfekt korrelierte Zufallsbits generieren

Erkläre, warum die obige Berechnung in QUIRKY den Zustand  $\frac{1}{2}[00] + \frac{1}{2}[11]$  generiert.

Wenn du prüfen willst, ob eine beliebige Zwei-Bit-Verteilung

$$p = p_{00}[00] + p_{01}[01] + p_{10}[10] + p_{11}[11]$$

einem Produkt- oder korreliertem Zustand entspricht, kannst du einfach das folgende Maß berechnen:

$$\Delta(p) = p_{00}p_{11} - p_{01}p_{10}. \quad (3.28)$$

Wenn  $\Delta(p) = 0$ , dann ist  $p$  ein Produktzustand; anderenfalls ist es eine korrelierte Verteilung. Zum Beispiel gilt für den Zustand aus Gl. (3.27):

$$\Delta\left(\frac{1}{2}[00] + \frac{1}{2}[11]\right) = \frac{1}{2} \cdot \frac{1}{2} - 0 \cdot 0 = \frac{1}{4} \neq 0,$$

was bestätigt, dass dieser Zustand tatsächlich korreliert ist und keine Produktverteilung. Wenn du jetzt neugierig bist und wissen willst, wieso diese Bedingung korrekt ist, findest du die Erklärung unterhalb. Da sie aber nicht unbedingt erforderlich für das Verständnis ist, kannst du sie auch überspringen.

Wir können den Beweis, dass  $\Delta(p) = 0$  genau dann, wenn  $p$  ein Produktzustand ist, in zwei Schritte unterteilen. Zunächst zeigen wir, dass wenn  $p$  ein Produktzustand ist,  $\Delta(p) = 0$  gelten muss. Wenn  $p = q \otimes r$  ist, gilt tatsächlich

$$\Delta(p) = q_0r_0q_1r_1 - q_0r_1q_1r_0 = 0.$$

Aber was ist mit der Umkehrung – könnte  $\Delta(p) = 0$  sein, auch wenn  $p$  kein Produktzustand ist? Nein, das ist nicht möglich! Um das zu zeigen, nehmen wir zunächst an, dass  $\Delta(p) = 0$  ist und

zeigen, dass dadurch  $p = q \otimes r$ , wobei  $q, r$  probabilistische Bits sind. Wir wählen die Bits wie folgt:

$$\begin{aligned} q &= q_0[0] + q_1[1] = (p_{00} + p_{01})[0] + (p_{10} + p_{11})[1], \\ r &= r_0[0] + r_1[1] = (p_{00} + p_{10})[0] + (p_{01} + p_{11})[1]. \end{aligned} \quad (3.29)$$

Aus Abb. 3.3 geht hervor, dass  $q$  und  $r$  einfach die Verteilung der Messergebnisse sind, wenn wir jeweils das erste oder zweite Bit messen. Nun können wir überprüfen, dass unsere Wahl von  $q$  und  $r$  tatsächlich den Zustand  $p$  produziert:

$$\begin{aligned} q \otimes r &= ((p_{00} + p_{01})[0] + (p_{10} + p_{11})[1]) \otimes ((p_{00} + p_{10})[0] + (p_{01} + p_{11})[1]) \\ &= (p_{00} + p_{01})(p_{00} + p_{10})[00] + \dots \\ &= (p_{00}p_{00} + p_{00}p_{10} + p_{01}p_{00} + p_{01}p_{10})[00] + \dots \\ &= (p_{00}p_{00} + p_{00}p_{10} + p_{01}p_{00} + p_{00}p_{11})[00] + \dots \\ &= p_{00}(p_{00} + p_{10} + p_{01} + p_{11})[00] + \dots \\ &= p_{00}[00] + \dots \\ &= p. \end{aligned}$$

Hier haben wir zunächst Gl. (3.25) verwendet, dann ausmultipliziert, anschließend ausgenutzt, dass  $\Delta(p) = 0$  gilt und somit  $p_{01}p_{10}$  durch  $p_{00}p_{11}$  ersetzt werden kann (siehe Gl. (3.28)). Und zum Schluss haben wir  $p_{00} + p_{01} + p_{10} + p_{11} = 1$  zusammengefasst, da  $p$  eine Wahrscheinlichkeitsverteilung sein muss. Die drei als "...“ abgekürzten Terme können analog behandelt werden. Kannst du die Lücken selbst füllen und einen der Terme alleine verifizieren?

Wir haben bereits in Übungsaufgabe 3.5 gesehen, dass  $p$  kein Produktzustand sein kann, wenn die Messung des ersten Bits den Zustand des zweiten Bits "stört". Tatsächlich gilt auch die Umkehrung dieser Aussage, wie du in der folgenden Hausaufgabe beweisen kannst.

#### Hausaufgabe 3.4: Unabhängigkeit impliziert Produkt (optional)

Nimm an, dass  $p$  sich in einer beliebigen Zwei-Bit-Verteilung befindet, sodass der Zustand des zweiten Bits unabhängig vom Messergebnis des ersten Bit ist. Zeige, dass ein solches  $p$  eine Produktverteilung ist. Das schaffst du in zwei Schritten:

1. Das Messergebnis des ersten Bits kann entweder 0 oder 1 sein. Mit Abb. 3.3 kannst du die verbleibenden Zustände des zweiten Bits in beiden Fällen vergleichen und die folgenden Identitäten zeigen:

$$\frac{p_{00}}{p_{00} + p_{01}} = \frac{p_{10}}{p_{10} + p_{11}}, \quad \frac{p_{01}}{p_{00} + p_{01}} = \frac{p_{11}}{p_{10} + p_{11}}.$$

2. Nutze diese Gleichungen sowie Gl. (3.28), um zu zeigen, dass  $\Delta(p) = 0$

## 3.2 Zwei Quantenbits

Wir können zwei Quantenbits ähnlich wie zwei probabilistische Bits beschreiben. Der größte Unterschied liegt wieder darin, dass wir mit Amplituden anstelle von Wahrscheinlichkeiten arbeiten, wobei diese wieder negativ sein können und anders normalisiert werden (siehe §2.1.1). Ein allgemeiner Zwei-Qubit-Zustand sieht dann so aus:

$$|\psi\rangle = \psi_{00}|00\rangle + \psi_{01}|01\rangle + \psi_{10}|10\rangle + \psi_{11}|11\rangle \quad (3.30)$$

wobei  $\psi_{ij} \in [-1, 1]$  und

$$\psi_{00}^2 + \psi_{01}^2 + \psi_{10}^2 + \psi_{11}^2 = 1.$$

Wir schreiben nun  $|a, b\rangle$  anstelle von  $[a, b]$ , um klarzustellen, dass es sich um Quantenbits und nicht probabilistische Bits handelt. Genau wie wir in Gl. (3.3) die Basiszustände von zwei probabilistischen Bits festgestellt haben, können wir die vier Basisvektoren von  $|a, b\rangle$  identifizieren:

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \quad (3.31)$$

Genau wie in Gl. (3.1), können wir dann den allgemeinen Zwei-Qubit-Zustand aus Gl. (3.30) auch als 4-Vektor schreiben:

$$\begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{10} \\ \psi_{11} \end{pmatrix}.$$

Diese Vektoren werden aber recht schnell unhandlich, weshalb wir hauptsächlich die  $|a, b\rangle$  Schreibweise aus Gl. (3.30) verwenden.

Diese Notation vereinfacht auch das Zusammenfassen mehrerer Quantensysteme. In §3.1.7 haben wir das Tensorprodukt „ $\otimes$ “ genutzt, um zwei unabhängige probabilistische Bits zu einem Zwei-Bit-System zu kombinieren. Die gleiche Operation (nur, dass wir  $[a]$  durch  $|a\rangle$  ersetzen) können wir auch auf Qubits anwenden. Insbesondere können wir die Basisvektoren wieder wie auch schon in Gl. (3.24) zusammensetzen:

$$|0\rangle \otimes |0\rangle = |00\rangle, \quad |0\rangle \otimes |1\rangle = |01\rangle, \quad |1\rangle \otimes |0\rangle = |10\rangle, \quad |1\rangle \otimes |1\rangle = |11\rangle. \quad (3.32)$$

Beachte, dass das dem aneinanderreihen der Zeichenketten entspricht. Das gibt uns eine alternative Möglichkeit, die vier Basisvektoren in Gl. (3.31) zu betrachten.

Wir können das Tensorprodukt wie folgt auf beliebige Ein-Qubit-Zustände  $|\alpha\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$  und  $|\beta\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle$  erweitern:

$$\begin{aligned} |\alpha\rangle \otimes |\beta\rangle &= (\alpha_0 |0\rangle + \alpha_1 |1\rangle) \otimes (\beta_0 |0\rangle + \beta_1 |1\rangle) \\ &= \alpha_0 \beta_0 |00\rangle + \alpha_0 \beta_1 |01\rangle + \alpha_1 \beta_0 |10\rangle + \alpha_1 \beta_1 |11\rangle. \end{aligned} \quad (3.33)$$

Zwei-Qubit-Zustände dieser Form nennen wir **Produktzustände**. In §3.2.3 werden wir sehen, wie man solche Zustände durch Quantenoperationen herstellt. Wichtig ist, dass wie auch bei probabilistischen Bits, nicht alle Zwei-Qubit-Zustände Produktzustände sind.

### Übungsaufgabe 3.7: Tensorprodukt und Produktzustände

Erinnere dich an die Zustände  $|+\rangle$  und  $|-\rangle$  aus Übungsaufgabe 2.1.

1. Schreibe  $|+\rangle \otimes |-\rangle$  in der Form aus Gl. (3.30).
2. Ist der Zustand  $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$  ein Produktzustand?

Im Folgenden werden wir lernen, welche Regeln beim Messen und Manipulieren von zwei Qubits gelten. Zwar sind diese Regeln analog zu den bekannten Regeln von zwei probabilistischen Bits, aber dennoch werden wir neue und unerwartete Phänomene entdecken. Erfreulicherweise hilft uns QUIRKY bei dieser Quest, die Welt der Zwei-Qubit-Systeme zu erkunden. Um zu starten, besuche

<https://www.quantum-quest.org/quirky>

und klicke auf “Quest 3” und anschließend auf “Two Qubits”. Dein Browser sollte dann in etwa wie in Abb. 3.4 aussehen. Im Vergleich zu letzter Woche hat QUIRKY zwei *Quantenbits* die im Zustand  $|00\rangle$  starten. Außerdem gibt es drei neue Boxen in der Toolbox:  $\boxed{Z}$ ,  $\boxed{H}$ , und  $\bullet$  (außerdem fehlt die Mysteriöse Box). Im Laufe dieses Kapitels werden wir diese Operationen besprechen.

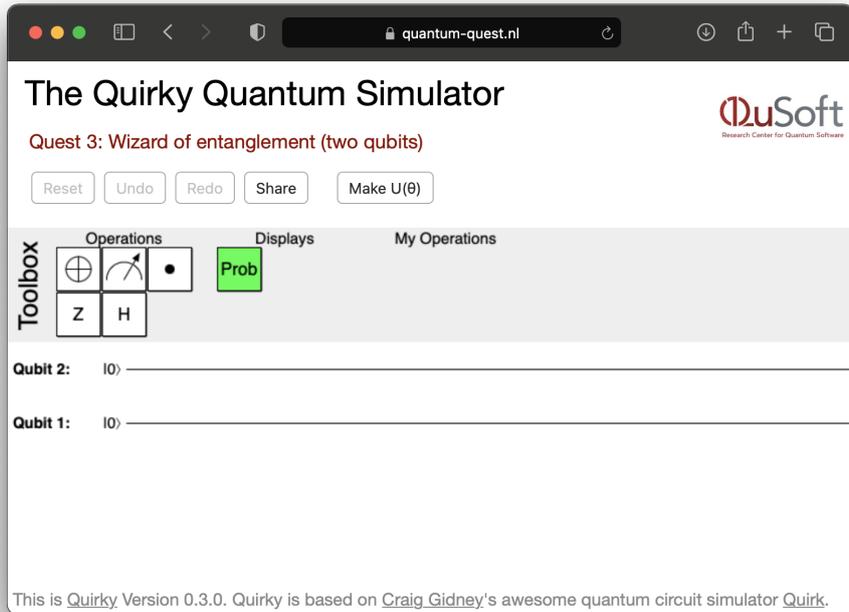
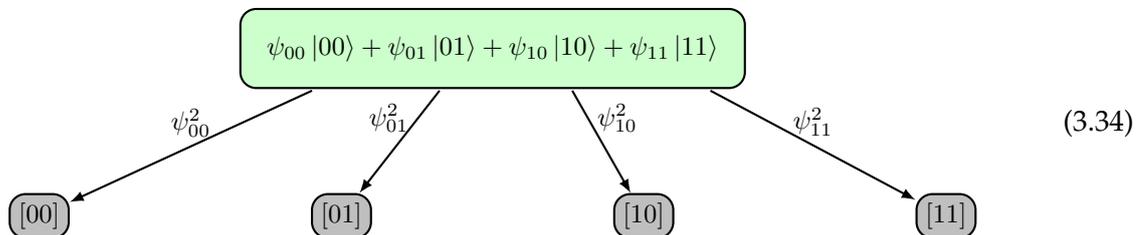


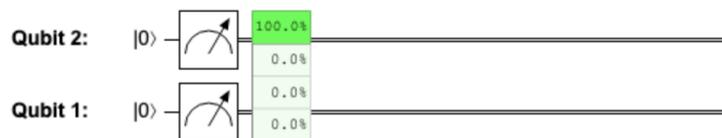
Abbildung 3.4: QUIRKY für Quest 3.

### 3.2.1 Zwei Qubits messen

Das Messen von zwei Qubits ist sehr ähnlich definiert zu dem von einem Qubit (s. Gl. (2.7)). Der Unterschied besteht darin, dass man zwei Bits erhält mit den folgenden Wahrscheinlichkeiten:



Hier bezeichnet der grün hinterlegte Teil die Qubits und der grau hinterlegte Teil die resultierenden deterministischen Bits, die wir nach dem Messen erhalten. Wie können wir beide Qubits in QUIRKY messen? Wir führen einfach auf jedem Qubit eine Messung durch und benutzen die Anzeige zur Wahrscheinlichkeit wie zuvor.



### 3.2.2 Lokale Operationen

Wenn du zwei Qubits hast, führt eine *lokale Operation* nur zu Veränderungen auf einem der Qubits. Jede Operation, die man auf einem Qubit ausführen kann, kann man als lokale Operation betrachten, die auf zwei Qubits operiert, so wie wir dies schon in §3.1.2 getan haben. Erinnern wir uns beispielsweise an die Ein-Bit-NOT-Operation von Gl. (1.9), die wir in lokale NOT Operationen in Gl. (3.6) und (3.7) umgewandelt haben, so können wir dieses analog für die Ein-Qubit-NOT-Operation machen. Die resultierenden lokalen Quantum NOT-Operationen

sehen sehr ähnlich aus:

$$\begin{aligned} \text{NOT}_1 |00\rangle &= |10\rangle, & \text{NOT}_1 |01\rangle &= |11\rangle, & \text{NOT}_1 |10\rangle &= |00\rangle, & \text{NOT}_1 |11\rangle &= |01\rangle, \\ \text{NOT}_2 |00\rangle &= |01\rangle, & \text{NOT}_2 |01\rangle &= |00\rangle, & \text{NOT}_2 |10\rangle &= |11\rangle, & \text{NOT}_2 |11\rangle &= |10\rangle. \end{aligned}$$

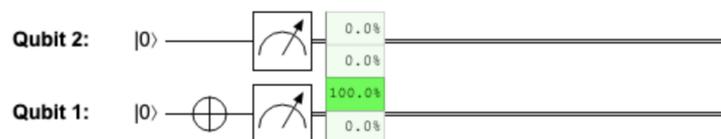
Der einzige Unterschied ist, dass wir die Bit-Notation  $[ab]$  durch die Qubit-Notation  $|ab\rangle$  ersetzt haben.

Schauen wir uns nun eine Anwendung an: Nehmen wir an, wir wollen aus  $|00\rangle$  alle möglichen Zwei-Qubit Basiszustände bilden. Dies kann durch NOT-Operationen getan werden:

$$|00\rangle = |00\rangle, \quad |01\rangle = \text{NOT}_2 |00\rangle, \quad |10\rangle = \text{NOT}_1 |00\rangle, \quad |11\rangle = \text{NOT}_2 \text{NOT}_1 |00\rangle.$$

Beachte, dass wir im letzten Fall die NOT-Operationen in umgekehrter Reihenfolge hätten durchführen können, weil beide Sequenzen beide Bits negieren.

Um eine lokale Operation in QUIRKY durchzuführen, platzieren wir die entsprechende Box auf entweder den ersten oder den zweiten Draht. Zum Beispiel führt die folgende Sequenz zum Zustand  $|10\rangle$  und zeigt die Messwahrscheinlichkeiten, wenn man beide Qubits misst:



Dies ergibt Sinn, da der untere Draht in QUIRKY dem ersten Qubit entspricht.

Dies kann man am Besten mit der Tensorprodukt-Notation aus Gl. (3.33) und Linearität darstellen. Wenn  $U$  eine beliebige Ein-Qubit Operation ist, dann können wir  $U_1$  definieren als die Zwei-Qubit Operation, die auf jedem Basisvektor  $|a, b\rangle = |a\rangle \otimes |b\rangle$  (wobei  $a, b \in \{0, 1\}$ ) wie folgt definiert ist:

$$U_1 |a, b\rangle = U|a\rangle \otimes |b\rangle. \quad (3.35)$$

Die rechte Seite bedeutet  $(U|a\rangle) \otimes |b\rangle$ , sprich das Tensorprodukt von den beiden Zuständen  $U|a\rangle$  und  $|b\rangle$ . Dies ist intuitiv, da wir einfach  $U$  nur auf das erste Qubit anwenden und das zweite Qubit unverändert lassen.

Um  $U_1$  auf einen beliebigen Zwei-Qubit Zustand anzuwenden, erweitern wir diese Vorschrift mittels *Linearität*. Wie in der ersten Woche bedeutet dies, dass wir erst  $|\psi\rangle$  erweitern wie in Gl. (3.30) und dann die Operation auf jeden Basisvektor anwenden. Dies ergibt also

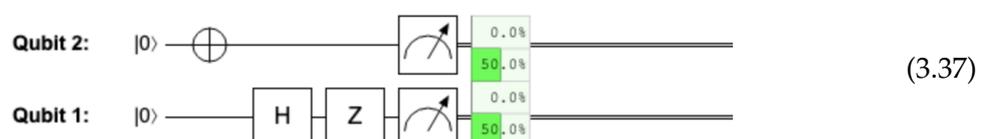
$$U_1 |\psi\rangle = \psi_{00} U_1 |00\rangle + \psi_{01} U_1 |01\rangle + \psi_{10} U_1 |10\rangle + \psi_{11} U_1 |11\rangle$$

Jetzt können wir Gl. (3.35) auf jeden der vier Terme anwenden. Wir definieren  $U_2$  analog:

$$U_2 |a, b\rangle = |a\rangle \otimes U|b\rangle \quad (3.36)$$

und erweitern es mittels Linearität. Dies ist analog zu den Formeln in Gl. (3.26) für gewöhnliche Bits.

Zusätzlich zur NOT-Operation sind zwei wichtige Quantum Operationen, die wir ständig benutzen, die Z-Operation aus Gl. (2.12) und die Hadamard-Operation aus Gl. (2.20). Da beide so wichtig sind, haben wir ihnen ihre eigenen Boxen in QUIRKY gegeben, nämlich  $\boxed{Z}$  und  $\boxed{H}$ .



Welchen Zustand erhalten wir hierdurch? Der mathematische Ausdruck für den Zustand ist

$$Z_1 H_1 \text{NOT}_2 |00\rangle. \quad (3.38)$$

Beachte, dass im Unterschied zur graphischen Beschreibung (3.37) sich der Eingabezustand  $|00\rangle$  in dieser Ausdrucksform auf der rechten Seite befindet. Dies führt dazu, dass die Reihenfolge der Operationen in umgekehrter Reihenfolge erscheinen. Jedoch beschreiben sowohl (3.37) als auch Gl. (3.38) den gleichen Prozess – die erste Operation, die wir anwenden auf  $|00\rangle$  ist  $\text{NOT}_2$ , dann  $H_1$  und schlussendlich  $Z_1$ . Der einzige Unterschied zwischen (3.37) und Gl. (3.38) ist die Vereinbarung, wie wir den Zeitverlauf darstellen: Er geht von links nach rechts in (3.37) und von rechts nach links in Gl. (3.38). Leider sind diese beiden abweichenden Konventionen Standard im Quantum Computing, wir können also nichts dagegen tun. Du musst also sorgfältig sein, wenn du QUIRKY Bilder in Gleichung übersetzt und umgekehrt!

### Übungsaufgabe 3.8: Ist QUIRKY korrekt?

Berechne den Zwei-Qubit Zustand in Gl. (3.38) (also den Zustand direkt vor dem Messen in (3.37)). Berechne die Wahrscheinlichkeit für die Messergebnisse und vergleiche deine Ergebnisse mit QUIRKY.

Letzte Woche haben wir in §2.4.3 besprochen, dass jegliche Operation auf einem einzelnen Qubit entweder eine Rotation  $U(\theta)$  oder eine Spiegelung  $V(\theta) = \text{NOT} U(\theta)$  ist. Da wir beliebige Rotationen in QUIRKY (s. §2.4.1) konstruieren können, können wir also beliebige lokale Operationen auf beiden Qubits ausführen mittels QUIRKY.

Es ist sogar so, dass die Regeln von Gl. (3.35) und (3.36) nicht nur auf Basisvektoren funktionieren, sondern sogar auf beliebigen Produktzuständen. Sprich, wenn  $|\alpha\rangle$  und  $|\beta\rangle$  beliebige einzelne Qubit-Zustände sind, dann gilt

$$U_1 (|\alpha\rangle \otimes |\beta\rangle) = U |\alpha\rangle \otimes |\beta\rangle, \quad (3.39)$$

$$U_2 (|\alpha\rangle \otimes |\beta\rangle) = |\alpha\rangle \otimes U |\beta\rangle. \quad (3.40)$$

### Übungsaufgabe 3.9: Lokale Operationen auf Produktzuständen (optional)

Kannst du Gl. (3.39) oder (3.40) nachweisen?

### 3.2.3 Parallele Operationen

Wenn wir eine Operation auf das erste Qubit anwenden und eine zweite Operation auf das zweite Qubit, dann ist die Reihenfolge der beiden Operationen egal. Das heißt, wenn  $U$  und  $V$  beliebige Operationen auf einem Qubit sind, dann gilt

$$U_1 V_2 = V_2 U_1. \quad (3.41)$$

Wir können diesen intuitiven Fakt beweisen, indem wir Gl. (3.39) und (3.40) benutzen. Für jeden Basiszustand  $|a, b\rangle$ , wobei  $a, b \in \{0, 1\}$ , gilt

$$U_1 V_2 |a, b\rangle = U_1 (|a\rangle \otimes V |b\rangle) = U |a\rangle \otimes V |b\rangle = V_2 (U |a\rangle \otimes |b\rangle) = V_2 U_1 |a, b\rangle,$$

Gl. (3.41) folgt durch Linearität.

Dadurch, dass die beiden Operationen auf unterschiedlichen Qubits ausgeführt werden können, können wir sie sogar parallel ausführen! Dies legt nahe, eine neue Notation für die Operation in Gl. (3.41) einzuführen:

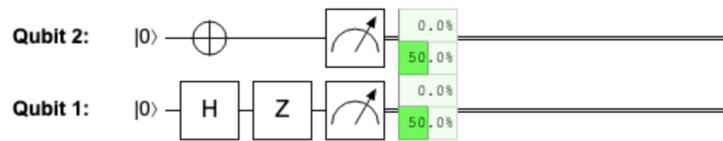
$$U \otimes V.$$

Wir verwenden erneut das Symbol für das Tensorprodukt, welches wir ursprünglich dafür verwendet hatten um zwei unabhängige Ein-Qubit Zustände in einen Zwei-Qubit Zustand zusammenzuführen. Die gleiche Bedeutung erweitert sich auch auf Quantenoperationen:  $U \otimes V$  bezeichnet die kombinierte Operation, die daraus besteht  $U$  und  $V$  auf zwei Untersystem von einem größeren System anzuwenden. Diese Notation ist besonders praktisch, da sie im Zusammenspiel mit dem ursprünglichen Tensorprodukt für Zustände ist:

$$(U \otimes V)(|\alpha\rangle \otimes |\beta\rangle) = U|\alpha\rangle \otimes V|\beta\rangle. \quad (3.42)$$

Diese Gleichung sagt lediglich Folgendes aus: Wenn man zwei unabhängige Zustände hat und dann eine Operation ausführt, welche unabhängig auf beiden Zuständen wirkt, dann führt man lediglich beide Operationen unabhängig auf den entsprechenden Zuständen aus. Wir nennen  $U \otimes V$  das **Tensorprodukt** von zwei Quantenoperationen oder eine **Paralleloperation**.

QUIRKY erlaubt es uns, lokale Quantenoperationen parallel auszuführen. Zum Beispiel hätten wir die Sequenz von Operationen in Gl. (3.37) schreiben können als



wo die NOT-Operation und die Hadamard-Operation nun parallel ausgeführt werden.

Diskutieren wir ein weiteres Beispiel. Was passiert, wenn wir  $H$  auf beide Qubits anwenden? Diese Operation bezeichnen wir als  $H \otimes H$  und nach Gl. (2.20) und (3.42) gilt:

$$\begin{aligned} (H \otimes H) |00\rangle &= (H|0\rangle) \otimes (H|0\rangle) \\ &= \left( \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \otimes \left( \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \\ &= \frac{1}{2} |0\rangle \otimes |0\rangle + \frac{1}{2} |0\rangle \otimes |1\rangle + \frac{1}{2} |1\rangle \otimes |0\rangle + \frac{1}{2} |1\rangle \otimes |1\rangle \\ &= \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle). \end{aligned}$$

Der resultierende Zustand wird **uniforme Superposition** auf den beiden Qubits genannt, da dieser Zustand jeden Zwei-Qubit Basiszustand mit gleicher Amplitude enthält. Wenn wir eine Messung auf diesem Zustand durchführen, erhalten wir alle vier Ergebnisse mit gleicher Wahrscheinlichkeit. Wir können dies bestätigen mittels QUIRKY:



### Übungsaufgabe 3.10: Einen Produktzustand konstruieren

1. Schreibe  $\frac{1}{2} (|00\rangle + |01\rangle - |10\rangle - |11\rangle)$  als Tensorprodukt von zwei Ein-Qubit Zuständen.
2. Wie kannst du diesen Zustand konstruieren, indem du eine Sequenz von lokalen Operationen auf  $|00\rangle$  anwendest?
3. Implementiere die Sequenz von Operation aus Schritt 2 in QUIRKY.

Gl. (3.42) zeigt, dass, wenn wir eine Paralleloperation auf einen Produktzustand anwenden, wir einen weiteren Produktzustand erhalten.

### Hausaufgabe 3.5: Produktzustände von Paralleloperationen

Zeige, dass jeder Produktzustand konstruiert werden kann, indem man parallele Quantenrotationen (sprich, eine Operation der Form  $U(\theta) \otimes U(\phi)$ ) auf den Zustand  $|00\rangle$  anwendet.

**Hinweis:** Durch Gl. (2.5) wissen wir, dass ein allgemeiner Ein-Qubit Zustand der Form  $|\psi(\theta)\rangle$  ist.

Wir beenden unsere Diskussion über lokale Operationen mit einigen allgemeinen Anmerkungen. Zuerst ist es leicht zu überprüfen, dass

$$(U \otimes V)(U' \otimes V') = UU' \otimes VV', \quad (3.43)$$

für alle vier Ein-Qubit Operationen  $U, U', V, V'$ . Kannst du ein Bild malen, was dies visualisiert?

Wir können nun Nutzen machen von der Einheitsmatrix  $I$  von Gl. (2.18), welche  $I|\psi\rangle = |\psi\rangle$  erfüllt (dies lässt sich trivialerweise erweitern als  $I_1$  und  $I_2$  auf zwei Qubits). Diese Matrix ist nutzlos, wenn man sie alleine benutzt, da sie nichts verändert. Sie kann aber von Nutzen sein, wenn man die Tensorproduktnotation verwendet. Zum Beispiel erlaubt uns die Operation Folgendes:

$$U_1 = U \otimes I \quad V_2 = I \otimes V,$$

welche klarmacht, dass z.B.  $U_1$  die  $U$ -Operation auf dem ersten Qubit ausführt und nichts auf dem zweiten Qubit. Zum Beispiel wird die Identität  $U_1V_2 = V_2U_1$  von Gl. (3.41) nun

$$(U \otimes I)(I \otimes V) = U \otimes V = (I \otimes V)(U \otimes I) \quad (3.44)$$

was ziemlich intuitiv ist.

Du könntest dich fragen, ob die Reihenfolge der Operationen jemals eine Rolle spielt, wenn wir sie auf dem *gleichen* Qubit anwenden. Betrachten wir zwei Rotationen, dann folgt aus Gl. (2.15), dass ihre Reihenfolge nicht wichtig ist, da

$$U(\theta)U(\theta') = U(\theta + \theta') = U(\theta' + \theta) = U(\theta')U(\theta).$$

Wenn  $U$  und  $V$  jedoch zwei beliebige Ein-Qubit Operationen (insbesondere, wenn eine davon eine Reflektion ist), dann ist ihre Zusammensetzung generell abhängig von der Reihenfolge (s. Übungsaufgabe 3.11 unten). Das heißt, dass

$$UV \neq VU.$$

Dieses Problem besteht natürlich weiterhin, wenn man ein weiteres Qubit hat, und weiterhin beide Operationen auf dem gleichen Qubit anwendet. Zum Beispiel

$$(U \otimes I)(V \otimes I) = UV \otimes I \neq VU \otimes I = (V \otimes I)(U \otimes I). \quad (3.45)$$

Wir können dies auch schreiben als  $U_1V_1 \neq V_1U_1$  (und gleichermaßen wenn wir beide Operationen auf dem zweiten Qubit anwenden).

Um intuitiv den Unterschied zwischen Gl. (3.44) und (3.45) zu verstehen, stell dir vor, dass  $U$  anzuwenden „eine Socke anzuziehen“ bedeutet und  $V$  „einen Schuh anzuziehen“ bedeutet. Es ist offensichtlich, dass es einen Unterschied macht, wenn man  $U$  und  $V$  auf den gleichen Fuß anwendet und die Reihenfolge ändert. Jedoch, wenn man  $U$  und  $V$  auf verschiedene Füße anwendet (sprich,  $U \otimes I$  und  $I \otimes V$  anwendet), dann ist das Resultat immer gleich, unabhängig von der Reihenfolge der Operationen. Wenn man richtig angezogen sein möchte, sollte man jedoch  $(U \otimes U)$  und dann  $(V \otimes V)$  anwenden.

### Übungsaufgabe 3.11: Die Reihenfolge ist wichtig

Zeige, dass  $HZ \neq ZH$ .

#### 3.2.4 Kontrollierte Operationen

Um über Produktzustände hinauszugehen, brauchen wir eine Operation, die es uns erlaubt, dass zwei Quantenbits interagieren. Wie zuvor (s. Gl. (2.18) für probabilistische Bits) werden wir eine *kontrollierte NOT-Operation* hierfür werden, welche wir in Analogie zu Gl. (3.18) und (3.19) definieren:

$$\begin{aligned}\text{CNOT}_{1 \rightarrow 2} |00\rangle &= |00\rangle, \\ \text{CNOT}_{1 \rightarrow 2} |01\rangle &= |01\rangle, \\ \text{CNOT}_{1 \rightarrow 2} |10\rangle &= |11\rangle, \\ \text{CNOT}_{1 \rightarrow 2} |11\rangle &= |10\rangle,\end{aligned}\tag{3.46}$$

oder knapper:

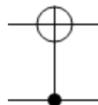
$$\text{CNOT}_{1 \rightarrow 2} |a, b\rangle = |a, a \oplus b\rangle.\tag{3.47}$$

D.h., wenn die Operation  $\text{CNOT}_{1 \rightarrow 2}$  auf Basiszustände angewendet wird, schaltet diese das zweite Qubit um (sprich ändert den Wert von 0 auf 1 und umgekehrt) je nach dem Wert des ersten Qubits. Wir können auch die Operation  $\text{CNOT}_{2 \rightarrow 1}$  definieren, welche das zweite Qubit als Kontrollwert nimmt und das erste Qubit als Ziel.

$$\text{CNOT}_{2 \rightarrow 1} |a, b\rangle = |a \oplus b, b\rangle.\tag{3.48}$$

Wie immer können wir diese Formeln mittels Linearität auf beliebige Zwei-Qubit Zustände erweitern.

In QUIRKY kannst du eine kontrollierte NOT-Operation für Quantenbits in der gleichen Art und Weise bilden, wie du es bereits für gewöhnliche Bits gelernt hast – siehe §3.1.6, falls du dich nicht erinnerst.



Viele der Dinge, die wir für probabilistische Bits bewiesen haben sind immer noch richtig für Quantenbits. Zum Beispiel erlaubt dir deine Lösung für Hausaufgabe 3.2 auch zwei Qubits zu tauschen! Ein weiteres Beispiel hierfür ist die Tatsache, dass das Ausführen der gleichen kontrollierten NOT-Operation zweimal hintereinander nichts bewirkt. Zum Beispiel ist dies der Fall für  $\text{CNOT}_{1 \rightarrow 2}$ , weil

$$\text{CNOT}_{1 \rightarrow 2} \text{CNOT}_{1 \rightarrow 2} |a, b\rangle = \text{CNOT}_{1 \rightarrow 2} |a, a \oplus b\rangle = |a, a \oplus a \oplus b\rangle = |a, b\rangle$$

da  $a \oplus a = 0$  für jedes  $a \in \{0, 1\}$ . Als Konsequenz daraus, ist die kontrollierte NOT-Operation die Inverse zu sich selbst:

$$\text{CNOT}_{1 \rightarrow 2}^{-1} = \text{CNOT}_{1 \rightarrow 2}\tag{3.49}$$

wobei  $M^{-1}$  die inverse Operation zu  $M$  bezeichnet (s. §2.4.2).

Wenn du ein bisschen mit QUIRKY herumspielst, wirst du vielleicht bereits erkannt haben, dass du  mit beliebigen Ein-Qubit Operationen kombinieren kannst, nicht nur mit der NOT-Operation. Tatsächlich können wir die **kontrollierte U-Operation** für jede Ein-Qubit Operation  $U$  definieren. Wir bezeichnen diese als  $\text{CU}_{1 \rightarrow 2}$  und  $\text{CU}_{2 \rightarrow 1}$ , je nachdem welches Qubit

das Kontrollbit und welches das Zielbit ist. Zum Beispiel ist  $CU_{1 \rightarrow 2}$  wie folgt auf den vier Basiszuständen definiert:

$$\begin{aligned} CU_{1 \rightarrow 2} |00\rangle &= |0\rangle \otimes |0\rangle, \\ CU_{1 \rightarrow 2} |01\rangle &= |0\rangle \otimes |1\rangle, \\ CU_{1 \rightarrow 2} |10\rangle &= |1\rangle \otimes U |0\rangle, \\ CU_{1 \rightarrow 2} |11\rangle &= |1\rangle \otimes U |1\rangle. \end{aligned}$$

Du kannst leicht überprüfen, dass wir für  $U = \text{NOT}$  wieder  $\text{CNOT}_{1 \rightarrow 2}$  erhalten.

### 3.2.5 Verschränkte Zustände

In Gl. (3.33) haben wir das Tensorprodukt benutzt, um einen Zwei-Qubit Zustand aus zwei Ein-Qubit Zuständen zu basteln. In §3.2.3 haben wir gesehen, dass diese *Produktzustände* genau die gleichen Zustände sind, die wir erhalten, wenn wir lokale Quantenoperationen auf  $|00\rangle = |0\rangle \otimes |0\rangle$  anwenden (was wiederum ein Produktzustand ist). Nun, es existieren auch Zwei-Qubit Zustände, die *nicht* Produktzustände sind. Wir nennen diese Zustände **verschränkt** und wir werden sehen, dass sie von Bedeutung für Quantum Computing sind.

Wie können wir entscheiden, ob ein Zustand ein Produktzustand ist oder nicht? Obwohl Quantenzustände durch Amplituden und nicht durch Wahrscheinlichkeiten spezifiziert sind, können wir die gleiche Methode benutzen wie in §3.1.8 um zu entscheiden ob eine Wahrscheinlichkeitsverteilung korreliert ist oder nicht. Haben wir einen Zwei-Qubit Zustand der Form (3.30), benutzen wir zuerst Gl. (3.28) um Folgendes zu berechnen:

$$\Delta(|\psi\rangle) = \psi_{00}\psi_{11} - \psi_{01}\psi_{10}. \quad (3.50)$$

Dann ist  $|\psi\rangle$  ein Produktzustand genau dann wenn  $\Delta(|\psi\rangle) = 0$ . Auf diese Weise sind verschränkte Zustände analog zu korrelierten Wahrscheinlichkeitsverteilungen.

Ein simples aber wichtiges Beispiel eines verschränkten Zwei-Qubit Zustandes ist

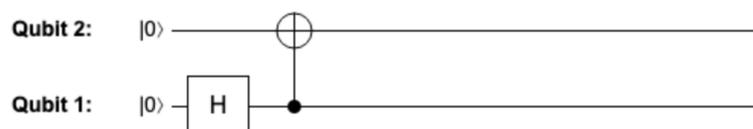
$$|\Phi^+\rangle = \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle. \quad (3.51)$$

Dieser Zustand ist analog zu einem Paar perfekt korrelierter zufälliger Bits aus Gl. (3.27). Er ist verschränkt, da

$$\Delta(|\Phi^+\rangle) = \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} - 0 \cdot 0 = \frac{1}{2} \neq 0.$$

Wir nennen  $|\Phi^+\rangle$  den **maximal verschränkten Zustand** von zwei Qubits (obwohl weder der Grund für den Namen noch die merkwürdige Notation zu diesem Zeitpunkt klar sind).

Wie können wir verschränkte Zustände bilden? Genau wie wir auch korrelierte Zwei-Bit-Zustände gebildet haben, können wir die kontrollierte NOT-Operation benutzen, um zwei Qubits interagieren zu lassen. Zum Beispiel bereitet die folgende Sequenz von Operationen den maximal verschränkten Zustand  $|\Phi^+\rangle$  vor:



Verifizieren wir dies kurz:

$$\text{CNOT}_{1 \rightarrow 2}(H \otimes I)|00\rangle = \text{CNOT}_{1 \rightarrow 2}\left(\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle\right) = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle.$$

Beachte, dass die Anwendung von CNOT auf  $|00\rangle$  (oder jeglichen anderen Basiszustand) nicht funktioniert hätte (s. Gl. (3.46)).

### Hausaufgabe 3.6: Ein weiterer verschränkter Zustand

1. Verifiziere, dass der Zustand  $|\psi\rangle = \frac{1}{2}|01\rangle + \frac{\sqrt{3}}{2}|10\rangle$  verschränkt ist.  
**Hinweis:** Berechne Gl. (3.50).
2. Finde eine Sequenz von Operationen in QUIRKY, die den Zustand  $|\psi\rangle$  vorbereitet.  
**Hinweis:** Du musst möglicherweise eine geeignete Rotation konstruieren.
3. Was sind die Wahrscheinlichkeiten für Messergebnisse, wenn du beide Qubits von  $|\psi\rangle$  misst? Benutze QUIRKY um dein Ergebnis zu überprüfen.

Der maximal verschränkte Zustand in Gl. (3.51) ist aus einer Familie von vier Zuständen, die **Bell Zustände** genannt werden. Die Bell Zustände sind wie folgt definiert:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle, \quad (3.52)$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle, \quad (3.53)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle, \quad (3.54)$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle. \quad (3.55)$$

Die Zustände sind nach John Stewart Bell benannt, welcher einer der ersten Menschen war, die die außergewöhnlichen Eigenschaften von Quantenverschränkung (engl. *quantum entanglement*) erkannt hat. Wie können wir diese vier Bell Zustände erzeugen? Oben haben wir gesehen, dass wir  $|\Phi^+\rangle$  erzeugen können, indem wir die Hadamard- und CNOT-Operation auf den Basiszustand  $|00\rangle$  anwenden. Es ist eine einfache Aufgabe zu überprüfen, dass die anderen drei Bell Zustände ähnlich konstruiert werden können, also durch Anwenden der gleichen Operationen auf die anderen drei Basiszustände. Anders gesagt, wenn wir die folgende Operationen definieren:

$$U_{\text{Bell}} = \text{CNOT}_{1 \rightarrow 2}(H \otimes I) \quad (3.56)$$

dann

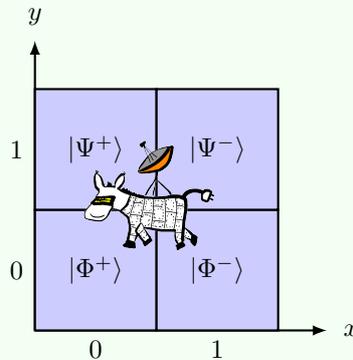
$$\begin{aligned} |\Phi^+\rangle &= U_{\text{Bell}}|00\rangle, & |\Phi^-\rangle &= U_{\text{Bell}}|10\rangle, \\ |\Psi^+\rangle &= U_{\text{Bell}}|01\rangle, & |\Psi^-\rangle &= U_{\text{Bell}}|11\rangle. \end{aligned}$$

### Übungsaufgabe 3.12: Bell Zustände vorbereiten

Zeichne, wie du die anderen drei Bell Zustände in QUIRKY konstruieren würdest:  $|\Phi^-\rangle$ ,  $|\Psi^+\rangle$ , and  $|\Psi^-\rangle$ .

### Übungsaufgabe 3.13: Bell Zustände unterscheiden

Alices Roboteresel ist während einer Entdeckungsmission abhandengekommen! Er will Alice schnell seine Position wissen lassen, damit sie ihn retten kann. Der Esel ist in einer der vier Gegenden um die Schule. Um mitzuteilen in welcher, sendet der Esel eine Zwei-Qubit Quantennachricht  $|x, y\rangle$ , wobei  $x \in \{0, 1\}$  die  $x$  Koordinate und  $y \in \{0, 1\}$  die  $y$  Koordinate der Lage beschreiben:



Leider hat Alices böse Klassenkameradin Eve das Signal blockiert, d.h. was Alice stattdessen erhält ist einer der vier Bell Zustände wie oben gezeigt. Hilf Alice dabei, korrekt das Signal zu dekodieren und den Esel zu orten! D.h., finde eine Sequenz von Operationen, die jeden der vier Bell Zustände auf den entsprechenden Basiszustand  $|x, y\rangle$  zurückführt.



### 3.2.6 Verschränkung und Korrelationen

Durch die Ähnlichkeit zwischen verschränkten Zuständen und korrelierten Wahrscheinlichkeitsverteilungen magst du dich bereits gefragt haben, inwiefern die beiden Notationen verwandt sind. Um sie zu vergleichen, lass uns die generelle Relation zwischen Quantenzuständen und Wahrscheinlichkeitsverteilungen diskutieren.

Um anzufangen, nehmen wir an, dass wir den Ein-Qubit Zustand  $|\psi\rangle = \psi_0 |0\rangle + \psi_1 |1\rangle$  haben und ihn messen. Dann wissen wir von §2.2, dass das Ausgabebit entweder 0 oder 1 ist mit Wahrscheinlichkeiten  $\psi_0^2$  und  $\psi_1^2$ . Wir können dies durch eine Wahrscheinlichkeitsverteilung modellieren:

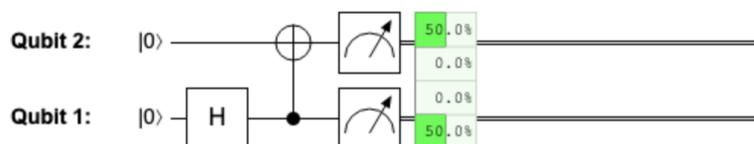
$$\psi_0^2[0] + \psi_1^2[1].$$

Intuitiv modelliert dies die Situation, wo wir das Qubit messen aber nicht das Ergebnis angeschaut haben (hätten wir dies getan, hätten wir nicht ein probabilistisches Bit, aber ein deterministisches, welches entweder 0 oder 1 ist).

Die gleiche Logik funktioniert auch für zwei Qubits. Wenn wir eine Zwei-Qubit Zustand  $|\psi\rangle = \psi_{00} |00\rangle + \psi_{01} |01\rangle + \psi_{10} |10\rangle + \psi_{11} |11\rangle$  messen, können wir das Ergebnis beschreiben durch die folgende Zufallsverteilung:

$$p = \psi_{00}^2[00] + \psi_{01}^2[01] + \psi_{10}^2[10] + \psi_{11}^2[11]. \quad (3.57)$$

Wenn wir beispielsweise den maximal verschränkten Zustand  $|\Phi^+\rangle$  vorbereiten und messen, dann erhalten wir ein perfekt korreliertes Paar von Zufallsbits in Gl. (3.27). Wir können dies mittels QUIRKY bestätigen:



Das Gleiche gilt natürlich, wenn wir den Bell Zustand  $|\Phi^-\rangle$  stattdessen benutzen. (Was ist mit den anderen beiden Bell Zuständen  $|\Psi^+\rangle$  oder  $|\Psi^-\rangle$ ? Wenn man diese misst, erhält man ein Paar von perfekt *anti-korrelierten* Bits, welches man durch die Wahrscheinlichkeitsverteilung  $\frac{1}{2}[01] + \frac{1}{2}[10]$  beschreiben kann.)

Das vorherige Beispiel war kein Zufall. Es ist tatsächlich so, dass die Wahrscheinlichkeitsverteilung  $p$  in Gl. (3.57), die man durch Messen eines Zwei-Qubit Zustands erhält, genau dann korreliert ist, wenn der korrespondierende Quantenzustand  $|\psi\rangle$  verschränkt ist. Um dies zu sehen, nehmen wir an dass  $|\psi\rangle$  ein Produktzustand ist, sodass  $\Delta(|\psi\rangle) = 0$ . Dann ist  $p$  eine Produktverteilung, da

$$\begin{aligned}\Delta(p) &= p_{00}p_{11} - p_{01}p_{10} = \psi_{00}^2\psi_{11}^2 - \psi_{01}^2\psi_{10}^2 \\ &= (\psi_{00}\psi_{11} - \psi_{01}\psi_{10})(\psi_{00}\psi_{11} + \psi_{01}\psi_{10}) \\ &= \Delta(|\psi\rangle)(\psi_{00}\psi_{11} + \psi_{01}\psi_{10}) = 0.\end{aligned}\tag{3.58}$$

Dies belegt die Behauptung, dass korrelierte Messergebnisse Verschränkung im gemessenen Zustand implizieren.

Beachte, dass Quantenzustände generell mindestens so nützlich sind wie probabilistische Bits, da jede Wahrscheinlichkeitsverteilung durch Messen eines geeigneten Quantenzustands erhalten werden kann. D.h., gegeben eine beliebige Wahrscheinlichkeitsverteilung  $p$  können wir immer einen Quantenzustand  $|\psi\rangle$  finden, wessen Messergebnis gemäß  $p$  verteilt ist. Zum Beispiel können wir für eine Zwei-Bit Verteilung  $p$  einfach

$$|\psi\rangle = \sqrt{p_{00}}|00\rangle + \sqrt{p_{01}}|01\rangle + \sqrt{p_{10}}|10\rangle + \sqrt{p_{11}}|11\rangle.$$

wählen.

Insbesondere bedeutet dies, dass Verschränkung generell mindestens so nützlich ist wie probabilistische Korrelationen, da jede korrelierte Verteilung über zwei probabilistische Bits erzeugt werden kann, indem wir einen bestimmten Zwei-Qubit Zustand messen.

### 3.2.7 Die Macht von Verschränkung

In Wirklichkeit sind Quantenbits tatsächlich mächtiger als probabilistische Bits! Im Folgenden werden wir nur einen flüchtigen Blick darauf werfen, in den kommenden Wochen werden wir viele weitere Beispiele betrachten.

Lass uns die Macht von Verschränkung illustrieren, indem wir eine kurze Geschichte erzählen, die auf Übungsaufgabe 3.13 folgt. Dort hast du Alice geholfen, die Position von ihrem Esel zu dekodieren, welcher sich in einem von vier Orten befand, welche durch zwei Bits  $a, b \in \{0, 1\}$  gekennzeichnet waren. Alice hat keine Zeit den Esel aufzusammeln, also möchte sie die Position an Bob weiterleiten. Leider hat Alice kaum noch Geld auf ihrem Quantenhandy, also kann sie nur ein einziges Qubit an Bob senden. Aber das Senden eines einzelnen Qubits wird sicherlich *nicht* genug sein um zwei Bits zu kommunizieren. Wir wissen dies, da die einzige Möglichkeit für Bob Informationen zu extrahieren darin besteht, das Qubit zu messen – Aber durch das Messen lernt er nur ein einzelnes Bit und der ursprüngliche Zustand vom Qubit ist verschwunden.

Glücklicherweise teilen sich Alice und Bob schon vorher einen maximal verschränkten Zustand  $|\Phi^+\rangle$ . Das bedeutet für uns, dass Alice das erste Qubit hat und Bob das zweite Qubit. Können wir Alice dabei helfen, den Ort des Esel (sprich, die beiden Bits  $a$  und  $b$ ) zu senden, wobei wir nur ein *einziges* Qubit senden? Die Antwort ist ja! Der Schlüssel dazu ist der Folgende:



### Übungsaufgabe 3.14: Einen Bell Zustand in einen Anderen überführen

Zeige, dass Alice den maximal verschränkten Zustand  $|\Phi^+\rangle$  in jeden anderen Bell Zustand  $|\Phi^-\rangle$ ,  $|\Psi^+\rangle$ , oder  $|\Psi^-\rangle$  überführen kann mittels lokaler Operationen nur auf ihrem Qubit.

Es ist jetzt klar wie Alice und Bob das Hindernis lösen können. Nehmen wir an, dass Alice und Bob vorher vereinbart haben, wie sie die vier möglichen Werte der zwei Bits  $[ab]$  auf die vier Bell Zustände abbilden (nehmen wir an, dass  $[00]$  zu  $|\Phi^+\rangle$  korrespondiert,  $[01]$  zu  $|\Psi^+\rangle$ , etc.). Mittels Übungsaufgabe 3.14 wendet Alice zunächst eine Operation an auf ihr Qubit des maximal verschränkten Zustands um den Zustand in den Bell Zustand ihrer Wahl zu überführen (sprich, in den Zustand der zum Ort des Esels korrespondiert). Dann sendet sie ihr Qubit an Bob. Bob hat nun beide Qubits in seinem Besitz und er weiß, dass sie in einem von vier Bell Zuständen sind. D.h., er kann einfach die gleichen Operationen wie in Übungsaufgabe 3.13 anwenden und dann die zwei Qubits messen, um den Ort herauszufinden.

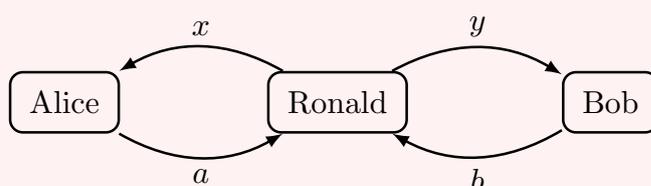
Das obige Verfahren ist bekannt als das **Superdense-Coding-Protokoll** (engl. für "Superdichte-Kodierung"), da wir zwei deterministische Bits übertragen, indem wir ein Qubit senden (mit den Extrakosten, dass wir einen maximal verschränkten Zustand, den Alice und Bob sich teilen, „aufbrauchen“). Hätten Alice und Bob dafür auch probabilistische Bits verwenden können anstelle des verschränkten Zustands, um die Herausforderung zu lösen (z.B. ein Paar von perfekt korrelierten Bits)? Interessanterweise lautet die Antwort nein. Wir werden dies nicht im Detail beweisen, aber man kann zeigen, dass wenn wir die probabilistischen Bits verwenden hätten können, um das Problem perfekt zu lösen, dann hätte auch eine deterministische Strategie dies tun können. Diese Strategie hätte ein einzelnes Bit gesendet, was natürlich nicht zwei Bits an Information beinhalten kann. Also demonstriert superdense coding, dass Quantenverschränkung einen Vorteil für Kommunikation verschafft.

In der folgenden Aufgabe begegnest du einer weiteren berühmten Situation, wo Quantenverschränkung einen eindeutigen Vorteil verschafft.

### Hausaufgabe 3.7: Ein verschränktes Spiel (anspruchsvoll)

Alice und Bob langweilen sich im Unterricht, also fragen sie ihren Lehrer für Quantenmechanik Ronald nach einem anspruchsvollen Puzzle. Nach nur einer kurzen Pause erklärt Ronald ihnen ein interessantes Spiel. Das Ziel des Spiels ist es, dass Alice und Bob so gut wie möglich kooperieren (sie spielen *nicht* gegeneinander). Während des Spiels dürfen sie jedoch *nicht* miteinander kommunizieren. Die Regeln des Spiels sind wie folgt:

- Um anzufangen, wirft Ronald heimlich zwei faire Münzen. Er nennt Alice das Ergebnis des ersten Wurfs (Bit  $x$ ) und Bob das Ergebnis des zweiten Wurfs (Bit  $y$ ). Wir nennen diese Bits die Eingabebits.
- Nachdem die Beiden die Bits erhalten haben, müssen sowohl Alice als auch Bob sich selbst ein Bit überlegen und dies angeben (Bits  $a$  und  $b$ ).
- Alice und Bob gewinnen das Spiel unter folgenden Bedingungen: Wenn  $x = y = 1$ , dann gewinnen sie wenn  $a \neq b$ ; ansonsten gewinnen sie, wenn  $a = b$ .



$x$	$y$	Gewinnbedingung
0	0	$a = b$
0	1	$a = b$
1	0	$a = b$
1	1	$a \neq b$

Bevor das Spiel anfängt, diskutieren Alice und Bob kurz ihre Strategie. Zuerst erwägen sie es, zwei Funktionen  $f, g : \{0, 1\} \rightarrow \{0, 1\}$  auf ihre Bits  $x$  und  $y$  anzuwenden und ihre Antworten wie folgt zu berechnen:  $a = f(x)$  und  $b = g(y)$ .

1. Zeige, dass Alice und Bob in diesem Fall das Spiel mit Wahrscheinlichkeit 75% gewinnen können, aber nicht höher.

Als Nächstes erwägen sie es, ihre Antworten zu berechnen mittels geteiltem Zufall. Bob schlägt vor, kompliziertere Funktionen  $f$  und  $g$  mit einem zusätzlichen binären Eingabebit zu verwenden und die Antworten wie folgt zu berechnen:  $a = f(x, r)$  und  $b = g(y, s)$ . Hier sind  $r$  und  $s$  zwei zufällige Bits die gemeinsam einer Zwei-Bit Zufallsverteilung entspringen.

2. Zeige, dass die Beiden noch immer nicht mit einer Wahrscheinlichkeit von mehr als 75% gewinnen können, egal welche Funktionen  $f$  und  $g$  sie benutzen und was die Zufallsverteilung für  $r$  und  $s$  ist.

Langsam wird den Beiden bewusst, dass Ronald sicherlich eine quantenmechanische Strategie im Kopf hatte. Alice hat eine geniale Idee und schlägt vor, dass sie und Bob sich einen maximal verschränkten Zustand  $|\Phi^+\rangle$  bevor das Spiel startet teilen. Sie schlägt vor, dass wenn sie ihre Bits erhalten, sie ihr Qubit rotiert mittels einem Winkel  $\theta_x$  (welcher von ihrem Eingabebit  $x$  abhängt) und dann misst um ihre Antwort  $a$  zu erhalten. Bob stattdessen soll anhand eines anderen Winkels  $\omega_y$  rotieren (welcher von seinem Eingabebit  $y$  abhängt) und dann messen um seine Antwort  $b$  zu erhalten.

3. Schreibe den Zustand nach den Rotationen von Alice und Bob auf. Der Zustand soll von der Form (3.30) sein. Bestätige, dass die Gewinnwahrscheinlichkeit von der folgenden Form ist:

$$\frac{1}{4} (\cos^2(\theta_0 - \omega_0) + \cos^2(\theta_0 - \omega_1) + \cos^2(\theta_1 - \omega_0) + \sin^2(\theta_1 - \omega_1)).$$

**Hinweis:** Benutze die trigonometrischen Formeln aus Gl. (2.16) und (2.22).

Alice und Bob finden schnell heraus, dass  $\theta_0 = 0$ ,  $\theta_1 = \pi/4$ , und  $\omega_0 = \pi/8$  gute Wahlen sind. Sie haben jedoch Probleme mit dem letzten Winkel und die Zeit läuft ab.

4. Finde eine Winkel  $\omega_1$ , sodass die Beiden mit einer Wahrscheinlichkeit gewinnen, die höher als 75% ist.

In den ersten beiden Teilen der Hausaufgabe hast du gezeigt, dass jede Strategie mit klassischen Bits nicht mit einer Wahrscheinlichkeit von mehr als 75% gewinnen kann. Dies ist ein Beispiel der *Bell-Ungleichheit*. Die obige Version geht zurück auf John Clauser, Michael Horne, Abner Shimony, and Richard Holt, d.h. das Spiel, welches Alice und Bob mit Ronald spielen wird oft *CHSH-Spiel* genannt. Die quantenmechanische Strategie, die du in der Hausaufgabe entdeckt hast, *verletzt* diese Bell Ungleichheit. Es ist ein empirischer Fakt, dass Bell Ungleichheiten mittels Quantenmechanik verletzt werden. Dies wurde zuerst durch Alain Aspect in den 80ern entdeckt und kürzlich unter sehr strengen Konditionen in einem Experiment von Ronald Hanson und seinem Team in der TU Delft bewiesen.

Interessanterweise können Alice und Bob nicht nur das CHSH Spiel mit höherer Wahrscheinlichkeit gewinnen, wenn sie einen maximal verschränkten Zustand haben, sie können Ronald sogar davon überzeugen, dass sie Quantentricks verwenden müssen um das Spiel so gut spielen zu können! Da die probabilistischen Strategien nicht mit einer Wahrscheinlichkeit von mehr als 75% gewinnen können, ist es so, dass wenn die Beiden mit einer höheren Wahr-

scheinlichkeit gewinnen, dann ist die einzig mögliche Erklärung, dass sie etwas Mächtigeres benutzen. Mit weiteren Tricks können sie Ronald sogar davon überzeugen, dass sie den maximal verschränkten Zustand benutzen müssen und die Rotationen mit den bestimmten Winkeln von oben. Dies bedeutet, dass sie Ronald unwiderleglich beweisen können, dass sie kleine Quantencomputer besitzen, welche einzelne Qubits manipulieren können, und verschränkte Qubits teilen. Dies ist eine wichtige Beobachtung, da sie einen das CHSH Spiel benutzen lässt, um zu beweisen, dass man tatsächlich einen Quantencomputer gebaut hat! Sollten tatsächlich zwei deiner Klassenkameraden behaupten, dass sie jeweils einen Quantencomputer in ihrer Garage gebaut haben, dann kannst du sie einfach dazu auffordern, das CHSH Spiel gegen dich zu spielen. Wenn sie mit einer Wahrscheinlichkeit, die signifikant größer als 75% ist, gewinnen, dann weißt du, dass sie tatsächlich die Wahrheit sagen und echte Quantencomputer besitzen. Wenn sie jedoch nicht mit einer Wahrscheinlichkeit von mehr als 75% gewinnen, dann kannst du ihre Behauptungen entkräften. Ist das nicht verblüffend?

Diese Arten von Verifikations-Prozeduren sind etwas, woran Forscher:innen auf der ganzen Welt momentan aktiv arbeiten. Dies ist ein sehr wichtiges Problem, da einige große Firmen, z.B. IBM, Google und Microsoft versuchen, einen Quantencomputer zu bauen. Wenn sie behaupten, einen gebaut zu haben, würdest du ihnen wahrscheinlich mehr glauben als deine Klassenkameraden. Es ist jedoch trotzdem toll, eine Möglichkeit zu haben, dies zu verifizieren.

### 3.3 Lösungen der Übungsaufgaben

#### Lösung Übungsaufgabe 3.2

1. Wir bezeichnen mit  $p$  die Wahrscheinlichkeitsverteilung von Alices zwei Münzwürfen. Wir benutzen Abb. 3.3 um die Wahrscheinlichkeiten zu bestimmen. Wir wissen, dass der erste Münzwurf von Alice verteilt ist nach  $u$ , was bedeutet, dass wenn wir das erste Bit messen, wir jedes Bit mit Wahrscheinlichkeit je  $1/2$  erhalten sollten:

$$p_{00} + p_{01} = \frac{1}{2}, \quad p_{10} + p_{11} = \frac{1}{2}.$$

Wenn der erste Münzwurf eine 0 ergab, dass sollte der Zustand des zweiten Bits beschrieben werden durch die Münze  $q$ . D.h., wir müssen haben:

$$\frac{p_{00}[0] + p_{01}[1]}{p_{00} + p_{01}} = \frac{3}{4}[0] + \frac{1}{4}[1],$$

Davon können wir ableiten, dass  $p_{00} = \frac{3}{8}$  and  $p_{01} = \frac{1}{8}$ . Analog dazu gilt: Wenn der erste Münzwurf eine 1 war, dann wird der Zustand des zweiten Bits beschrieben durch die Münze  $r$ , also

$$\frac{p_{10}[0] + p_{11}[1]}{p_{10} + p_{11}} = \frac{1}{3}[0] + \frac{2}{3}[1],$$

ergo  $p_{10} = \frac{1}{6}$  and  $p_{11} = \frac{2}{6}$ . Zusammen gilt:

$$p = \frac{3}{8}[00] + \frac{1}{8}[01] + \frac{1}{6}[10] + \frac{2}{6}[11],$$

2. Der Wert von Bob's Bit ist der Gleiche wie das Ergebnis von Alices zweitem Münzwurf, also folgen wir einfach Abb. 3.3 um die Wahrscheinlichkeit der Ergebnisse zu bestimmen, wenn wir das zweite Bit messen. Die Wahrscheinlichkeit, dass Bobs Ausgabe 0 ist, ist also

$$p_{00} + p_{10} = \frac{3}{8} + \frac{1}{6} = \frac{13}{24}$$

und die Wahrscheinlichkeit, dass seine Ausgabe 1 ist, ist  $1 - \frac{13}{24} = \frac{11}{24}$ . Wir können dies durch die folgende Zufallsverteilung beschreiben:

$$\frac{13}{24}[0] + \frac{11}{24}[1].$$

3. Wir benutzen wieder Abb. 3.3. Wenn das zweite Bit gemessen wird und das Ergebnis 0 ist, dann ist der Zustand des ersten Bits gegeben durch

$$\frac{p_{00}[0] + p_{10}[1]}{p_{00} + p_{10}} = \frac{9}{13}[0] + \frac{4}{13}[1].$$

Es ist also wahrscheinlicher, dass Alices erster Münzwurf das Ergebnis 0 zeigt.

Analog dazu, wenn das Ergebnis vom Messen des zweiten Bits 1 ist, dann wird der Zustand des erstens Bits beschrieben durch

$$\frac{p_{01}[0] + p_{11}[1]}{p_{01} + p_{11}} = \frac{3}{11}[0] + \frac{8}{11}[1].$$

In diesem Fall ist es wahrscheinlicher, dass Alices erster Münzwurf 1 ist.

### Lösung Übungsaufgabe 3.1

$$\text{NOT}_1 \begin{pmatrix} p_{00} \\ p_{01} \\ p_{10} \\ p_{11} \end{pmatrix} = \begin{pmatrix} p_{10} \\ p_{11} \\ p_{00} \\ p_{01} \end{pmatrix}.$$

### Lösung Übungsaufgabe 3.3

$$\text{SWAP} \begin{pmatrix} p_{00} \\ p_{01} \\ p_{10} \\ p_{11} \end{pmatrix} = \begin{pmatrix} p_{00} \\ p_{10} \\ p_{01} \\ p_{11} \end{pmatrix}.$$

### Lösung Übungsaufgabe 3.4

1.  $\text{CNOT}_{2 \rightarrow 1}[a, b] = [a \oplus b, b] = [b \oplus a, b]$ .
2. Dies kann getan werden, indem man zuerst eine SWAP-Operation ausführt, dann  $\text{CNOT}_{1 \rightarrow 2}$ , und am Ende wieder SWAP. Tatsächlich, wenn wir Gl. (3.17) und (3.19) nutzen, gilt

$$\begin{aligned} \text{SWAP}(\text{CNOT}_{1 \rightarrow 2}(\text{SWAP}[a, b])) &= \text{SWAP}(\text{CNOT}_{1 \rightarrow 2}[b, a]) \\ &= \text{SWAP}[b, b \oplus a] = [b \oplus a, b]. \end{aligned}$$

### Lösung Übungsaufgabe 3.5

Nutzen wir Abb. 3.3, so können wir den Zustand des zweiten Bits nach dem Messen berechnen als

$$\frac{p_{a0}[0] + p_{a1}[1]}{p_{a0} + p_{a1}} = \frac{q_a r_0[0] + q_a r_1[1]}{q_a r_0 + q_a r_1} = \frac{r_0[0] + r_1[1]}{r_0 + r_1} = r_0[0] + r_1[1] = r,$$

wobei wir  $q_a$  gestrichen haben und benutzt haben, dass  $r_0 + r_1 = 1$ .

### Lösung Übungsaufgabe 3.6

Der Zustand vor der kontrollierten NOT-Operation ist

$$\left( \frac{1}{2}[0] + \frac{1}{2}[1] \right) \otimes [0] = \frac{1}{2}[00] + \frac{1}{2}[10].$$

Nachdem wir die kontrollierte NOT-Operation angewendet haben, erhalten wir

$$\text{CNOT}_{1 \rightarrow 2} \left( \frac{1}{2}[00] + \frac{1}{2}[10] \right) = \frac{1}{2}[00] + \frac{1}{2}[11].$$

### Lösung Übungsaufgabe 3.7

1.

$$\begin{aligned} |+\rangle \otimes |-\rangle &= \left( \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \otimes \left( \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle \right) \\ &= \frac{1}{2} |00\rangle - \frac{1}{2} |01\rangle + \frac{1}{2} |10\rangle - \frac{1}{2} |11\rangle. \end{aligned}$$

2.

$$\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle \otimes |+\rangle.$$

Es handelt sich also tatsächlich um einen Produktzustand.

### Lösung Übungsaufgabe 3.8

Wir starten mit  $|00\rangle$ . Nach der NOT-Operation auf dem zweiten Qubit erhalten wir  $|01\rangle$ . Die Hadamard-Operation auf dem ersten Qubit wandelt diesen Zustand um in  $|+\rangle \otimes |1\rangle = \frac{1}{\sqrt{2}} |01\rangle + \frac{1}{\sqrt{2}} |11\rangle$ . Mit der finalen Z-Operation erhalten wir den folgenden zwei-Qubit Zustand direkt vor dem Messen:

$$\frac{1}{\sqrt{2}} |01\rangle - \frac{1}{\sqrt{2}} |11\rangle.$$

Also erhalten wir entweder 01 oder 11, beide jeweils mit Wahrscheinlichkeit 50%.

### Lösung Übungsaufgabe 3.9

Wir zeigen nur, wie wir Gl. (3.39) verifizieren können (die andere Gleichung verläuft analog). Dafür schreiben wir  $|\alpha\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$  und  $|\beta\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle$ . Dann gilt

$$\begin{aligned} U_1(|\alpha\rangle \otimes |\beta\rangle) &= U_1(\alpha_0\beta_0 |00\rangle + \alpha_0\beta_1 |01\rangle + \alpha_1\beta_0 |10\rangle + \alpha_1\beta_1 |11\rangle) \\ &= \alpha_0\beta_0 U_1 |00\rangle + \alpha_0\beta_1 U_1 |01\rangle + \alpha_1\beta_0 U_1 |10\rangle + \alpha_1\beta_1 U_1 |11\rangle \\ &= \alpha_0\beta_0 U |0\rangle \otimes |0\rangle + \alpha_0\beta_1 U |0\rangle \otimes |1\rangle + \alpha_1\beta_0 U |1\rangle \otimes |0\rangle + \alpha_1\beta_1 U |1\rangle \otimes |1\rangle \\ &= \alpha_0 U |0\rangle \otimes (\beta_0 |0\rangle + \beta_1 |1\rangle) + \alpha_1 U |1\rangle \otimes (\beta_0 |0\rangle + \beta_1 |1\rangle) \\ &= \alpha_0 U |0\rangle \otimes |\beta\rangle + \alpha_1 U |1\rangle \otimes |\beta\rangle \\ &= (\alpha_0 U |0\rangle + \alpha_1 U |1\rangle) \otimes |\beta\rangle \\ &= U |\alpha\rangle \otimes |\beta\rangle \end{aligned}$$

durch Gl. (3.33), die Definition von  $U_1$  und Linearität.

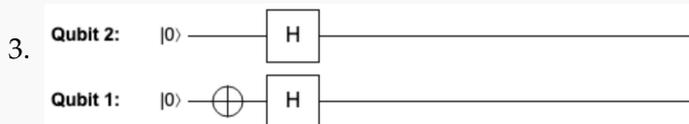
### Lösung Übungsaufgabe 3.10

1. Der Zustand kann geschrieben werden als

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

2. Dies ist äquivalent zu

$$H|1\rangle \otimes H|0\rangle = H\text{NOT}|0\rangle \otimes H|0\rangle = (H\text{NOT} \otimes H)|00\rangle.$$



### Lösung Übungsaufgabe 3.11

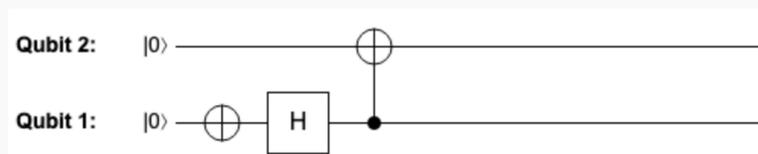
Um zu zeigen, dass  $HZ \neq ZH$ , reicht es, dass wir verifizieren, dass die Anwendung beider Operationen auf den Zustand  $|0\rangle$  unterschiedliche Ergebnisse liefert:

$$HZ|0\rangle = H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle,$$

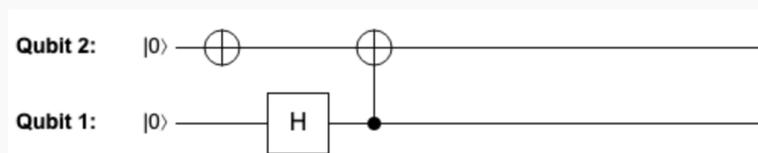
$$ZH|0\rangle = Z\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle.$$

### Lösung Übungsaufgabe 3.12

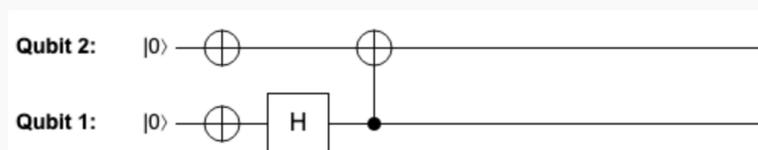
•  $|\Phi^-\rangle$ :



•  $|\Psi^+\rangle$ :



•  $|\Psi^-\rangle$ :



### Lösung Übungsaufgabe 3.13

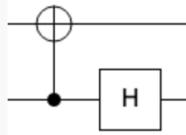
Beachte, dass dies das Gleiche ist, wie die Operation  $U_{\text{Bell}}$  in Gl. (3.56) zu invertieren. Erinnern wir uns, dass  $U_{\text{Bell}} = \text{CNOT}_{1 \rightarrow 2} (H \otimes I)$ . Dies ist eine Zusammensetzung von Operationen, also wissen wir durch Übungsaufgabe 2.5, dass

$$U_{\text{Bell}}^{-1} = (H \otimes I)^{-1} \text{CNOT}_{1 \rightarrow 2}^{-1} = (H^{-1} \otimes I) \text{CNOT}_{1 \rightarrow 2}^{-1}.$$

Durch Gl. (3.49) wissen wir, dass  $\text{CNOT}_{1 \rightarrow 2}$  selbstinvers ist, also  $\text{CNOT}_{1 \rightarrow 2}^{-1} = \text{CNOT}_{1 \rightarrow 2}$ . Es ist auch wahr, dass  $H^{-1} = H$  (dies gilt tatsächlich für jede Art von Reflektion). Dies impliziert, dass wir  $U_{\text{Bell}}$  rückgängig machen können, indem wir die beiden Operationen in umgekehrter Reihenfolge anwenden:

$$U_{\text{Bell}}^{-1} = (H \otimes I) \text{CNOT}_{1 \rightarrow 2}.$$

D.h., wir wenden zuerst  $\text{CNOT}_{1 \rightarrow 2}$  und dann  $H$  an, wie in:



### Lösung Übungsaufgabe 3.14

Beachte Gl. (3.52) bis (3.55): Die Bell Zustände unterscheiden sich nur durch Bit Flips und Vorzeichen. Erinnere dich dann, dass wir ein Bit umdrehen können durch eine lokale NOT-Operation und dass wir Vorzeichen einführen können, indem wir lokale Z-Operationen anwenden können, wo  $Z$  definiert ist wie in Gl. (2.12). Um  $|\Phi^-\rangle$  zu erzeugen, kann Alice einfach eine Z-Operation auf ihr Qubit anwenden:

$$(Z \otimes I) |\Phi^+\rangle = (Z \otimes I) \left( \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle \right) = \frac{1}{\sqrt{2}} |00\rangle - \frac{1}{\sqrt{2}} |11\rangle = |\Phi^-\rangle.$$

Um  $|\Psi^+\rangle$  zu erzeugen, wendet Alice stattdessen die NOT-Operation auf ihrem Qubit an:

$$(\text{NOT} \otimes I) |\Phi^+\rangle = (\text{NOT} \otimes I) \left( \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle \right) = \frac{1}{\sqrt{2}} |10\rangle + \frac{1}{\sqrt{2}} |01\rangle = |\Psi^+\rangle.$$

Um  $|\Psi^-\rangle$  zu erzeugen, wendet Alice erst die NOT-Operation an und dann die Z-Operation auf ihrem Qubit:

$$(Z \text{NOT} \otimes I) |\Phi^+\rangle = (Z \otimes I) \left( \frac{1}{\sqrt{2}} |10\rangle + \frac{1}{\sqrt{2}} |01\rangle \right) = -\frac{1}{\sqrt{2}} |10\rangle + \frac{1}{\sqrt{2}} |01\rangle = |\Psi^-\rangle.$$



## Quest 4: Quantenkompositionen

Letzte Woche haben wir über die Zustände zweier Quantenbits gesprochen und darüber, welche Operationen man auf Ihnen durchführen kann. Außerdem haben wir *verschränkte* Quanten Zustände kennengelernt und herausgefunden, dass man mit diesen effizienter kommunizieren (z.B. da zwei Bits übertragen werden, wenn man ein qubit sendet) und bei bestimmten Spielen deterministische oder probabilistische Strategien schlagen kann. Diese Woche werden wir mit beliebig vielen Quantenbits arbeiten und damit anfangen, Quantenschaltungen zu bauen, die aus vielen Operationen bestehen, um mit diesen interessante Probleme zu lösen.

### 4.1 Quantenschaltungen

Wenn man auf viele Qubits viele verschiedene Operationen anwendet, verliert man schnell den Überblick. Genau wie ein:e Komponist:in eine Partitur, also beschriebene Notenblätter, nutzen kann, um die Noten für jedes einzelne Instrument in einem Orchester zu jedem Zeitpunkt zuzuordnen, erlauben Quantenschaltungen uns eine bildliche Beschreibung, welche Operation auf welches Qubit zu einem bestimmten Zeitpunkt angewendet werden soll. Jede Operation ist sozusagen wie eine Note und die verschiedenen Quantenbits entsprechen den unterschiedlichen Instrumenten in einem Orchester!

Anders als in einem Orchester, können manche Quanten Operationen aber auch zwei (und manchmal sogar drei!) Quantenbits gleichzeitig beeinflussen. Das wäre, als würden wir eine Gitarristin bitten, auf der Geige zu klimpern, während der Geiger die Gitarre spielt! Solche Interaktionen zwischen den Musizierenden (oder Quantenbits) können viel Spaß machen und den Klang bereichern. In der Praxis können sie allerdings schwierig sein (vielleicht muss der Geiger komplett über die Bühne rennen, um die Gitarristin zu erreichen!) Außerdem könnten sich beispielsweise bei der hektischen Zusammenarbeit die Haare des Geigers in den großen Ohrringen der Gitarristin verfangen, weswegen die beiden das ganze Stück zusammen spielen müssten.

Je mehr Interaktionen zwischen den Qubits stattfinden, desto verschränkter werden sie normalerweise. Am Ende des Liedes sind oft alle Qubits so miteinander verschränkt, dass es unmöglich ist, sie zu unterscheiden. Die einzige Möglichkeit, sie zu sortieren und in einen sinnvollen Zustand zu bringen, besteht darin, sie zu messen! Genau so läuft eine Quantenberechnung ab. Formal besteht eine *Quantenschaltung* aus drei Teilen:

1. einem Initialzustand, typischerweise jedes Qubit im Zustand  $|0\rangle$ ,
2. einer Reihe von Quantenoperationen, wobei jede typischerweise auf wenigen Qubits gleichzeitig wirkt (meist ein oder zwei),
3. Messungen, um Qubits auszulesen (häufig werden alle Qubits gemessen).

Wir können Quantenschaltungen grafisch mit den selben Bildern darstellen, die wir schon von QUIRKY kennen. Wenn wir über Quantenschaltung sprechen, werden Operationen und Messungen häufig als **Quantengatter** oder **Quanten Gate** bezeichnet (z.B. "Hadamard Gatter" anstelle "Hadamard Operation"). Wir betrachten nun, wie sich diese drei Teile in einem großen Orchester von Qubits verhalten.

#### 4.1.1 Viele Quantenbits

Die Regeln, die wir bisher für ein und zwei Qubits gelernt haben, lassen sich ganz natürlich auf Quantensysteme mit vielen Qubits verallgemeinern. Zum Beispiel kann ein beliebiger Zustand

von drei Qubits wie folgt geschrieben werden:

$$|\psi\rangle = \psi_{000} |000\rangle + \psi_{001} |001\rangle + \psi_{010} |010\rangle + \psi_{011} |011\rangle + \psi_{100} |100\rangle + \psi_{101} |101\rangle + \psi_{110} |110\rangle + \psi_{111} |111\rangle, \quad (4.1)$$

wobei  $\psi_{ijk} \in [-1, 1]$  und die Quadrate dieser Amplituden sich wieder zu eins summieren, also

$$\psi_{000}^2 + \psi_{001}^2 + \psi_{010}^2 + \psi_{011}^2 + \psi_{100}^2 + \psi_{101}^2 + \psi_{110}^2 + \psi_{111}^2 = 1.$$

Insgesamt gibt es also  $8 = 2^3$  Amplituden, eine für jeden Bitstring der Länge drei. Wir können auch  $|\psi\rangle$  als einen Vektor mit acht Einträgen betrachten:

$$|\psi\rangle = \begin{pmatrix} \psi_{000} \\ \psi_{001} \\ \psi_{010} \\ \psi_{011} \\ \psi_{100} \\ \psi_{101} \\ \psi_{110} \\ \psi_{111} \end{pmatrix}.$$

Im Allgemeinen kann ein Zustand mit  $n$  Qubits mit  $2^n$  Amplituden  $\psi_{a_1, \dots, a_n}$  beschrieben werden, eine für jeden Bitstring der Länge  $n$ :

$$|\psi\rangle = \psi_{00\dots 00} |00\dots 00\rangle + \psi_{00\dots 01} |00\dots 01\rangle + \dots + \psi_{11\dots 11} |11\dots 11\rangle \quad (4.2)$$

Auch hier sollte jede Amplitude  $\psi_{a_1, \dots, a_n}$  in  $[-1, 1]$  liegen und ihre Summe sollte eins ergeben:

$$\psi_{00\dots 00}^2 + \psi_{00\dots 01}^2 + \dots + \psi_{11\dots 11}^2 = 1 \quad (4.3)$$

Falls manche der Amplituden null sein sollten, können wir sie einfach weglassen. Zum Beispiel hat der fünf-Qubit Quantenzustand

$$\frac{1}{\sqrt{2}} (|00000\rangle + |11111\rangle)$$

32 Amplituden, von denen aber 30 null sind.

Da es  $2^n$  Amplituden gibt, können wir uns  $|\psi\rangle$  auch als Vektor in einem  $2^n$  dimensionalem Raum vorstellen. Was bedeutet Gl. (4.3) geometrisch? Für ein einzelnes Qubit haben wir in §2.1.2 gesehen, dass die Zustände Punkten auf dem Einheitskreis entsprechen, also zweidimensionalen Vektoren der Länge eins. Nach dem Satz des Pythagoras gilt in jeder Dimension, dass die Summe der Quadrate aller Einträge eines Vektors das Quadrat seiner Länge ist. Also bedeutet Gl. (4.3) geometrisch, dass  $|\psi\rangle$  einem Vektor der *Länge eins* oder einem *Einheitsvektor* in einem  $2^n$  dimensionalem Raum entspricht.

Man beachte, dass die Anzahl der Amplituden schnell mit der Anzahl der Qubits wächst. Das erklärt, warum es schnell unmöglich wird, Quantenzustände auf einem klassischen Computer zu speichern. So bräuchte man beispielsweise zum Darstellen eines Quantenzustands mit  $n = 300$  Qubits schon mehr Amplituden als es Atome im beobachtbaren Universum gibt! Deswegen kann man in QUIRKY nicht mehr als 10 Qubits nutzen, wir wollen ja nicht, dass deinem Browser der Speicher ausgeht.

Wie schon in Gl. (3.33) können wir das **Tensorprodukt** " $\otimes$ " nutzen, um Quantenzustände auf einer beliebigen Anzahl an Qubits zu kombinieren. Wenn wir zwei Basiszustände haben, definieren wir ihr Tensorprodukt einfach als die Konkatenation (Aneinanderreihung) der Bitstrings. So verallgemeinern wir den zwei-Qubit Fall aus Gl. (3.32) wie folgt:

$$|a_1, \dots, a_n\rangle \otimes |b_1, \dots, b_m\rangle = |a_1, \dots, a_n, b_1, \dots, b_m\rangle. \quad (4.4)$$

Zum Beispiel,

$$|101\rangle \otimes |01\rangle = |10101\rangle.$$

Im Allgemeinen gilt, wenn  $|\alpha\rangle$  ein beliebiger Quantenzustand von  $n$  Qubits und  $|\beta\rangle$  ein beliebiger Quantenzustand von  $m$  Qubits ist, dann ist ihr Tensorprodukt ein Zustand von  $n + m$  Qubits. Um diesen Zustand zu berechnen, können wir einfach "ausmultiplizieren" indem wir das Distributivgesetz anwenden und anschließend Gl. (4.4) auf jeden Term anwenden. Beispielsweise lässt sich das Tensorprodukt zweier maximal verschränkter Quantenzustände so berechnen:

$$\begin{aligned} & |\Phi^+\rangle \otimes |\Phi^+\rangle \\ &= \left( \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle \right) \otimes \left( \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle \right) \\ &= \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} |00\rangle \otimes |00\rangle + \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} |00\rangle \otimes |11\rangle + \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} |11\rangle \otimes |00\rangle + \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} |11\rangle \otimes |11\rangle \\ &= \frac{1}{2} |0000\rangle + \frac{1}{2} |0011\rangle + \frac{1}{2} |1100\rangle + \frac{1}{2} |1111\rangle. \end{aligned}$$

### Übungsaufgabe 4.1: Tensorprodukt der Bell-Zustände

Berechne das Tensorprodukt  $|\Phi^-\rangle \otimes |\Psi^-\rangle$  der zwei Bell-Zustände aus Gl. (3.53) und (3.55).

#### 4.1.2 Operationen

Welche Quantenoperationen können wir auf mehreren Qubits durchführen? Einerseits können wir jede in Abschnitte 2 und 3 besprochene Ein-Qubit- oder Zwei-Qubit-Operation auf alle ausgewählten Qubits eines Zustands mit vielen Qubits anwenden. Das funktioniert genauso, wie in §3.2.2.

Wenn beispielsweise  $U$  eine Operation auf einem Qubit ist, also eine Drehung oder Spiegelung, dann können wir eine Quantenoperation  $U_1$  definieren, die der Anwendung von  $U$  auf das erste Qubit eines  $n$ -Qubit-Zustands entspricht. Diese Operation ist wie folgt auf den Basiszuständen definiert:

$$U_1 |a_1, \dots, a_n\rangle = U |a_1\rangle \otimes |a_2, \dots, a_n\rangle.$$

Beachte dabei, dass das Tensorprodukt den Ein-Qubit-Zustand  $U |a_1\rangle$  mit dem  $(n - 1)$ -Qubit-Basiszustand  $|a_2, \dots, a_n\rangle$  kombiniert und so einen  $n$ -Qubit-Zustand bildet, wie wir beabsichtigt hatten. Wie sonst auch, erweitern wir  $U_1$  durch Linearität auf allgemeine Quantenzustände mit  $n$  Qubits. Genauso definieren wir  $U_2, U_3$ , usw., die der Anwendung von  $U$  auf das zweite, dritte, usw. Qubit entsprechen.

### Übungsaufgabe 4.2: Eine Ein-Qubit-Operation anwenden

Berechne das Ergebnis der Anwendung der Hadamard-Operation auf das zweite Qubit des Drei-Qubit-Zustands  $|\Phi^+\rangle \otimes |1\rangle$ . Anders gesagt, berechne  $H_2(|\Phi^+\rangle \otimes |1\rangle)$ . Schreibe dein Ergebnis in der Form aus Gl. (4.1).

Auf ähnliche Weise können wir herausfinden, wie eine Zwei-Qubit-Operation auf zwei ausgewählte Qubits eines  $n$ -Qubit-Zustands angewendet werden kann. Hauptsächlich interessieren uns kontrollierte-NOT-Operationen:  $\text{CNOT}_{k \rightarrow l}$  mit  $k \neq l$  ist die Operation, die das  $l$ -te Qubit (das Ziel-Qubit) abhängig vom  $k$ -ten Qubit (dem Steuer-Qubit) flippt. Mathematisch gesehen, wirkt sich die Operation wie folgt auf die Basiszustände aus:

$$\text{CNOT}_{k \rightarrow l} |a_1, \dots, a_l, \dots, a_n\rangle = |a_1, \dots, a_l \oplus a_k, \dots, a_n\rangle,$$

und wir können diese Formel durch Linearität auf beliebige  $n$ -Qubit-Zustände erweitern. Beispielsweise ist die kontrollierte-NOT-Operation  $\text{CNOT}_{1 \rightarrow 3}$  wie folgt auf Vier-Qubit-Basiszuständen definiert:

$$\text{CNOT}_{1 \rightarrow 3} |a_1, a_2, a_3, a_4\rangle = |a_1, a_2, a_3 \oplus a_1, a_4\rangle.$$

Wie schaut das ganze in QUIRKY aus? Lass uns zu

<https://www.quantum-quest.org/quirky>

gehen und auf "Quest 4" klicken, um das herauszufinden. Dein Browser zeigt dann ein Bild ähnlich zu Abb. 4.1.

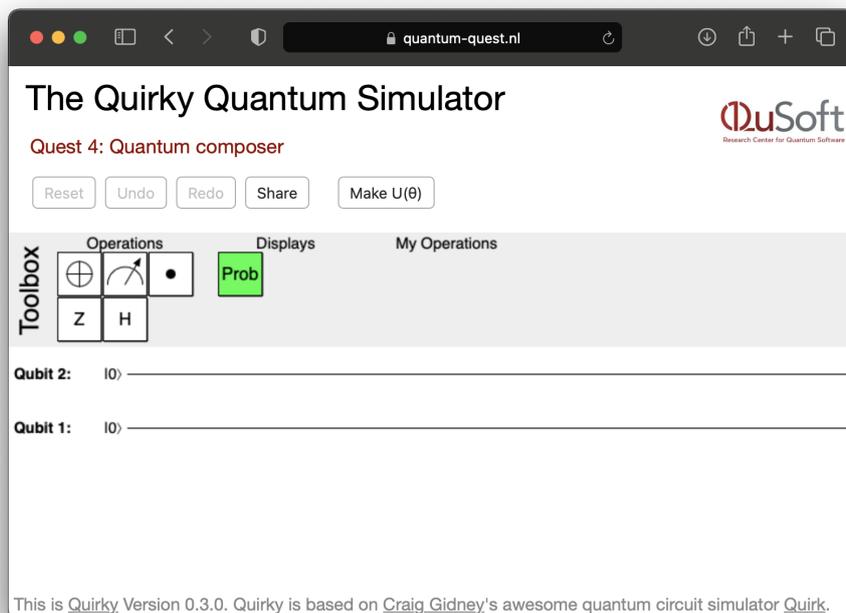
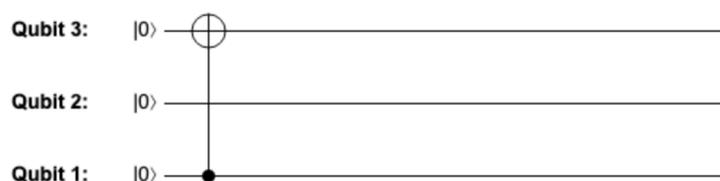


Abbildung 4.1: QUIRKY für Quest 4.

Moment mal, QUIRKY sieht doch genau wie letzte Woche aus?! Wenn du allerdings eine Operation aus der Toolbox nimmst, erscheint ein neuer Draht – also ein neues Qubit. (Natürlich ist die Anzahl der Qubits so beschränkt, dass dein Browser nicht überfordert wird!) Warum versuchst du nicht mal, die  $\text{CNOT}_{1 \rightarrow 3}$ -Operation wie im folgenden Bild zu erstellen?



Wenn Quantenoperationen sich auf unterschiedliche Qubits auswirken, können wir sie parallel durchführen. Wie auch in §3.2.3 verwenden wir hierfür das Symbol des Tensorprodukts. Wenn  $U$  eine Quantenoperation auf  $n$  Qubits, und  $V$  eine Quantenoperation auf  $m$  Qubits ist, dann können wir eine Quantenoperation  $U \otimes V$  auf  $n + m$  Qubits definieren, welche der parallelen Ausführung beider Operationen entspricht. Auf den Basiszuständen gilt,

$$(U \otimes V) |a_1, \dots, a_n, b_1, \dots, b_m\rangle = U |a_1, \dots, a_n\rangle \otimes V |b_1, \dots, b_m\rangle, \quad (4.5)$$

was wir wieder durch Linearität auf allgemeine Zustände erweitern können. Aus Gl. (4.5) folgt dann

$$(U \otimes V)(|\alpha\rangle \otimes |\beta\rangle) = U|\alpha\rangle \otimes V|\beta\rangle,$$

aber nur, wenn  $|\alpha\rangle$  ein  $n$ -Qubit-Zustand und  $|\beta\rangle$  ein  $m$ -Qubit-Zustand ist! In der folgenden Übung passen die Tensorprodukte *nicht* so aufeinander, weswegen du diese Regel dort *nicht* verwenden kannst!

### Übungsaufgabe 4.3: Versetzte Tensorprodukte

Betrachte den Drei-Qubit-Zustand  $(\text{CNOT}_{2 \rightarrow 1} \otimes I)(|0\rangle \otimes |\Phi^-\rangle)$ .

1. Wie kannst du diesen Zustand mit QUIRKY erstellen?
2. Schreibe den Zustand in der Form aus Gl. (4.1).

Wir können das Tensorprodukt wiederholt nutzen, um iterativ größer und größer werdende Quantenoperationen zu bauen. Hier siehst du drei Beispiele für verschiedene Anzahlen an Qubits:

1.  $I \otimes I \otimes U \otimes I$  ist die selbe Vier-Qubit-Operation wie  $U_3$ ,
2.  $I \otimes \text{CNOT}_{1 \rightarrow 2} \otimes I \otimes I$  ist die kontrollierte-NOT-Operation  $\text{CNOT}_{2 \rightarrow 3}$  für fünf Qubits,
3.  $Z \otimes I \otimes X$  ist die Quantenoperation, welche  $Z$  auf das erste Qubit, und parallel  $X$  auf das dritte Qubit anwendet (wir könnten die Operation auch als  $Z_1 X_3$  oder  $X_3 Z_1$  schreiben).

#### 4.1.3 Die allgemeinsten Quantenoperationen

Was sind die allgemeinsten Operationen, die wir auf Quantenzustände mit  $n$  Qubits anwenden können? Jede Operation, die die folgenden drei Eigenschaften aufweist:

1. sie ist linear,
2. sie bildet Quantenzustände auf Quantenzustände ab,
3. sie ist invertierbar

ist eine gültige Quantenoperation!

### Übungsaufgabe 4.4: Toffoli

Definiere die **Toffoli-Operation** auf drei Qubits durch

$$T|a, b, c\rangle = |a, b, c \oplus ab\rangle$$

auf Basiszuständen ( $ab$  ist dabei das Produkt der zwei Bits  $a, b \in \{0, 1\}$ , und  $\oplus$  wurde in Gl. (3.20) definiert), und erweitere sie durch Linearität auf beliebige Drei-Qubit-Zustände. Zeige, dass  $T$  alle Quantenzustände auf Quantenzustände abbildet, und dass  $T$  invertierbar ist.

**Bemerkung:**  $T$  invertiert das dritte Bit genau dann, wenn beide ersten Bits beide eins sind – es ist also eine “zweifach-kontrollierte“-NOT-Operation.

Übungsaufgabe 4.4 zeigt, dass die Toffoli-Operation eine gültige Quantenoperation auf drei Qubit ist. Interessanterweise ist es tatsächlich möglich  $T$  als eine Reihe von Ein- und Zwei-Qubit-Operationen zu schreiben. Tatsächlich ist das für *jede* Quantenoperation auf  $n$  Qubits möglich – aber das werden wir nicht in diesem Kurs lernen, da man viel Erfahrung mit Quantenoperationen braucht, um zu verstehen, wie das funktioniert!

#### 4.1.4 Regeln für Schaltungen

Wenn man es mit einer sehr komplizierten Quantenschaltung zu tun hat, hilft es, einige Tricks zur Vereinfachung der Dinge auf Lager zu haben. Solche Tricks erleichtern nicht nur das Verständnis der Schaltung, sondern machen sie auch effizienter und damit schneller für die Ausführung auf einem Quantencomputer. Schauen wir uns ein paar einfache Beispiele für solche Tricks an, die nur ein einziges Qubit betreffen.

Aus §2.4.3 wissen wir, dass jede Ein-Qubit-Operation entweder eine Drehung oder eine Spiegelung ist. Visuell können wir diese Operationen darstellen, wenn wir uns an §2.1.2 erinnern, wo wir gesehen haben, dass Quantenzustände den Einheitskreis bilden. Eine Beobachtung ist, dass das zweifache Anwenden einer bestimmten Spiegelung auf das gleiche Qubit den Zustand unverändert lässt. Dies ist in der Tat visuell klar, da eine doppelte Spiegelung um dieselbe Achse alles wiederherstellt, wie es war (dies kann man zum Beispiel in Abb. 2.4 für die Operation NOT sehen). Dies gilt insbesondere für die Hadamard-Operation  $H$ , von der wir aufgrund von Gl. (2.21) wissen, dass sie eine Spiegelung ist. Wir wollen diese geometrische Intuition durch eine kleine Berechnung verifizieren und auch zeigen, dass das Hadamard-Gatter eine Umwandlung zwischen den NOT- und Z-Gattern ermöglicht.

##### Übungsaufgabe 4.5: Z und NOT

Aus Gl. (2.20) wissen wir, dass das Hadamard-Gate  $H$  sich wie folgt auswirkt:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle, \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle.$$

1. Prüfe, dass ein erneutes anwenden von  $H$  wieder zu den Zuständen  $|0\rangle$  und  $|1\rangle$  führt.

$$H|+\rangle = |0\rangle, \quad H|-\rangle = |1\rangle.$$

2. Zeige, dass  $HZH = \text{NOT}$ , wobei  $Z$  in Gl. (2.12) definiert ist.
3. Zeige, dass  $H\text{NOT}H = Z$ .

Auch interessant ist die Frage, was passiert, wenn man zwei beliebige Drehungen oder Spiegelungen nacheinander anwendet. Klar ist, dass es wieder eine Drehung oder Spiegelung sein muss. Aber was von beiden, und mit welchem Winkel? Zwei aufeinanderfolgende Drehungen entsprechen einfach einer Drehung um die Summe der beiden Winkel, also  $U(\varphi_2)U(\varphi_1) = U(\varphi_1 + \varphi_2)$ . Die folgende Übung gibt dir eine Regel zur Vereinfachung zwei aufeinanderfolgende Spiegelungen zu einer einzelnen *Drehung*.

##### Übungsaufgabe 4.6: Spiegelungen und Drehungen (optional)

Zeige, dass das Produkt zweier Spiegelungen eine Rotation ist. Zeige also, dass

$$V(\theta_2)V(\theta_1) = U(\theta),$$

für einen Winkel  $\theta$ . Kannst du  $\theta$  relativ zu  $\theta_1$  und  $\theta_2$  bestimmen?

**Hint:** Nutze Gl. (2.19) und die Gleichung  $U(\varphi_2)U(\varphi_1) = U(\varphi_1 + \varphi_2)$ .

Die zwei übrigen Fälle, die eine Drehung und eine Spiegelung beinhalten kannst du selbst ausrechnen und prüfen, dass beide zu einer Spiegelung  $V(\theta)$  für einen bestimmten Winkel  $\theta$  führen.

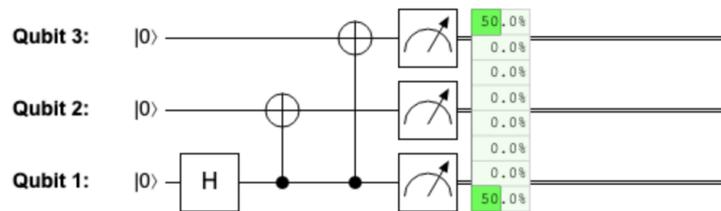
#### 4.1.5 Alle Qubits messen

Nachdem wir fertig damit sind, Operationen auf unsere Qubits anzuwenden, wollen wir Informationen auslesen. Wie bisher geht das nur durch das Messen der Qubits.

Welche Regeln gelten beim messen eines Quantenzustands mit  $n$  Qubits? Wenn wir **alle  $n$  Qubits messen**, erhalten wir  $n$  Bits als Ergebnis, also einen Bitstring  $a_1 \dots a_n$ . Wie vorher auch ist die Wahrscheinlichkeit ein bestimmtes Ergebnis  $a_1 \dots a_n$  zu erhalten das Quadrat der Amplitude:

$$p_{a_1, \dots, a_n} = \psi_{a_1, \dots, a_n}^2. \quad (4.6)$$

Mit QUIRKY können wir alle Qubits messen, indem wir für jedes Qubit eine Messung hinzufügen. Um die Wahrscheinlichkeiten der Messergebnisse anzusehen, können wir eine Wahrscheinlichkeitsanzeige hinzufügen – allerdings müssen wir diese so vergrößern, dass sie alle Drähte umfasst. Versuche doch, das folgende Beispiel nachzubauen:



Diese Reihe von QUIRKY-Operationen erstellt den Zustand

$$\frac{1}{\sqrt{2}} |000\rangle + \frac{1}{\sqrt{2}} |111\rangle \quad (4.7)$$

und misst anschließend alle drei Qubits. Verstehst du, warum das so funktioniert?

#### 4.1.6 Einzelne Qubits messen

Natürlich können wir auch nur eine Teilmenge der Qubits messen. (Das haben wir letzte Woche nicht mal für zwei Qubits besprochen, da es so viel anderes zu tun gab.) Nehmen wir beispielsweise mal an, wir haben einen Drei-Qubit-Zustand  $|\psi\rangle$ , wie den in Gl. (4.1), aber anstelle alle drei Qubits zu messen, *misst du nur das erste Qubit*. Mit welcher Wahrscheinlichkeit  $p_a$  erhältst du das Ergebnis  $a \in \{0, 1\}$ ? Es ist einfach die Summe aller Wahrscheinlichkeiten aus Gl. (4.6) die einem Bitstring entsprechen, der mit  $a$  beginnt:

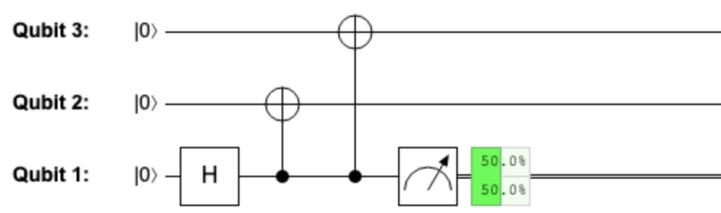
$$p_a = \psi_{a00}^2 + \psi_{a01}^2 + \psi_{a10}^2 + \psi_{a11}^2. \quad (4.8)$$

+ Wenn wir beispielsweise das erste Qubit des Zustands

$$\frac{1}{\sqrt{8}} |000\rangle + \sqrt{\frac{2}{8}} |010\rangle + \sqrt{\frac{5}{8}} |111\rangle$$

messen, erhalten wir das Ergebnis 0 mit Wahrscheinlichkeit  $1/8 + 2/8 = 3/8$ .

In QUIRKY können wir einzelne Qubits messen, indem wir Messungen auf die Drähte ziehen, an denen wir interessiert sind. Um die Wahrscheinlichkeiten der Messergebnisse anzusehen, ziehen wir eine Wahrscheinlichkeitsanzeige auf den Schaltkreis. Probier's doch direkt mal aus! Die folgende Abfolge von Operationen erstellt beispielsweise den Zustand aus Gl. (4.7) und misst nur das erste Qubit:



(4.9)

Tatsächlich, laut Gl. (4.8) sollten wir 0 und 1 mit gleicher Wahrscheinlichkeit erhalten.

In welchem Quantenzustand befinden sich aber das zweite und dritte Qubit, nachdem du das erste Qubit gemessen und als Ergebnis  $a \in \{0, 1\}$  erhalten hast? Nach dem gleichen Verfahren wie bei probabilistischen Bits aus §3.1.3 sammeln wir zunächst alle Terme von  $|\psi\rangle$ , bei denen sich das erste Qubit in dem korrekten Zustand befindet:

$$\psi_{a00} |a00\rangle + \psi_{a01} |a01\rangle + \psi_{a10} |a10\rangle + \psi_{a11} |a11\rangle .$$

Anschließend lassen wir das erste Qubit aus allen vier Termen einfach weg, da es schon gemessen wurde:

$$\psi_{a00} |00\rangle + \psi_{a01} |01\rangle + \psi_{a10} |10\rangle + \psi_{a11} |11\rangle .$$

Schlussendlich normalisieren wir das Ergebnis noch, um einen gültigen Zwei-Qubit-Zustand zu erhalten. Dafür müssen wir die Zahl  $c$  finden, sodass  $\frac{\psi_{a00}}{c} |00\rangle + \frac{\psi_{a01}}{c} |01\rangle + \frac{\psi_{a10}}{c} |10\rangle + \frac{\psi_{a11}}{c} |11\rangle$  ein gültiger Quantenzustand ist, also,

$$\left(\frac{\psi_{a00}}{c}\right)^2 + \left(\frac{\psi_{a01}}{c}\right)^2 + \left(\frac{\psi_{a10}}{c}\right)^2 + \left(\frac{\psi_{a11}}{c}\right)^2 = 1.$$

Das globale Vorzeichen ist dabei unwichtig, wir können also einfach

$$c = \sqrt{\psi_{a00}^2 + \psi_{a01}^2 + \psi_{a10}^2 + \psi_{a11}^2},$$

nutzen, was die Quadratwurzel der Wahrscheinlichkeit aus Gl. (4.8) ist.

Zusammenfassend, wenn du das **erste Qubit eines Drei-Qubit-Zustandes misst** wie in Gl. (4.1), dann erhältst du das Ergebnis  $a \in \{0, 1\}$  mit Wahrscheinlichkeit

$$p_a = \psi_{a00}^2 + \psi_{a01}^2 + \psi_{a10}^2 + \psi_{a11}^2 \quad (4.10)$$

und der resultierende Zwei-Qubit-Zustand  $|\psi_a\rangle$  auf den übrig bleibenden Qubits ist

$$|\psi_a\rangle = \frac{\psi_{a00} |00\rangle + \psi_{a01} |01\rangle + \psi_{a10} |10\rangle + \psi_{a11} |11\rangle}{\sqrt{\psi_{a00}^2 + \psi_{a01}^2 + \psi_{a10}^2 + \psi_{a11}^2}}. \quad (4.11)$$

Was bedeutet das im Falle von (4.9), wo wir den Zustand  $\frac{1}{\sqrt{2}} |000\rangle + \frac{1}{\sqrt{2}} |111\rangle$  erzeugen und anschließend das erste Qubit messen? Wenn das Messergebnis 0 ist (was mit Wahrscheinlichkeit  $1/2$  passiert), sind die zwei übrigen Qubits im Zustand

$$\frac{\frac{1}{\sqrt{2}} |00\rangle}{\sqrt{\frac{1}{2}}} = |00\rangle .$$

Wenn das Messergebnis stattdessen 1 ist, sind die übrigen Qubits analog im Zustand  $|11\rangle$ .

Da das Messen eines Qubits von vielen schwierig sein kann, wollen wir uns noch eine andere Methode anschauen. Betrachten wir wieder einen allgemeinen Drei-Qubit-Zustand  $|\psi\rangle$  wie den in Gl. (4.1) und nehmen an, dass wir das erste Qubit messen wollen. Dann können wir die acht Terme im Ausdruck von  $|\psi\rangle$  wie folgt umschreiben:

$$\begin{aligned} |\psi\rangle &= \sqrt{p_0} |0\rangle \otimes \frac{\psi_{000} |00\rangle + \psi_{001} |01\rangle + \psi_{010} |10\rangle + \psi_{011} |11\rangle}{\sqrt{p_0}} \\ &+ \sqrt{p_1} |1\rangle \otimes \frac{\psi_{100} |00\rangle + \psi_{101} |01\rangle + \psi_{110} |10\rangle + \psi_{111} |11\rangle}{\sqrt{p_1}}. \end{aligned} \quad (4.12)$$

Wir können das nun schreiben als

$$|\psi\rangle = \sqrt{p_0} |0\rangle \otimes |\psi_0\rangle + \sqrt{p_1} |1\rangle \otimes |\psi_1\rangle, \quad (4.13)$$

wobei die  $p_a$  Wahrscheinlichkeiten sind (genauer gesagt die aus Gl. (4.10)) und die  $|\psi_a\rangle$  Quantenzustände sind (die Zwei-Qubit-Zustände aus Gl. (4.11)).

Tatsächlich kann man die Wahrscheinlichkeiten der Messergebnisse und den übrig bleibenden Quantenzustand immer dann auf diese Weise ablesen, wenn es gelingt den Quantenzustand in der Form von Gl. (4.13) zu schreiben. Zum Beispiel,

$$\frac{1}{\sqrt{2}} |000\rangle + \frac{1}{\sqrt{2}} |111\rangle = \frac{1}{\sqrt{2}} |0\rangle \otimes |00\rangle + \frac{1}{\sqrt{2}} |1\rangle \otimes |11\rangle,$$

Das bestätigt noch einmal, das in unserem Beispiel (4.9) beide Ergebnisse mit Wahrscheinlichkeit 1/2 auftreten und dass der übrige bleibende Quantenzustand entweder  $|00\rangle$  oder  $|11\rangle$  ist, je nach Messergebnis. Diese Methode, die Terme zuerst zu gruppieren und anschließend zu normalisieren ist sehr intuitiv und im Allgemeinen sehr hilfreich. Man muss dabei allerdings darauf achten, die Zustände richtig zu normalisieren! Soll heißen, welche Konstanten  $\sqrt{p_a}$  auch immer für die zwei Basiszustände aus Gl. (4.12) und (4.13) gezogen werden sollten  $p_0 + p_1 = 1$  erfüllen und die Zustände  $|\psi_0\rangle$  und  $|\psi_1\rangle$  auf den übrigen Qubits sollten korrekt normalisiert werden, siehe Gl. (4.3).

Wir können analog vorgehen, wenn wir mehr als drei Qubits haben oder ein anderes als das erste Qubit messen wollen oder wenn wir mehr als ein einzelnes Qubit messen wollen! Nehmen wir beispielsweise an, wir wollen die ersten *zwei* Qubits des allgemeinen Drei-Qubit-Zustandes  $|\psi\rangle$  aus Gl. (4.1) messen. Dann besteht das Messergebnis aus zwei Bits,  $a$  und  $b$ , welche mit den Wahrscheinlichkeiten

$$p_{a,b} = \psi_{ab0}^2 + \psi_{ab1}^2, \quad (4.14)$$

auftreten und das übrig gebliebene Qubit ist nach der Messung im Zustand

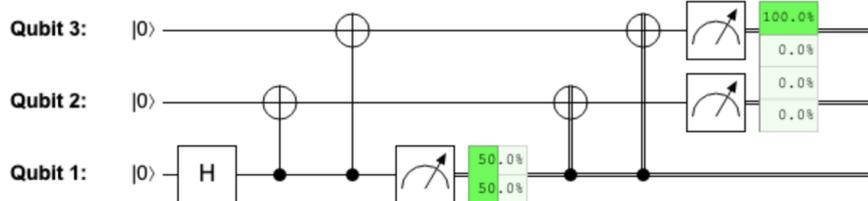
$$|\psi_{a,b}\rangle = \frac{\psi_{ab0} |0\rangle + \psi_{ab1} |1\rangle}{\sqrt{\psi_{ab0}^2 + \psi_{ab1}^2}}. \quad (4.15)$$

### Übungsaufgabe 4.7: Zwei von Drei

Mit welchen Wahrscheinlichkeiten ergibt das Messen der ersten beiden Qubits des Drei-Qubit-Zustands aus Gl. (4.7) welche Messergebnisse? Überprüfe dein Ergebnis mit QUIRKY.

Wenn wir einige der Qubits messen, andere aber nicht, werden wir oft das **Messergebnis** verwenden wollen, um zu bestimmen, ob eine Operation auf die verbleibenden Qubits angewendet werden soll oder nicht. Nehmen wir zum Beispiel an, dass man in der Situation von (4.9) die verbleibenden zwei Qubits auf den Zustand  $|00\rangle$  zurücksetzen möchte. Wenn das Ergebnis der Messung null ist, muss nichts getan werden. Ist das Messergebnis jedoch eins, dann wissen wir, dass sich die beiden verbleibenden Qubits im Zustand  $|11\rangle$  befinden und wir möchten sie wieder auf  $|00\rangle$  zurücksetzen. Das können wir durch Anwendung einer NOT-Operation auf beide Qubits erreichen. Woher wissen wir jedoch, ob wir diese Operation wirklich anwenden sollten oder nicht, das hängt schließlich von dem früheren Messergebnis des ersten Qubits ab. Daher möchten wir sie nur anwenden, wenn das Messergebnis 1 ist. Mit anderen Worten, wir wollen ein kontrolliertes-NOT-Gate anwenden, bei dem das Kontroll-Bit nun ein klassisches Bit ist (das Ergebnis der Messung), das Ziel-Bit aber immer noch ein Qubit ist.

In QUIRKY können wir das genau so wie erwartet lösen, indem man einfach ein klassisches Bit als Kontrol-Bit und ein Qubit als Ziel-Bit verwendet, wie man hier sieht:



Hier werden, nachdem das erste Qubit gemessen wurde, zwei weitere kontrollierte-NOT-Gatter angewandt, die von dem Messergebnis kontrolliert werden und im Anschluss werden die übrigen zwei Qubit gemessen. Das Bild zeigt, dass die Qubits tatsächlich erfolgreich zurückgesetzt wurden, da das Messergebnis mit 100% [00] ist.

Wenn wir wollten, könnten wir diese Operationen auf "hybriden" Zuständen beschreiben, die aus einem Bit und zwei Qubits bestehen, z.B.,

$$\text{CNOT}_{1 \rightarrow 2}[a] \otimes |b, c\rangle = [a] \otimes |a \oplus b, c\rangle,$$

aber wir brauchen diese Formalität nicht.

## 4.2 Quanten-Überraschungen

Im Anschluss werden wir über interessante Phänomene sprechen, die im Umgang mit Quantenbits auftreten. Die folgenden Abschnitte können größtenteils unabhängig voneinander gelesen werden, fang also ruhig mit dem Thema an, das dich am meisten interessiert.

### 4.2.1 Unklonbarkeit

Mit einem klassischen Bit ist *kopieren* oder *klonen* sehr einfach – wir müssen es einfach nur anschauen und kopieren, was wir sehen:

$$\begin{aligned} [0] &\mapsto [00], \\ [1] &\mapsto [11]. \end{aligned}$$

Können wir Qubits auch klonen?

Nehmen wir einmal an, dass es möglich ist Qubits zu klonen. Das würde bedeuten, es existiert eine Quantenoperation  $C$ , die gegeben eines beliebigen Ein-Qubit-Zustand  $|\psi\rangle$  und eines neuen Qubits im Zustand  $|0\rangle$  sich so verhält

$$C(|\psi\rangle \otimes |0\rangle) = |\psi\rangle \otimes |\psi\rangle \tag{4.16}$$

dass sie zwei Kopien von  $|\psi\rangle$  aus einer einzelnen Kopie erstellt. (Warum geben wir das neue Qubit? Ganz einfach, damit  $C$  genauso viele Eingabe wie Ausgabe Qubits hat).

Der Kloner würde sich beispielsweise so auf den Basiszuständen verhalten:

$$\begin{aligned} C|00\rangle &= |00\rangle, \\ C|10\rangle &= |11\rangle. \end{aligned} \tag{4.17}$$

Genau wie es einfach ist, klassische Bits zu klonen, können wir auch einfach eine Quantenoperation finden, die die Basiszustände kloniert. Die kontrollierte-NOT-Operation  $\text{CNOT}_{1 \rightarrow 2}$  aus Gl. (3.46) macht zum Beispiel genau das.

Aber gibt es auch eine Quantenoperation die beliebige unbekannte Quantenzustände klonen kann, nicht nur die Basiszustände? In der folgenden Hausaufgabe wirst du zeigen, dass das *nicht* möglich ist.

### Hausaufgabe 4.1: Unklonbarkeit

In dieser Hausaufgabe wollen wir beweisen, dass es keine Quantenoperation  $C$  geben kann, die Gl. (4.16) erfüllt. Wir nutzen dafür einen Trick, der sich Widerspruchsbeweis nennt. Das bedeutet, wir werden zeigen, dass die Existenz einer Klon-Operation  $C$  etwas impliziert, von dem wir wissen, dass es nicht stimmt (z.B. "0 = 1"). Daraus können wir dann schließen, dass kein solches  $C$  existieren kann.

Lass uns also zu Beginn annehmen, dass es eine Quantenoperation  $C$  gibt, die Gl. (4.16) erfüllt. Nun kannst du  $C(|+\rangle \otimes |0\rangle)$  auf zwei verschiedene Arten berechnen:

1. Nutze zuerst Gl. (4.16) und schreibe das Ergebnis dann in der Form von Gl. (3.30).
2. Schreibe erst  $|+\rangle \otimes |0\rangle$  in der Form von Gl. (3.30), nutze dann die Linearität von  $C$  und wende abschließend Gl. (4.16) an.

Erhältst du in beiden Fällen das gleiche Ergebnis? Wenn nicht, was kannst du daraus schließen?

Dieses berühmte Ergebnis ist bekannt als das **No-Cloning-Theorem** (engl. für "Unklonbarkeits-Theorem"). Die gleiche Schlussfolgerung (und wahrscheinlich auch das Argument, dass du in Hausaufgabe 4.1 angewendet hast) gilt auch für probabilistische Bits! Hier hast du eine intuitive Erklärung, warum wir weder Probabilistische- noch Quanten-Information klonen können. Wenn das möglich wäre, könnten wir mit einem unbekanntem probabilistischen Bit  $p$  oder einem Qubit in unbekanntem Zustand  $|\psi\rangle$  so viele Kopien von  $p$  und  $|\psi\rangle$  erstellen wie wir wollen. Diese Kopien könnten wir dann so lange auf verschiedenste Art und Weise messen, bis wir eine beliebig genaue Annäherung an die Wahrscheinlichkeiten oder Amplituden haben (genau wie wir es in §2.5.1 gemacht haben, um das Mysterium der gelben Box zu lösen). Also könnten wir aus einem einzelnen Probabilistischen- oder Quanten-Bit eine beliebige Menge an Information lernen. Das sollte offensichtlich nicht möglich sein!

Tatsächlich würden wir in einer sehr seltsamen Welt leben, wenn das möglich wäre (sogar noch seltsamer als die Welt der Quantenmechanik)! Wir könnten dann zum Beispiel eine Münze erstellen, deren Wahrscheinlichkeit für Kopf  $p = 0.1011010010\dots$  ist, wobei der Bitstring den gesamten Inhalt von Wikipedia und alle YouTube Videos und Katzenbilder im Internet kodiert. Wenn das Klonen von probabilistischen Bits möglich wäre, könnten wir diese Münze einmal werfen und das Ergebnis aufschreiben. Das entspricht einem probabilistischen Bit an Information, welches mit Wahrscheinlichkeit  $p$  gleich 0 ist. Wenn du diese Münze jetzt zugesendet bekommst und sie klonen kannst, könntest du dir beliebig viele Kopien erstellen und sie alle messen. Indem du dir die Messergebnisse anschaust und zählst, wie viele nullen du erhalten hast, kannst du die Wahrscheinlichkeit  $p$  abschätzen. Tatsächlich kannst du mit genügend Kopien die Wahrscheinlichkeit beliebig akkurat abschätzen! Insbesondere könntest du dir jedes einzelne Bit des Bitstrings der Wahrscheinlichkeit  $p$  anschauen, also auch all die kodierte Information, zum Beispiel das 65535te Katzenbild!

Offensichtlich sollte das in unserer Welt nicht funktionieren, da wir sonst keine USB-Sticks oder Datencenter bräuchten und nicht für unsere Mobilfunkverbindung zahlen müssten – wir könnten alle Information in einem einzelnen probabilistischen Bit speichern und sie übertragen indem wir dieses eine Bit versenden. Das ist auf jeden Fall zu schön um wahr zu sein!

#### 4.2.2 One-Time-Pad

Bevor wir über die Teleportation von Quantenzuständen sprechen, ist es hilfreich, zunächst das einfachere, *One-Time-Pad* genannte, Verfahren für probabilistische Bits zu verstehen. Dieses Protokoll erlaubt es Alice eine Nachricht so zu verschlüsseln und an Bob zu verschicken, dass nur Bob die Nachricht verstehen kann. Das bedeutet, sollte jemand anderes (z.B. die Klassenkame-

radin Eve) die verschlüsselte Nachricht abfangen, würde die Person keine Ahnung haben, was die tatsächliche Nachricht ist. Die Tatsache, dass ein solches Protokoll überhaupt möglich ist, ist ein wenig überraschend. Welchen Vorteil hat Bob gegenüber Eve, sodass er Alice's Nachricht korrekt entschlüsseln kann während Eve gar nichts über den Inhalt herausfinden kann?

Der Trick an der ganzen Sache ist, dass sich Alice und Bob zuvor in einem Café getroffen haben. Sie nehmen zwei Münzen, kleben sie mit einem Kaugummi zusammen und werfen die resultierende "Doppelmünze" nur um die Münzen anschließend wieder zu trennen. Beide nehmen dann jeweils eine der geworfenen Münzen. Jetzt teilen sie sich ein Paar zufälliger Bits, die sich in dem Zustand aus (3.5) befinden, also

$$r = \frac{1}{2}([00] + [11]).$$

Das kannst du dir jetzt als eine Art geteiltes Geheimnis vorstellen! Außerdem kennen nur Alice und Bob dieses Geheimnis – sie können einfach ihre Münze anschauen (also messen). Beide werden die gleiche Seite sehen, und die Wahrscheinlichkeit für jede der Seiten ist  $1/2$ . Dadurch ist das ein sehr gutes Geheimnis, da Eve den Wert nicht besser vorhersagen kann, als einfach zu raten!

Sehen wir doch mal, wie Alice und Bob dieses Geheimnis jetzt nutzen können. Nehmen wir mal an, Alice möchte die geheime Nachricht  $m \in \{0, 1\}$  an Bob senden. Dann ist der Gesamtzustand aller Bits durch den folgenden Zustand beschrieben

$$[m] \otimes r = \frac{1}{2}([m00] + [m11]), \quad (4.18)$$

wobei die ersten beiden Bits ( $m$  und das erste Bit von  $r$ ) Alice gehören und das dritte Bit (die zweite Bit von  $r$ ) zu Bob gehört.

Um ihre Nachricht zu senden, schaut sich Alice ihre Hälfte von  $r$  an und

1. wenn sie 0 sieht, sendet sie  $m$  direkt an Bob,
2. wenn sie 1 sieht, sendet sie  $\text{NOT}(m)$  an Bob.

Überlegen wir doch mal was passiert, wenn Eve diese Nachricht abfängt. Was sieht sie? Unabhängig vom Wert von  $m$  sieht sie 0 und 1 jeweils mit Wahrscheinlichkeit  $1/2$ . Das liegt daran, dass Alice  $m$  mit Wahrscheinlichkeit  $1/2$  invertiert, was die Nachricht so randomisiert, dass Eve sie als uniform zufällig verteiltes Bit sieht.

Aber was ist mit Bob? Wirkt Alice' Nachricht nicht auch auf ihn uniform zufällig verteilt? Zum Glück hat Bob die andere Hälfte des geteilten Geheimnis'  $r$ . Zwar wirkt die Nachricht auch auf ihn zu Beginn zufällig, er kann sie jedoch entschlüsseln, indem er genau dasselbe wie Alice macht: er schaut sich seine Hälfte von  $r$  an und

1. wenn er 0 sieht, nimmt er Alice' Nachricht, so wie sie ist,
2. wenn er 1 sieht, wendet er eine NOT-Operation auf die Nachricht an.

Insgesamt wird Alice' Nachricht also entweder direkt übertragen oder aber zweimal invertiert, was bedeutet, dass Bob sie in jedem Fall versteht. Aber aus Eves Sicht wird die Nachricht mit Wahrscheinlichkeit  $1/2$  invertiert, weshalb sie ein uniform zufälliges Bit sieht. Daher ist das eine perfekt sichere Möglichkeit mit der Alice und Bob kommunizieren können!

Machen wir uns doch ein genaueres Bild von dem, was hier passiert. Wenn Alice ihr erstes Bit invertiert, falls das zweite Bit 1 ist, dann wendet sie praktisch  $\text{CNOT}_{3 \rightarrow 1}$  auf ihre beiden Bits an. Anschließend sendet sie ihr erstes Bit an Bob. Bob entschlüsselt dann, indem er  $\text{CNOT}_{3 \rightarrow 1}$  anwendet (was er tun kann, da er neben dem dritten auch das erste Bit besitzt). Daraus resultiert dann der Zustand

$$\text{CNOT}_{3 \rightarrow 1} \text{CNOT}_{2 \rightarrow 1}([m] \otimes r).$$

Wir können diesen wie folgt ausrechnen:

$$\begin{aligned} \text{CNOT}_{3 \rightarrow 1} \text{CNOT}_{2 \rightarrow 1} \frac{1}{2}([m, 0, 0] + [m, 1, 1]) &= \text{CNOT}_{3 \rightarrow 1} \frac{1}{2}([m, 0, 0] + [\text{NOT}(m), 1, 1]) \\ &= \frac{1}{2}([m, 0, 0] + [\text{NOT}(\text{NOT}(m)), 1, 1]) = \frac{1}{2}([m, 0, 0] + [m, 1, 1]) = [m] \otimes r. \end{aligned}$$

Also ist das Nachrichten-Bit am Ende im Originalzustand, aber nun im Besitz von Bob.

Ein interessanter Aspekt des obigen One-Time-Pad-Protokolls ist, dass es Alice nicht nur erlaubt, eine deterministische Nachricht  $[m]$  an Bob zu senden, sondern sogar eine probabilistische Nachricht. Durch Linearität folgt, dass, wenn Alices Nachricht ein probabilistisches Bit mit der Verteilung  $p$  ist, der Anfangszustand  $p \otimes r$  und der Endzustand wiederum  $p \otimes r$  ist, wobei dieses Mal  $p$  in Bobs Besitz ist. Aus Eves Sicht ist die übermittelte Nachricht jedoch immer noch uniform zufällig. Das Erstaunliche daran ist, dass Alice durch das Senden eines uniform zufälligen Bits heimlich ein probabilistisches Bit übermitteln kann, dessen Verteilung sie vielleicht nicht einmal selbst kennt.

Dieses Verfahren ähnelt stark einer Quanten-Teleportation, bei der Alice ein Qubit im Zustand  $|\psi\rangle$  an Bob schicken kann, indem sie zwei (anstelle von einem) uniform zufällige Bits sendet. Für Teleportation müssen sich die beiden Parteien ein maximal verschränkter Zustand  $|\Phi^+\rangle$  anstelle eines zufälligen Bits  $r$  teilen. In beiden Fällen wird die geteilte Ressource während des Protokolls gemessen und dementsprechend verbraucht. Im folgenden Abschnitt findet ihr mehr Informationen dazu.

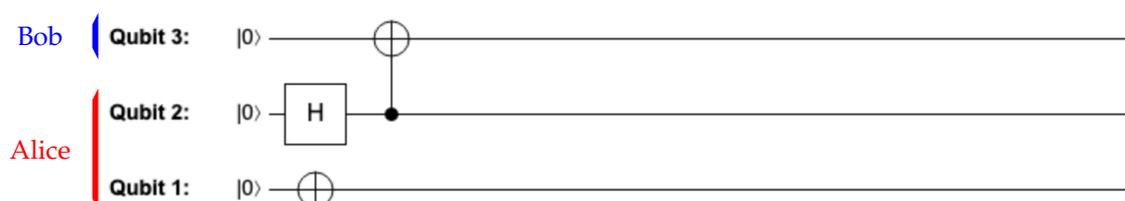
### 4.2.3 Quanten-Teleportation

Es mag zwar unmöglich sein, Qubits zu klonen, aber wir können sie auf jeden Fall von einer Stelle zur anderen *bewegen*. Tatsächlich ist jedes Qubit in einem physikalischen Objekt oder Teilchen gespeichert, welches das Qubit trägt. Wenn Alice ihr Qubit beispielsweise in der Polarisation eines Photons speichert, kann sie einfach dieses Photon an Bob senden. Überraschenderweise kann man allerdings ein Qubit auch bewegen, indem man eine begrenzte Anzahl an klassischen Bits sendet (tatsächlich genügen zwei Bits). Das ist besonders deswegen überraschend, weil der allgemeine Zustand eines Qubits  $|\psi(\theta)\rangle$  durch einen beliebigen Winkel  $\theta$  beschrieben wird, siehe Gl. (2.5), was im Allgemeinen nicht durch eine begrenzte Anzahl von Bits beschrieben werden kann. Aufgrund dieser unerwarteten Eigenschaft wird dieses Protokoll **Teleportation** genannt. Wir werden gleich sehen, dass wir Verschränkung benötigen, um das Protokoll zum Laufen zu bekommen.

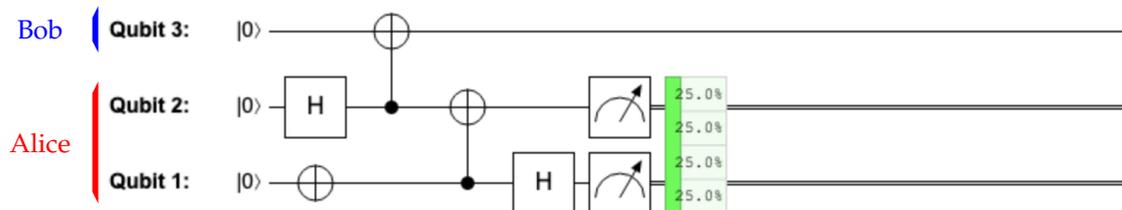
Ausgangspunkt für die Teleportation ist das folgende Szenario. Wir stellen uns vor, dass Alice zwei Qubits und Bob einen Qubit hat. Das erste Qubit von Alice ist das Nachrichten-Qubit, das sie an Bob senden möchte, und befindet sich zu Beginn in einem beliebigen Zustand  $|\psi\rangle$ , der Alice selbst nicht bekannt sein muss! Ihr zweites Qubit und Bobs Qubit befinden sich in einem maximal verschränkten Zustand  $|\Phi^+\rangle$ . Die drei Qubits befinden sich also in folgendem Zustand:

$$|\psi\rangle \otimes |\Phi^+\rangle,$$

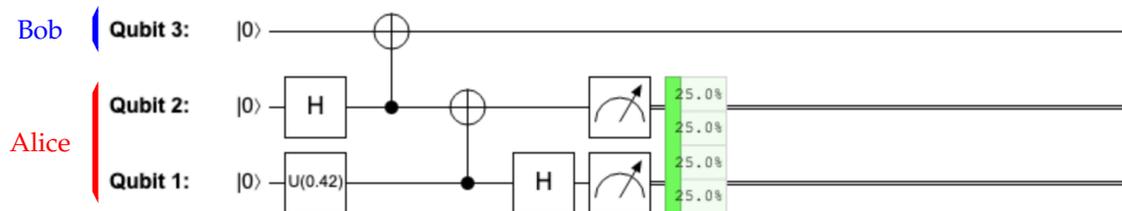
wobei die ersten beiden Qubits zu Alice und das letzte Qubit zu Bob gehört (wie aus Gl. (3.52) hervorgeht, ist  $|\Phi^+\rangle$  ein Zwei-Qubit-Zustand). Die folgende QUIRKY-Schaltung zeigt, wie dies für den Fall aussieht, dass Alice ein Qubit im Zustand  $|\psi\rangle = |1\rangle = \text{NOT}|0\rangle$  senden will:



Wie soll es also weitergehen? Das Ziel ist es, dass Bob das Qubit von Alice erhält - da wir das Qubit nicht klonen können, bedeutet dies, dass Alice etwas machen muss, dass ihr Qubit "zerstört". Eine gute Option dafür, ist eine Messung. Aber eine einfache Messung von Alices beiden Qubits wird den beiden nichts nützen, da man den Zustand von  $|\psi\rangle$  nicht aus einer einzigen Messung ableiten kann. Das bedeutet, dass Alice zuerst eine Quantenoperation auf ihre beiden Qubits anwenden und sie dann messen sollte. Es stellt sich heraus, dass die richtige Strategie für sie darin besteht, dieselben Operationen durchzuführen, die Sie zur Unterscheidung der vier Bell-Zustände in Übungsaufgabe 3.13 verwendet haben:



Bemerke, dass in diesem Beispiel Alice' vier Messergebnisse jeweils mit 25% auftreten. Hier ist ein weiteres Beispiel, in dem Alice stattdessen den Zustand  $|\psi(0.42)\rangle$  teleportieren will:



Es scheint, egal welchen Zustand Alice' Nachrichten-Qubit hat, die vier Messwahrscheinlichkeiten bleiben gleich. Das ist ermutigend, schließlich bedeutet das, dass die Messung Alice keine Information über ihr Nachrichten-Qubit gibt! Wenn wir uns daran erinnern, dass ihr ursprünglicher Zustand  $|\psi\rangle$  komplett bei Bob landen soll, ist es klar, dass sie keine Informationen über diesen Zustand extrahieren oder behalten sollte.

Im allgemeinen sieht der Zustand direkt vor Alice' Messung wie folgt aus:

$$(H \otimes I \otimes I) (\text{CNOT}_{1 \rightarrow 2} \otimes I) (|\psi\rangle \otimes |\Phi^+\rangle)$$

Beachte, dass die letzten zwei Tensorprodukte *nicht* korrekt ausgerichtet sind (siehe Übungsaufgabe 4.3). Daher berechnen wir:

$$\begin{aligned} & (H \otimes I \otimes I) (\text{CNOT}_{1 \rightarrow 2} \otimes I) (|\psi\rangle \otimes |\Phi^+\rangle) \\ &= \frac{1}{\sqrt{2}} (H \otimes I \otimes I) (\text{CNOT}_{1 \rightarrow 2} \otimes I) (\psi_0 |000\rangle + \psi_0 |011\rangle + \psi_1 |100\rangle + \psi_1 |111\rangle) \\ &= \frac{1}{\sqrt{2}} (H \otimes I \otimes I) (\psi_0 |000\rangle + \psi_0 |011\rangle + \psi_1 |110\rangle + \psi_1 |101\rangle) \\ &= \frac{1}{2} (\psi_0 |000\rangle + \psi_0 |100\rangle + \psi_0 |011\rangle + \psi_0 |111\rangle + \psi_1 |010\rangle - \psi_1 |110\rangle + \psi_1 |001\rangle - \psi_1 |101\rangle) \\ &= |00\rangle \otimes \frac{\psi_0 |0\rangle + \psi_1 |1\rangle}{2} + |01\rangle \otimes \frac{\psi_1 |0\rangle + \psi_0 |1\rangle}{2} \\ &+ |10\rangle \otimes \frac{\psi_0 |0\rangle - \psi_1 |1\rangle}{2} + |11\rangle \otimes \frac{-\psi_1 |0\rangle + \psi_0 |1\rangle}{2}. \end{aligned}$$

Das ist der Gesamtzustand direkt vor Alice' Messung. Wir können die Wahrscheinlichkeiten ihrer Messergebnisse wie in Gl. (4.14) besprochen berechnen. Genauer gesagt, um die

Wahrscheinlichkeit  $p_{a,b}$  des Ergebnisses  $[ab]$  zu berechnen, addieren wir einfach die quadrierten Amplituden der relevanten Basiszustände  $|ab0\rangle$  und  $|ab1\rangle$ . In jedem Fall ist eine der Amplituden  $\psi_0/2$  und die andere  $\pm\psi_1/2$ , also summieren sich die Quadrate zum gleichen Ergebnis:

$$\begin{aligned}
 p_{00} &= \left(\frac{\psi_0}{2}\right)^2 + \left(\frac{\psi_1}{2}\right)^2 = \frac{\psi_0^2 + \psi_1^2}{4} = \frac{1}{4}, \\
 p_{01} &= \left(\frac{\psi_1}{2}\right)^2 + \left(\frac{\psi_0}{2}\right)^2 = \frac{1}{4}, \\
 p_{10} &= \left(\frac{\psi_0}{2}\right)^2 + \left(\frac{-\psi_1}{2}\right)^2 = \frac{1}{4}, \\
 p_{11} &= \left(\frac{-\psi_1}{2}\right)^2 + \left(\frac{\psi_0}{2}\right)^2 = \frac{1}{4}.
 \end{aligned}$$

Jedes Ergebnis tritt mit 25% auf. Das bestätigt das Ergebnis aus QUIRKY.

Nach Alice' Messung bleibt nur noch Bobs Qubit übrig. Wir bezeichnen dessen Zustand als  $|\psi'_{a,b}\rangle$ , da es von Alice' Messergebnis abhängt. Den Zustand können wir mit Gl. (4.15) oder viel einfacher mit der Gruppierungs-Methode aus Gl. (4.13) bestimmen. Beide Fälle führen zu den folgenden vier möglichen Zuständen:

$$\begin{aligned}
 |\psi'_{00}\rangle &= \psi_0 |0\rangle + \psi_1 |1\rangle, \\
 |\psi'_{01}\rangle &= \psi_1 |0\rangle + \psi_0 |1\rangle, \\
 |\psi'_{10}\rangle &= \psi_0 |0\rangle - \psi_1 |1\rangle, \\
 |\psi'_{11}\rangle &= -\psi_1 |0\rangle + \psi_0 |1\rangle.
 \end{aligned}$$

Falls  $[ab] = [00]$ , ist Bobs Zustand genau der gleiche wie Alice' ursprünglicher Zustand  $|\psi\rangle$ , welchen sie teleportieren wollte:

$$|\psi'_{00}\rangle = |\psi\rangle.$$

In den anderen drei Fällen ist Bobs Quantenzustand ein bisschen verzerrt. Aber da Alice ihm ihre Messergebnisse sendet (also die zwei Bits  $a$  und  $b$ ), kann Bob die notwendigen Korrekturen anwenden um den Zustand zu "retten":

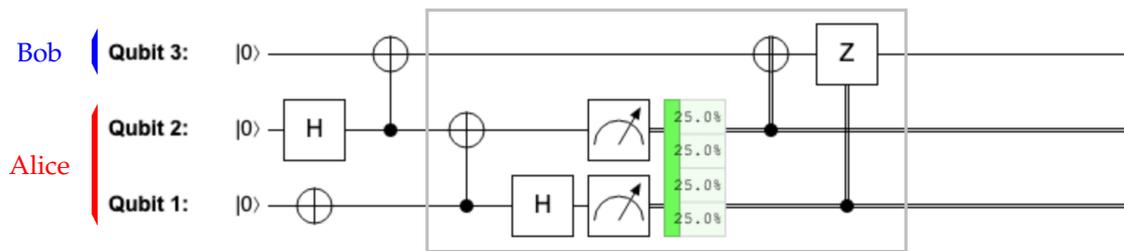
$$\begin{aligned}
 \text{NOT } |\psi'_{01}\rangle &= |\psi\rangle, \\
 Z |\psi'_{10}\rangle &= |\psi\rangle, \\
 Z \text{NOT } |\psi'_{11}\rangle &= |\psi\rangle.
 \end{aligned}$$

Diese vier Fälle können einfach zusammengefasst werden:

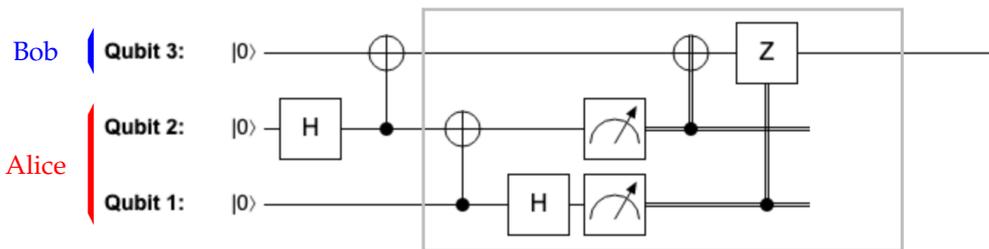
1. schaue das Bit  $b$  an und wenn  $b = 1$ , wende NOT an,
2. schaue das Bit  $a$  an und wenn  $a = 1$ , wende Z an.

Wir können diese Prozedur mit einem kontrolliertem-NOT- und einem kontrolliertem-Z-Gatter implementieren, wobei die Kontroll-Bits klassisch sind. Die kontrollierte-Z-Operation kann man genau wie die kontrollierte-NOT-Operation bauen – das haben wir am Ende von §3.2.4 besprochen. Puh, das war ein ganzes Stück Arbeit!

Wie sieht das ganze denn in QUIRKY aus? Das Endergebnis siehst du in der folgenden Quantenschaltung:

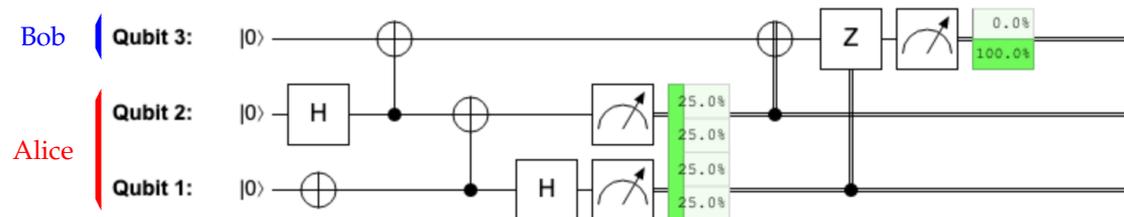


Der relevante Teil ist in der grauen Box – der **Teleportations Schaltkreis** – um ihn vom restlichen Teil abzugrenzen, der die Eingabezustände erstellt. Hier siehst du eine Quantenschaltung, die nur die Teleportation und das erstellen des maximal verschränkten Zustands zeigt, ohne Alice' Nachrichten-Qubit zu spezifizieren:



Außerdem haben wir die Drähte von Alice' zwei klassischen Bits abgeschnitten, da sie nicht mehr relevant sind, nachdem Alice die Messergebnisse an Bob geschickt hat. Die Wahrscheinlichkeitsanzeige wurde auch entfernt, da sie keine tatsächliche Quantenoperation ist, sondern uns nur erlaubt, den Schaltkreis in QUIRKY genauer anzuschauen. Es gibt also nur ein Eingabe-Bit für Alice' Nachricht, zwei maximal verschränkte Qubits und ein Ausgabe Qubit auf Bobs Seite. Die Teleportation hat also den Effekt einen beliebigen Quantenzustand  $|\psi\rangle$  von Alice' Eingabe-Qubit auf Bobs Seite zu transportieren. Entscheidend ist dabei, dass innerhalb der Box nur klassische Bits von Alice zu Bob geschickt werden!

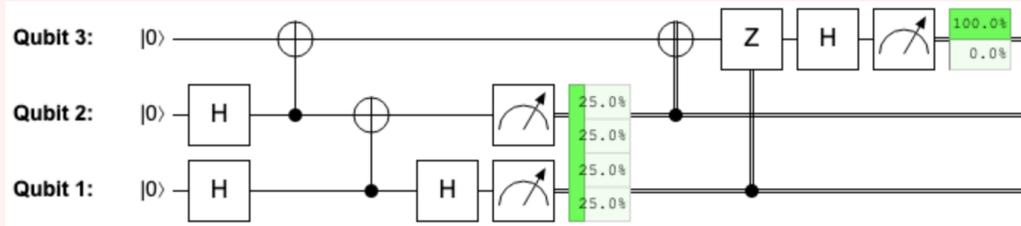
Lass uns noch einmal überprüfen, dass wir keine Fehler gemacht haben. Da wir erwarten, dass Bobs Qubit nach der Teleportation im Zustand  $|1\rangle$  sein sollte, können wir einfach eine Messung hinzufügen, um zu prüfen ob das tatsächlich der Fall ist:



Tatsächlich erhalten wir das Ergebnis 1 mit 100%, was bestätigt, dass die Teleportation des  $|1\rangle$  Zustands erfolgreich war! Nun ist  $|1\rangle$  aber kein besonders interessanter Zustand, den wir teleportieren können. Wie wäre es denn mit dem  $|+\rangle$  Zustand, der uns beim Klonen so Probleme bereitet hat? In der folgenden Hausaufgabe wirst du den Teleportations-Schaltkreis mit dem  $|+\rangle$  Zustand und anschließend für beliebige Ein-Qubit-Zustände verifizieren.

### Hausaufgabe 4.2: Teleportation testen

1. Warum teleportiert der folgende Schaltkreis den  $|+\rangle$  Zustand korrekt?



2. Wie kannst du auf ähnliche Weise prüfen, ob ein beliebiger Quantenzustand  $|\psi\rangle$  korrekt teleportiert wird?

Die Teleportationsschaltung ist recht bemerkenswert. Sie ermöglicht es, ein Quantenbit von Alice zu Bob zu senden, indem nur zwei klassische Bits übertragen werden, vorausgesetzt, dass Alice und Bob einen maximal verschränkten Zustand teilen. Sie ist jedoch nicht nur für Anwendungen nützlich (wir werden einige davon weiter unten besprechen), sondern bietet auch eine interessante Perspektive auf den Unterschied zwischen klassischen und Quantenbits. Ende letzter Woche haben wir über das Superdense-Coding-Protokoll gesprochen, welches es uns ermöglicht, zwei Bits zu senden, indem wir ein einzelnes Qubit übertragen, wobei wir auch wieder einen maximal verschränkten Zustand zwischen Alice und Bob genutzt haben. Das zeigt, dass:

*Bei unbegrenzt vorhandener Quantenverschränkung ist das Senden  
von zwei Bits gleichwertig zum Senden eines Quantenbits!*

Beachte, dass wir weder bei Teleportation noch bei Superdense-Coding die maximal verschränkten Zustände weiterverwenden können – sie sind der “Kraftstoff”, der beim Prozess verbraucht wird.

#### 4.2.4 Ein Blick auf Quantennetzwerke

Durch wiederholte Quantenteleportation können wir Quantenbits zwischen weit entfernten Knotenpunkten kommunizieren. Nehmen wir beispielsweise an, Alice, ihr Roboteresel und Bob befinden sich in der folgenden Situation:

$$\text{Alice' Roboteresel} \longleftrightarrow \text{Alice} \longleftrightarrow \text{Bob},$$

wobei jeder “ $\leftrightarrow$ ”-Pfeil einen maximal verschränkten Zustand darstellt. Dann ist der Gesamtzustand der drei Teilnehmenden in dem folgenden Vier-Qubit-Zustand:

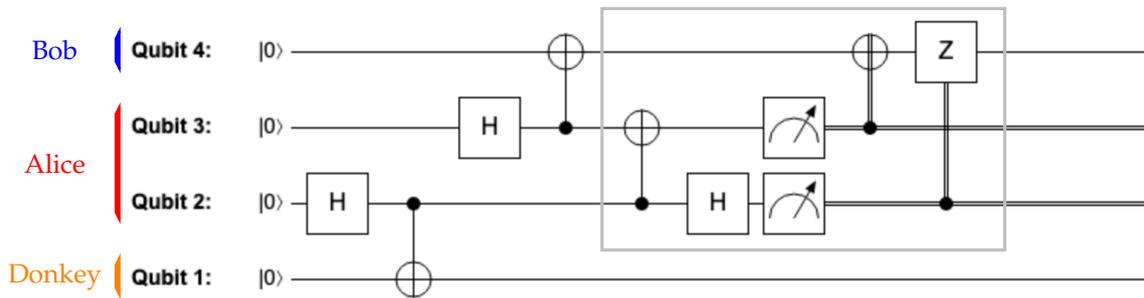
$$|\Phi^+\rangle \otimes |\Phi^+\rangle,$$

Wobei die mittleren beiden Bits Alice gehören – das erste davon ist mit dem Roboter verschränkt, während das zweite mit Bob verschränkt ist.

Dabei ist der Roboteresel nicht direkt mit Bob verschränkt! Trotzdem könnte der Roboter eine Quantennachricht an Bob schicken, indem die Teleportation zweimal durchgeführt wird: zunächst teleportieren wir die Nachricht vom Roboteresel zu Alice (wobei der erste verschränkte Zustand verbraucht wird) und anschließend von Alice zu Bob (sodass auch der zweite verschränkte Zustand verbraucht wird). Das entspricht in etwa einem Mobiltelefon, dass sich zunächst mit einer nahegelegenen Basisstation verbindet, welche das Signal wiederum an einen weiteren Masten weiterleitet (und so weiter). Zwar können wir Qubit wegen des Unklonbarkeit-Theorems (siehe Hausaufgabe 4.1) nicht kopieren, aber trotzdem können wir es über große Distanzen teleportieren!

Verschränkung ist nicht nur für Teleportation sondern auch für viele anderen Dinge nützlich. Könnten wir mit **Teleportation auch Verschränkung** zwischen Alice' Roboteresel und Bob **herstellen** (die diese dann für andere Zwecke nutzen können)?

Intuitiv scheint es, als müsste Alice nur ihr erstes Qubit (welches mit dem Roboteresel verschränkt ist) zu Bob teleportieren. In QUIRKY würde das so aussehen:



Dabei werden zuerst zwei maximal verschränkte Zustände erstellt und anschließend der schon bekannte Teleportations-Schaltkreis (graue Box) angewendet. Wir hoffen dabei, dass das zu einem maximal verschränkten Zustand zwischen dem Roboteresel und Bob führt. In der folgenden Hausaufgabe kannst du beweisen, dass das tatsächlich der Fall ist.

### Hausaufgabe 4.3: Teleportieren von Verschränkung

Bestätige mit QUIRKY, dass am Ende des Schaltkreises die Qubits vom Roboteresel und von Bob tatsächlich im maximal verschränkten Zustand  $|\Phi^+\rangle$  sind.

Auf die gleiche Art können wir Teleportation nutzen um Verschränkung zwischen immer weiter entfernten Knotenpunkten zu erstellen. Stell dir beispielsweise vor, wir haben folgende Situation:

$$\text{Alice' Roboteresel} \longleftrightarrow \text{Alice} \longleftrightarrow \text{Bob} \longleftrightarrow \text{Bobs Roboter-Eichhörnchen.}$$

Wenn Alice zunächst ihr erstes Qubit zu Bob teleportiert und dieser anschließend sein erstes Qubit zu seinem Roboter-Eichhörnchen teleportiert, erhalten wir einen maximal verschränkten Zustand zwischen den beiden Robotern. Hoffen wir, dass die Roboter die Verschränkung nur für gute Zwecke nutzen!

Verschränkung über große Distanzen herzustellen wird dann wichtig, wenn wir versuchen wollen, Quantencomputer in einem kleinen Netzwerk zu verbinden, oder in einer kühn- geträumten Zukunft für ein "Quanten-Internet". Einige von uns denken schon intensiv darüber nach, wie sich sowas in der Praxis umsetzen ließe und wofür wir diese Verschränkung über große Entfernung am besten nutzen können.

### 4.2.5 Die Unschärferelation

Für das letzte Phänomen, das wir besprechen wollen, benötigen wir nur ein einzelnes Qubit. Erinnern wir uns an die Regeln für die Messung eines einzelnen Qubits aus Gl. (2.6), die im folgenden Bild dargestellt sind:



Gegeben einen Quantenzustand  $|\psi\rangle = \psi_0 |0\rangle + \psi_1 |1\rangle$  erhalten wir die zwei möglichen Ergebnisse mit den Wahrscheinlichkeiten

$$p_0 = \psi_0^2, \quad p_1 = \psi_1^2. \quad (4.19)$$

Welches sind die deterministischen Zustände, bei denen wir eines der Ergebnisse mit Sicherheit erhalten? Das sind Zustände, bei denen die eine Wahrscheinlichkeit 100% und die andere 0%

beträgt. Mit anderen Worten: Zustände, bei denen die eine Amplitude  $\pm 1$  und die andere Null ist. Bis auf ein globales Vorzeichen, das, wie wir aus Übungsaufgabe 2.7 wissen, irrelevant ist, gibt es nur zwei solcher Zustände:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (4.20)$$

also die Basiszustände. Das sind auch die einzigen beiden Zustände, bei denen wir das Messergebnis perfekt vorhersagen können, also sagen wir, dass es keine **Ungewissheit** oder **Unschärfe** (engl. *uncertainty*) in dem Messergebnis gibt.

Was passiert, wenn wir zunächst eine Operation auf dem Qubit ausführen und anschließend messen? Beispielsweise könnten wir zuerst die Hadamard-Operation anwenden und dann messen, wie in diesem Bild:



Hier sind die einzigen Zustände, bei denen wir uns sicher über das Messergebnis sind

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad |-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad (4.21)$$

(bis auf ein globales Vorzeichen). Das liegt daran, dass die Hadamard-Operationen  $|+\rangle, |-\rangle$  zurück auf die Basiszustände  $|0\rangle, |1\rangle$  abbildet (das hast du in Übungsaufgabe 4.5 gezeigt); die letzten beiden sind dabei genau die Zustände, bei denen wir endgültige Gewissheit über das Messergebnis haben, wie oben besprochen. Wir können das auch überprüfen, indem wir die Wahrscheinlichkeiten der Messergebnisse berechnen

$$q_0 = \frac{(\psi_0 + \psi_1)^2}{2}, \quad q_1 = \frac{(\psi_0 - \psi_1)^2}{2}. \quad (4.22)$$

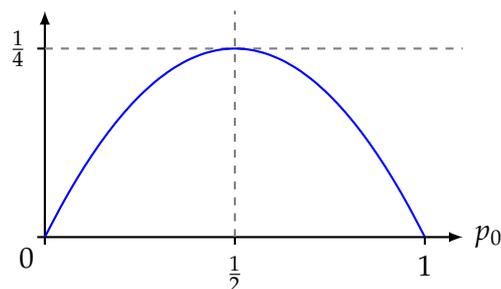
Wenn  $q_1 = 0$  ist, dann ist der Zustand  $|+\rangle$ , während für  $q_0 = 0$  sich der Zustand  $|-\rangle$  ergibt (bis auf globale Vorzeichen).

Das sind alles Berechnungen, die wir schon mehrfach gesehen haben – allerdings gibt es eine interessante Beobachtung, die wir bisher nicht gemacht haben. Da keiner der Zustände *sowohl* in Gl. (4.20) *als auch* in Gl. (4.21) vorkommt, muss für jeden Zustand eine der beiden Messvorgänge Ungewissheit im Messergebnis haben. Dieses Ergebnis ist genau die bekannte *Heisenbergsche Unschärferelation!*

Können wir diese Beobachtung quantitativer machen? Zunächst brauchen wir eine Möglichkeit, die Ungewissheit oder “Zufälligkeit” zu quantifizieren, die durch eine Wahrscheinlichkeitsverteilung  $p = \begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$  gegeben ist. Die folgende Funktion ist eine gute Wahl:

$$p(p) = p_0(1 - p_0) = p_0 p_1.$$

Sie bildet die Werte auf  $[0, 1/4]$  ab und ist minimal wenn eines der beiden Messergebnisse mit Wahrscheinlichkeit 0 auftritt (also wenn  $p_0 = 0$  oder  $p_1 = 0$ ) und ist genau dann maximal, wenn beide Ergebnisse gleich wahrscheinlich sind (also  $p_0 = p_1 = 1/2$ ). Hier siehst du die Funktion grafisch:



Nehmen wir jetzt an, dass wir mit dem Zustand  $|\psi\rangle$  starten und eine der beiden Messungen von oben vornehmen. Soll heißen, wir messen den Zustand entweder direkt, oder wir wenden vorher die Hadamard-Operation an und messen anschließend. Die entsprechenden Ungewissheiten sind  $(p)$  und  $(q)$  wobei die Verteilungen  $p = \begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$  und  $q = \begin{pmatrix} q_0 \\ q_1 \end{pmatrix}$  in Gl. (4.19) und (4.22) gegeben sind. Dann gilt:

$$(p) + (q) > 0.$$

Tatsächlich bedeutet diese Ungleichung genau, dass es keinen Zustand gibt, für den beiden Ungewissheiten gleichzeitig null sind. In der folgenden Hausaufgabe wirst du eine genauere Abschätzung zeigen.

#### Hausaufgabe 4.4: Das Gleichgewicht der Ungewissheit

Zeige, dass für jeden Quantenzustand  $|\psi\rangle$  gilt:

$$(p) + (q) = \frac{1}{4}. \quad (4.23)$$

Finde weiterhin einen Quantenzustand  $|\psi\rangle$  sodass  $(p) = (q)$ .

*Bonusaufgabe:* Erstelle diesen Zustand mit QUIRKY und zeige mit QUIRKY, dass tatsächlich  $(p) = (q)$ .

Die Gleichung, die du gerade in Hausaufgabe 4.4 bewiesen hast ist sehr bemerkenswert: sie zeigt, dass es ein einfaches Tradeoff zwischen den Ungewissheiten der beiden Messungen gibt. Genauer gesagt, wenn eine Messung eine Ungewissheit von null hat, ist das Ergebnis der anderen uniform zufällig!<sup>12</sup>

<sup>12</sup> Das können wir auch direkt sehen. Wenn wir z.B. den Zustand  $|+\rangle$  (ein Zustand, der bei der zweiten Messung keine Ungewissheit hat) messen, ohne vorher eine Hadamard-Operation anzuwenden, erhalten wir die Ergebnisse 0 und 1 mit gleicher Wahrscheinlichkeit.

### 4.3 Lösungen der Übungsaufgaben

#### Lösung Übungsaufgabe 4.1

$$\begin{aligned} & |\Phi^-\rangle \otimes |\Psi^-\rangle \\ &= \left( \frac{1}{\sqrt{2}} |00\rangle - \frac{1}{\sqrt{2}} |11\rangle \right) \otimes \left( \frac{1}{\sqrt{2}} |01\rangle - \frac{1}{\sqrt{2}} |10\rangle \right) \\ &= \frac{1}{2} |0001\rangle - \frac{1}{2} |0010\rangle - \frac{1}{2} |1101\rangle + \frac{1}{2} |1110\rangle. \end{aligned}$$

#### Lösung Übungsaufgabe 4.2

Lass uns zunächst den Produktzustand in die Form von Gl. (4.2) erweitern:

$$|\Phi^+\rangle \otimes |1\rangle = \frac{1}{\sqrt{2}} |001\rangle + \frac{1}{\sqrt{2}} |111\rangle.$$

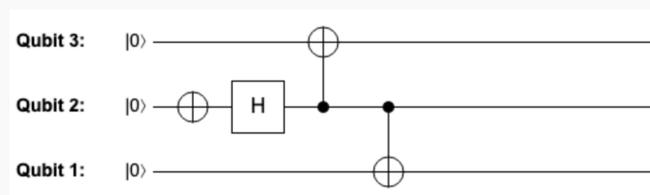
Daraus folgt:

$$\begin{aligned} H_2(|\Phi^+\rangle \otimes |1\rangle) &= H_2\left(\frac{1}{\sqrt{2}} |001\rangle + \frac{1}{\sqrt{2}} |111\rangle\right) = \frac{1}{\sqrt{2}} H_2 |001\rangle + \frac{1}{\sqrt{2}} H_2 |111\rangle \\ &= \frac{1}{\sqrt{2}} |0\rangle \otimes H|0\rangle \otimes |1\rangle + \frac{1}{\sqrt{2}} |1\rangle \otimes H|1\rangle \otimes |1\rangle \\ &= \frac{1}{\sqrt{2}} |0\rangle \otimes |+\rangle \otimes |1\rangle + \frac{1}{\sqrt{2}} |1\rangle \otimes |-\rangle \otimes |1\rangle \\ &= \frac{1}{\sqrt{2}} |0\rangle \otimes \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle\right) \otimes |1\rangle + \frac{1}{\sqrt{2}} |1\rangle \otimes \left(\frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle\right) \otimes |1\rangle \\ &= \frac{1}{2} |001\rangle + \frac{1}{2} |011\rangle + \frac{1}{2} |101\rangle - \frac{1}{2} |111\rangle, \end{aligned}$$

wobei wir Gl. (2.20) genutzt haben, um das  $H$  auf die Basiszustände anzuwenden

#### Lösung Übungsaufgabe 4.3

- In Übungsaufgabe 3.12 haben wir gesehen, wie man  $|\Phi^-\rangle$  herstellt. Daher funktioniert der folgende Schaltkreis:



- Hier ist der resultierende Zustand:

$$\begin{aligned} (\text{CNOT}_{2 \rightarrow 1} \otimes I)(|0\rangle \otimes |\Phi^-\rangle) &= (\text{CNOT}_{2 \rightarrow 1} \otimes I) \left( \frac{1}{\sqrt{2}} |000\rangle - \frac{1}{\sqrt{2}} |011\rangle \right) \\ &= \frac{1}{\sqrt{2}} |000\rangle - \frac{1}{\sqrt{2}} |111\rangle. \end{aligned}$$

#### Lösung Übungsaufgabe 4.4

Sei

$$|\psi\rangle = \psi_{000}|000\rangle + \psi_{001}|001\rangle + \psi_{010}|010\rangle + \psi_{011}|011\rangle \\ + \psi_{100}|100\rangle + \psi_{101}|101\rangle + \psi_{110}|110\rangle + \psi_{111}|111\rangle$$

ein beliebiger Drei-Qubit-Zustand. Das Ergebnis einer Toffoli-Operation ist dann

$$|\psi'\rangle = T|\psi\rangle = \psi_{000}|000\rangle + \psi_{001}|001\rangle + \psi_{010}|010\rangle + \psi_{011}|011\rangle \\ + \psi_{100}|100\rangle + \psi_{101}|101\rangle + \psi_{110}|111\rangle + \psi_{111}|110\rangle.$$

Die zwei Basiszustände sind fett hervorgehoben. Bemerke, dass sich einzig die Amplituden von  $|110\rangle$  und  $|111\rangle$  vertauscht haben. Demnach gilt wenn  $\sum_{a,b,c=0}^1 \psi_{a,b,c}^2 = 1$ , dass auch  $\sum_{a,b,c=0}^1 (\psi'_{a,b,c})^2 = 1$ . Also bildet  $T$  Quantenzustände auf Quantenzustände ab.

#### Lösung Übungsaufgabe 4.5

1.

$$H|+\rangle = H\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{2}((|0\rangle + |1\rangle) + (|0\rangle - |1\rangle)) = |0\rangle,$$

$$H|-\rangle = H\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{2}((|0\rangle + |1\rangle) - (|0\rangle - |1\rangle)) = |1\rangle.$$

2. Um diese Identität zu zeigen, müssen wir nur prüfen, ob  $HZH$  sich auf den Basiszuständen genauso wie NOT verhält (durch Linearität würde das bedeuten, dass sie sich auf allen Quantenzuständen gleich verhalten):

$$HZH|0\rangle = HZ\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = H\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = H|-\rangle = |1\rangle,$$

$$HZH|1\rangle = HZ\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = H\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = H|+\rangle = |0\rangle.$$

In beiden Fällen invertiert  $HZH$  das Bit, also ist es die selbe Operation wie NOT.

3. Der erste Aufgabenteil zeigt, dass eine zweifache Anwendung des Hadamard-Gatters nichts ändert:  $HH = I$ . Also gilt,

$$Z = (HH)Z(HH) = H(HZH)H = H \text{ NOT } H$$

wobei wir im letzten Schritt das Ergebnis aus Aufgabe 2 verwendet haben.

### Lösung Übungsaufgabe 4.6

1. Aus Gl. (2.15) folgt

$$U(\theta_2)U(\theta_1) |\psi(\alpha)\rangle = U(\theta_2) |\psi(\alpha + \theta_1)\rangle = |\psi(\alpha + \theta_1 + \theta_2)\rangle = U(\theta_1 + \theta_2) |\psi(\alpha)\rangle .$$

Das gilt für jeden Quantenzustand  $|\psi(\alpha)\rangle$ , woraus folgt, dass  $U(\theta_2)U(\theta_1) = U(\theta)$  wobei  $\theta = \theta_1 + \theta_2$ . Auch geometrisch ist das logisch: Wenn man zuerst um den Winkel  $\theta_1$  und anschließend um den Winkel  $\theta_2$  rotiert, ergibt das insgesamt eine Rotation um den Winkel  $\theta = \theta_1 + \theta_2$ .

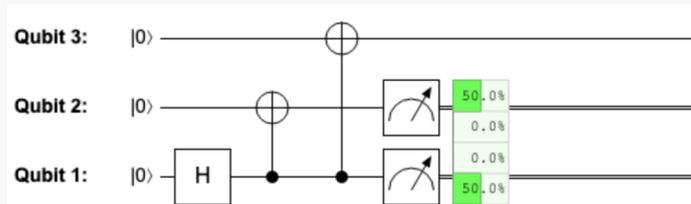
2. Ein eleganter Beweis könnte so aussehen. Aus Gl. (2.19) wissen wir, dass wir eine allgemeine Spiegelung auf zwei unterschiedliche Weisen ausdrücken können:  $V(\theta) = \text{NOT}U(\theta) = U(-\theta)\text{NOT}$ . Wenn man für  $V(\theta_1)$  die erste und für  $V(\theta_2)$  die zweite Form nimmt, erhält man

$$V(\theta_2)V(\theta_1) = U(-\theta_2)\text{NOTNOT}U(\theta_1) = U(-\theta_2)U(\theta_1) = U(\theta_1 - \theta_2),$$

wobei wir ausgenutzt haben, dass sich zwei aufeinanderfolgende NOT-Operationen aufheben und zwei aufeinanderfolgende Drehungen einer einzelnen um die Summe der Winkel entspricht, wie wir oben gezeigt haben.

### Lösung Übungsaufgabe 4.7

Nach Gl. (4.14) sollten wir  $[00]$  und  $[11]$  jeweils mit 50% Wahrscheinlichkeit erhalten. Tatsächlich:





## Quest 5: Virtuose Algorithmen

Einer der Hauptgründe, warum wir die Quanteninformatik erforschen, ist, dass Quantencomputer manche Probleme viel schneller lösen können, als alle uns aktuell zur Verfügung stehenden Computer. Um Quantencomputer von normalen Computern zu unterscheiden, verwenden wir den Begriff **klassischer Computer** für all jene Computer, die wir zurzeit haben. Das beinhaltet deinen Laptop oder PC, sehr kleine Computer wie dein Smartphone oder eine Smartwatch sowie auch sehr große und leistungsstarke *Supercomputer* die so groß wie ganze Räume sein können. Was all diese Computer von Quantencomputern unterscheidet ist nicht ihre Größe oder Rechengeschwindigkeit, sondern dass ihre internen Funktionen durch **klassische Physik** (genauer gesagt Elektromagnetismus) beschrieben werden können. Ihre Hardware funktioniert also auf eine Art und Weise, die mit alten physikalischen Theorien beschrieben werden, die es schon viel länger als die Quantenphysik gibt. Das ist in etwa so, wie die von Mozart und Bach komponierte Musik als *klassische Musik* bezeichnet wird. Genau wie bei klassischen Computern werden auch in der klassischen Musik keine modernen Musikinstrumente wie E-Gitarren oder Synthesizer benutzt.

Aufgrund der Hardware, die klassische Computer nutzen, werden alle Informationen die sie speichern – sei es ein Bild, eine Tondatei, ein Video oder eine Website – durch Bitstrings, also eine Folge von Nullen und Einsen, dargestellt. Diese Informationen werden dann entsprechend einiger Regeln so verarbeitet, dass das Ergebnis eine sinnvolle Antwort darstellt. Wir nennen diese Folge von Befehlen einen **Algorithmus**. Du kannst dir einen Algorithmus vorstellen wie ein Rezept – wenn du den enthaltenen Anweisungen genau folgst, erhältst du das gewünschte Ergebnis, zum Beispiel Schokoladen-Brownies! Ein Algorithmus könnte zum Beispiel zwei Bitstrings als Eingabe nehmen und einen anderen Bitstring ausgeben, der die Summe der ersten beiden Bitstrings (als Zahlen interpretiert) enthält. Genau wie Rezepte wurden auch Algorithmen ursprünglich von Menschen ausgeführt. Tatsächlich bezeichnet das Wort Computer (engl. für "Rechner:in") ursprünglich eine Person, die rechnet. Heutzutage werden Algorithmen aber auf echten Computern ausgeführt. Da Computer selbst nicht besonders intelligent sind, müssen diese Algorithmen ihnen sehr genau beschrieben werden. Diese Beschreibung, ein **Programm**, ist eine konkrete Umsetzung (Implementierung) des abstrakten Algorithmus, sodass der Computer sie verstehen kann. Dafür müssen wir eine **Programmiersprache** benutzen, die der Computer selbst in Operationen auf Nullen und Einsen übersetzen kann, um sie dann anschließend auf der Hardware auszuführen.

Wenn du einen Algorithmus entwirfst, ihn in deinen Computer einprogrammierst und ausführst, möchtest du die Antwort in einer angemessenen Zeit erhalten. Wie lange es aber tatsächlich braucht, um die Lösung zu berechnen, hängt von vielen Faktoren ab:

1. wie schnell dein Computer elementare Operationen ausführen kann,
2. ob die Eingabe deines Programms von der Festplatte, aus einer Netzwerkverbindung oder direkt vom Speicher ausgelesen wird,
3. welche Programmiersprache du für dein Programm genutzt hast (und die Version des Compilers oder Interpreters, der das Programm ausführt),

und so weiter. Das macht es sehr schwer, verschiedene Algorithmen zu vergleichen.

Um verschiedene Algorithmen besser vergleichen zu können, betrachten Informatiker:innen nicht, wie lange das Ausführen auf einem bestimmten Computer dauert. Stattdessen zählen sie, wieviele elementare Operationen der Algorithmus ausführt. So können sie sichergehen, dass sie die Algorithmen selbst vergleichen und nicht die Computer, auf denen die Algorithmen laufen (sonst könnte ein guter Algorithmus auf einem langsamen Computer schlechter aussehen, als ein schlechter Algorithmus auf einem schnellen Computer). Genauer gesagt interessiert

Informatiker:innen wie die Anzahl der Operationen mit der *Größe* des Problems, das der Algorithmus lösen soll, wächst. Denn je mehr Daten verarbeitet werden müssen, desto länger braucht der Algorithmus, egal wie gut er ist. Man will also wissen, ob der Algorithmus in der Lage ist, mit sehr großen Datenmengen umzugehen. Dieser Forschungsbereich wird auch **Komplexitätstheorie** genannt.

In der Quanteninformatik sind wir daran interessiert, **Quantenalgorithmen** zu entwerfen, welche Rechenprobleme lösen, indem sie Quantenzustände anstelle von Bitstrings manipulieren. Die elementaren Anweisungen dabei sind Gatter, wie das Hadamard-Gatter, das kontrollierte-NOT-Gatter oder eine Messung. Wir werden diese Quantenalgorithmen bildlich durch Quantenschaltungen darstellen, die du schon oft in QUIRKY erstellt hast. Es ist aber auch möglich, eine textuelle Darstellung zu verwenden, wie man sie von einem üblichen Computerprogramm erwarten würde. Zum Beispiel entspricht die Quantenschaltung auf der linken Seite dem rechten Programmtext:



wobei q1 und q2 sich auf die zwei Qubits beziehen (bedenke, dass der untere Draht dem *ersten* Qubit entspricht). Gegeben eines Rechenproblems können wir dann vergleichen, wie viele elementare Operationen die besten bekannten klassischen- und Quantenalgorithmen jeweils benötigen, um das Problem zu lösen. So können wir den Vorteil zukünftiger Quantencomputer gegenüber klassischen Computern besser verstehen.

## 5.1 Mit Quantenorakeln sprechen

In dieser Quest werden wir uns einige Quantenalgorithmen anschauen und mit klassischen Algorithmen vergleichen, die dasselbe Problem lösen. Da es im Allgemeinen sehr schwer ist, die tatsächliche Anzahl an elementaren Operationen für das Lösen eines Problems zu zählen, werden wir hier ein vereinfachtes Maß der Komplexität nutzen (das machen Informatiker:innen immer dann, wenn sie sich das Leben einfacher machen wollen).

Wenn ein Computer ein Programm ausführt, muss er verschiedene Arten von Operationen ausführen. Die langsamsten davon sind jene, welche auf Daten zugreifen müssen. Zum Beispiel, wenn Daten aus dem Speicher, von einer Festplatte – oder im schlimmsten Fall – von einem anderen Computer über das Internet ausgelesen werden müssen. Sobald die relevanten Daten eingelesen wurden, kann das eigentliche Verarbeiten sehr schnell gehen. Daher können wir die Laufzeit eines Programms grob einschätzen, indem wir nur die Operationen zählen, die auf Daten zugreifen.

Das kennst du vielleicht vom Öffnen von komplizierten Webseiten oder großen Dokumenten. Es kann eine Weile dauern, bis diese geladen sind, danach ist das Interagieren mit der Webseite oder das Einfügen einer weiteren Zeile in das Dokument recht schnell.

Man könnte sich das auch so vorstellen, dass die Informationen, auf die man zugreifen will, eigentlich durch einen anderen Algorithmus, oder eine Subroutine (ein Unterprogramm) des Algorithmus' produziert wird. Und diese Subroutine ist sehr langsam. Zum Beispiel könnte sie die Informationen aus der Festplatte auslesen oder über das Internet empfangen. Oder die Informationen könnten gar nicht direkt verfügbar sein und müssen erst durch sehr komplizierte Berechnungen produziert werden. Auf jeden Fall braucht die Subroutine sehr lange um die Informationen zu liefern, also rufst du sie so selten wie möglich auf. Wir nennen solch eine Subroutine **Orakel**, da sie sehr klug ist und alle Antworten kennt, aber auch ein bisschen langsam ist und eine Weile nachdenken muss, um die Antwort zu liefern.

Formal gesagt ist ein **klassisches Orakel** einfach eine Funktion  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  wobei  $\{0, 1\}^n$  die Menge aller Bitstrings der Länge  $n$  bezeichnet. Du kannst dir die Eingabe  $x \in \{0, 1\}^n$

als Frage und die Ausgabe, das Bit  $f(x)$  als Antwort vorstellen. Jedes mal wenn du die Funktion  $f$  für eine Eingabe auswertest, stellst du eine Frage, die dieser Eingabe entspricht, und erhältst eine Ja/Nein-Antwort, die dem Wert der Funktion entspricht.

Beispielsweise kannst du Zugriff auf eine Festplatte oder anderen Speicher durch ein Orakel modellieren. Wenn du beispielsweise 4 Bits Speicherplatz hast, kannst du diesen als Funktion  $f : \{0, 1\}^2 \rightarrow \{0, 1\}$  modellieren, die, gegeben der binären Adresse, das zugehörige Bit ausgibt. Willst du dann alle vier Bits herausfinden, musst du  $f$  vier mal ausführen, um die vier Bitwerte  $f(00), f(01), f(10), f(11)$  zu erhalten.

Wichtig ist, dass es bei den Rechenproblemen, die wir lösen wollen *nicht* darum geht, den Wert von  $f$  auf einer bestimmten Eingabe zu finden. Solche Probleme wären trivial, da man sie lösen kann indem man das Orakel nur einmal fragt, da das genau die Aufgabe des Orakels ist – es gibt dir den Wert einer Funktion für eine beliebige Eingabe deiner Wahl. Die Probleme, an denen wir interessiert sind, sind etwas komplizierter. Wir wollen stattdessen eine Eigenschaft der Funktion  $f$  bestimmen, indem wir sie so selten wie möglich auswerten.

Zum Beispiel könnten wir wissen wollen, ob  $f(x) = 0$  für alle  $x \in \{0, 1\}^n$  gilt. In diesem Fall könnten wir das Orakel zufällige Werte von  $x$  fragen, bis wir eines finden, für das  $f(x) = 1$  gilt. Wir werden bald noch andere solche Beispiele sehen.

### 5.1.1 Umkehrbare Berechnungen

Bevor wir spannende neue Quantenalgorithmen suchen, sollten wir erst einmal sichergehen, dass wir auf Quantencomputern immer noch alles berechnen können, was wir auch auf klassischen Computern berechnen können. Wir wollen also sichergehen, dass Quantencomputer nicht *weniger* leistungsfähig sind als klassische Computer! Das ist nicht direkt offensichtlich, da Quantencomputer ganz anders als klassische funktionieren. Insbesondere müssen alle Operationen, die ein Quantencomputer ausführen kann **umkehrbar** oder **invertierbar** sein, wie wir schon in §2.4.2 und §4.1.3 gesehen haben, aber das ist für klassische Computer normalerweise nicht der Fall. Wer hat noch nie versehentlich eine Datei gelöscht oder vergessen, die Änderungen in einem Dokument zu speichern, und damit die gesamte Arbeit verloren? Wenn alle Vorgänge umkehrbar wären, bräuchte man sich um solche Kleinigkeiten keine Gedanken machen.

Das wirft also folgende Frage auf: woher wissen wir, dass Quantencomputer alles berechnen können, was klassische Computer auch berechnen können? Man kann diese Frage beantworten, indem man zeigt, dass Umkehrbarkeit die Rechenmöglichkeiten eines gewöhnlichen Computers nicht einschränkt und daher auch keine Einschränkung für Quantencomputer darstellt. Anders gesagt: Wir zeigen, dass jede Berechnung umkehrbar gemacht werden kann, also auch auf einem Quantencomputer ausgeführt werden kann.

Um zu verstehen, wie das funktionieren kann, schauen wir uns beispielhaft das berechnen der logischen AND Funktion (engl. für “UND”) auf zwei Bits an. Wenn beide Bits 1 sind, ergibt die Funktion auch 1, ansonsten ergibt sie 0. Also können wir die Funktion mit folgender Wahrheitstabelle beschreiben:

$x_1$	$x_2$	$\text{AND}(x_1, x_2)$
0	0	0
0	1	0
1	0	0
1	1	1

(5.1)

Mit der Notation aus §3.1 können wir sie mathematisch als Operation auf zwei Bits schreiben:

$$[x_1, x_2] \mapsto [\text{AND}(x_1, x_2)].$$

Diese Operation ist offensichtlich nicht umkehrbar, da sie nicht genauso viele Eingabe- wie Ausgabe-Bits hat. Wenn wir wissen, dass das Ergebnis der AND Operation 0 ist, können wir

keine präzisen Aussagen über die beiden Eingabebits treffen – wie die Wahrheitstabelle zeigt, gibt es drei Möglichkeiten.

Wie können wir dieses Problem lösen? Lass uns doch mal versuchen, das erste Bit zu behalten und ein zweites Ausgabebit hinzuzufügen, welches das Ergebnis beinhaltet:

$$[x_1, x_2] \mapsto [x_1, \text{AND}(x_1, x_2)].$$

Schon besser, nun haben wir zwei Eingabe- und zwei Ausgabe-Bits. Aber ist die Operation umkehrbar? Also können wir aus jeder Ausgabe die Eingabe rekonstruieren? Offensichtlich können wir das erste Bit  $x_1$  immer rekonstruieren, da es in der Ausgabe enthalten ist. Aber was ist mit  $x_2$ ? Wenn  $x_1 = 0$  gilt, muss laut Wahrheitstabelle das Ergebnis immer  $[00]$  sein – egal welchen Wert  $x_2$  hat. Daher ist auch diese Operation nicht umkehrbar und dieser Ansatz funktioniert nicht.

Es scheint hoffnungslos zu sein. Ist es überhaupt möglich die AND-Funktion umkehrbar zu bauen? Wenn man nicht weiter kommt, muss man manchmal um die Ecke denken! Warum müssen wir uns überhaupt auf nur zwei Bits beschränken? Durch behalten der beiden Eingabebits und hinzuzufügen eines dritten Bits zur Ausgabe, sollte die Operation umkehrbar werden:

$$[x_1, x_2, 0] \mapsto [x_1, x_2, \text{AND}(x_1, x_2)]. \quad (5.2)$$

Jetzt ist es ganz einfach die Operation zu invertieren – gegeben eines beliebigen Ausgabe Bitstrings können wir einfach das letzte Bit vergessen und durch  $[0]$  ersetzen um die Eingabe wiederherzustellen. Ist zum Beispiel die Ausgabe  $[111]$ , so muss die Eingabe  $[110]$  gewesen sein.

Sind wir also jetzt fertig? Noch nicht ganz! Beachte, dass Gl. (5.2) die Operation nur *teilweise* festlegt – wenn die Eingabe  $[111]$ , oder ein beliebiger anderer auf 1 endender Bitstring ist, sagt uns Gl. (5.2) *nichts* darüber wie sich die Operation verhalten sollte. Da die vier Inputstrings, welche auf 0 enden, auf vier verschiedene Ausgaben abgebildet werden, können wir Gl. (5.2) auf eine beliebige Art zu einer umkehrbaren Operation auf drei Bits erweitern. Aber gibt es einen systematischen Ansatz dafür?

Bemerke, dass Gl. (5.2) das letzte Bit von  $[0]$  zu  $[1]$  flippt, wenn  $\text{AND}(x_1, x_2) = 1$  gilt. Genauso könnten wir sagen, wenn das letzte Bit  $[1]$  ist, flippt unsere Operation es zurück zu  $[0]$ , falls  $\text{AND}(x_1, x_2) = 1$  gilt. Wir können also Gl. (5.2) wie folgt auf *alle* möglichen Eingaben erweitern:

$$[x_1, x_2, y] \mapsto [x_1, x_2, y \oplus \text{AND}(x_1, x_2)], \quad (5.3)$$

für alle  $x_1, x_2, y \in \{0, 1\}$ , wobei ' $\oplus$ ' die Addition Modulo 2 darstellt.

Die Operation in Gl. (5.3) ist also auf allen möglichen Eingaben definiert. Aber ist sie auch endlich umkehrbar? Ja, ist sie! Tatsächlich ist sie sogar ihr eigenes Inverses! Das bedeutet, wenn man sie zweimal hintereinander ausführt, erhält man wieder die ursprüngliche Eingabe:

$$\begin{aligned} [x_1, x_2, y] &\mapsto [x_1, x_2, y \oplus \text{AND}(x_1, x_2)] \\ &\mapsto [x_1, x_2, y \oplus \text{AND}(x_1, x_2) \oplus \text{AND}(x_1, x_2)] = [x_1, x_2, y]. \end{aligned}$$

Im dritten Schritten haben wir ausgenutzt, dass  $a \oplus a = 0$  für alle  $a \in \{0, 1\}$ , siehe Gl. (3.20). Also haben wir erfolgreich einen Weg gefunden, die AND-Funktion umkehrbar umzusetzen.

### 5.1.2 Bit-Orakel

Dieser Ansatz funktioniert nicht nur für die logische AND-Funktion, sondern für jede beliebige Funktion

$$f: \{0, 1\}^n \rightarrow \{0, 1\}$$

die  $n$  Bits als Eingabe erhält und ein einzelnes Bit ausgibt. Für jede solche Funktion können wir eine Operation auf  $(n + 1)$  Bits definieren:

$$[x_1, \dots, x_n, y] \mapsto [x_1, \dots, x_n, y \oplus f(x_1, \dots, x_n)] \quad (5.4)$$

für alle  $x_1, \dots, x_n, y \in \{0, 1\}$ . Genau wie Gl. (5.3) ist diese Operation ihr eigenes Inverse, zweimaliges anwenden verändert den Zustand also nicht.

Es ist beruhigend, dass wir jede Funktion  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  wie in Gl. (5.4) umkehrbar implementieren können. Zunächst einmal bedeutet das, dass jede Berechnung auf einem klassischen Computer so durchgeführt werden kann, dass die ursprüngliche Eingabe wiederhergestellt werden kann. Weiterhin können Berechnungen, die umkehrbar sind, auch auf Quantencomputern durchführen. Das bedeutet, dass Quantencomputer alles berechnen können was auch klassische Computer können! Außerdem können wir damit jede Funktion  $f$  als Orakel auf einem Quantencomputer implementieren.

Lass uns einmal sehen, wie das funktioniert. Die Quantenversion der Operation aus Gl. (5.4) ist wie folgt definiert:

$$U_f |x_1, \dots, x_n, y\rangle = |x_1, \dots, x_n, y \oplus f(x_1, \dots, x_n)\rangle \quad (5.5)$$

für alle  $x_1, \dots, x_n, y \in \{0, 1\}$ . Gl. (5.5) bestimmt dabei, wie  $U_f$  sich auf allen Basiszuständen von  $n + 1$  Qubits verhält. Wie immer können wir diese Definition durch Linearität auf beliebige Zustände auf  $n + 1$  Qubits erweitern. Da  $U_f$  einfach nur die Basiszustände permutiert, kannst du wie in Übungsaufgabe 4.4 zeigen, dass es eine gültige Quantenoperation darstellt.

Wir nennen die Quantenoperation  $U_f$  aus Gl. (5.5) das **Bit-Orakel für  $f$**  (wir könnten auch die Funktion aus Gl. (5.4) als klassisches Bit-Orakel bezeichnen, aber wir brauchen diesen Namen nicht). Die Bezeichnung ‘Orakel’ bedeutet einfach, dass sich das Anwenden dieser Operation so verhält, als würden wir ein allmächtiges Orakel bitten, uns den Wert der Funktion für jede beliebige Eingabe zu geben. Wir wissen nicht genau, wie das Orakel implementiert ist oder woher es die Antworten bekommt, wir werden nur zählen, wie viele Fragen wir an das Orakel stellen müssen, um eine gewisse Eigenschaft der Funktion  $f$  herauszufinden. Viele interessante Algorithmische Probleme können so modelliert werden, wie wir im Rest der Quest sehen werden.

Das Konzept eines Orakels ist ein wenig wie das ‘Rate meine Zahl’-Spiel. Ein:e Freund:in von dir denkt sich eine Zahl aus und du fragst dann Fragen der Form “Ist deine Zahl  $x$ ”? Darauf erhältst du dann entweder “Ja” oder “Nein” als Antwort. Mit anderen Worten stellt dein:e Freund:in eine Funktion dar, die “Nein” auf allen Eingaben außer einer einzigen (der Zahl, die festgelegt wurde) ausgibt. Jede Frage die du stellst gibt dir mehr Informationen über die Zahl. Du könntest dir zum Beispiel Gedanken darüber machen, wie viele Fragen du stellen musst, um die Zahl zu finden. Und was wäre, wenn du deine Fragen an ein Quantenorakel  $U_f$  stellen könntest, anstelle an deine:n Freund:in? Könntest du die Lösung dann mit weniger Fragen finden? In dieser Woche wirst du mehrere interessante Beispiele kennenlernen, wo das tatsächlich der Fall ist!

Lass uns mal einige Beispiele für Bit-Orakel besprechen. Nehmen wir an,  $f$  ist die AND-Funktion. Laut Gl. (5.1) können wir AND als Multiplikation modulo 2 interpretieren, da  $AND(x_1, x_2) = x_1 x_2$ . Das entsprechende Bit-Orakel

$$U_{AND} |a, b, c\rangle = |a, b, c \oplus ab\rangle$$

ist nichts anderes als das Toffoli Gatter aus Übungsaufgabe 4.4. Selbst für  $n = 1$  und die Funktion  $f(x) = x$ , die einfach ihre Eingabe ausgibt, erhalten wir ein interessantes Ergebnis: für alle  $a, b \in \{0, 1\}$  gilt dann

$$U_f |a, b\rangle = |a, b \oplus a\rangle.$$

Das entspricht einfach dem bekannten kontrolliertem-NOT-Gatter  $\text{CNOT}_{1 \rightarrow 2}$  aus Gl. (3.47) in §3.2.4! Das Bit-Orakel bildet also mehrere interessante Quantenoperationen, die wir zuvor per Hand definiert hatten. In der folgenden Übung kannst du versuchen, alle anderen Ein-Bit-Orakel zu implementieren.

### Übungsaufgabe 5.1: Bit-Orakel für Ein-Bit-Funktionen

Sei  $f : \{0, 1\} \rightarrow \{0, 1\}$  eine Funktion mit einem einzelnen Eingabe- und Ausgabe-Bit. Eine solche Funktion ist vollständig durch die Werte  $f(0), f(1) \in \{0, 1\}$  definiert. Das sind zwei Bits, also gibt es genau vier solcher Funktionen. Wir haben gerade besprochen, wie man die das Bit-Orakel  $U_f$  für die Funktion  $f(x) = x$  implementiert. Kannst du die Bit-Orakel  $U_f$  für die anderen drei Funktionen in QUIRKY implementieren?

#### 5.1.3 Phasen-Orakel

Da das Bit-Orakel  $U_f$  eine Quantenoperation ist, können wir sie nicht nur auf Basiszustände  $|x_1, \dots, x_n, y\rangle$ , sondern auch auf allgemeinere Quantenzustände anwenden. Aber warum würden wir das tun wollen? Nun, wenn wir dem Orakel nur 'klassische' Fragen stellen, werden wir wahrscheinlich keine Quanten-Verbesserung erreichen! Deswegen wollen wir einmal untersuchen, wie sich das Bit-Orakel  $U_f$  für eine beliebige Funktion  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  verhält, wenn wir das letzte Register auf  $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$  setzen. Zuerst bemerken wir den folgenden interessanten Fakt:

$$\text{NOT } |-\rangle = \frac{1}{\sqrt{2}}(\text{NOT } |0\rangle - \text{NOT } |1\rangle) = \frac{1}{\sqrt{2}}(|1\rangle - |0\rangle) = -|-\rangle.$$

Soll bedeuten, wenn wir ein Qubit im Zustand  $|-\rangle$  invertieren erhalten wir ein Vorzeichen. Genauso können wir auch berechnen, wie sich das Bit-Orakel auf einen Zustand der Form  $|x_1, \dots, x_n\rangle \otimes |-\rangle$  auswirkt. Durch Linearität gibt uns Gl. (5.5), dass

$$\begin{aligned} & U_f (|x_1, \dots, x_n\rangle \otimes |-\rangle) \\ &= U_f \left( \frac{1}{\sqrt{2}} |x_1, \dots, x_n, 0\rangle - \frac{1}{\sqrt{2}} |x_1, \dots, x_n, 1\rangle \right) \\ &= \frac{1}{\sqrt{2}} |x_1, \dots, x_n, f(x_1, \dots, x_n)\rangle - \frac{1}{\sqrt{2}} |x_1, \dots, x_n, f(x_1, \dots, x_n) \oplus 1\rangle \\ &= |x_1, \dots, x_n\rangle \otimes \frac{1}{\sqrt{2}} (|f(x_1, \dots, x_n)\rangle - |f(x_1, \dots, x_n) \oplus 1\rangle) \\ &= (-1)^{f(x_1, \dots, x_n)} |x_1, \dots, x_n\rangle \otimes |-\rangle. \end{aligned}$$

Mit anderen Worten erhalten wir wieder den Zustand  $|-\rangle$ , allerdings mit einem negativen Vorzeichen wenn  $f(x_1, \dots, x_n) = 1$ . Somit haben wir also folgende Quantenoperation auf den ersten  $n$  Qubit gebaut:

$$O_f |x_1, \dots, x_n\rangle = (-1)^{f(x_1, \dots, x_n)} |x_1, \dots, x_n\rangle, \quad (5.6)$$

für alle Bitstrings  $x_1, \dots, x_n \in \{0, 1\}$ . Wir nennen  $O_f$  das **Phasen-Orakel für  $f$** .

Interessanterweise arbeitet das Phasen-Orakel  $O_f$  für eine Funktion  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  auf  $n$  Qubits, da das Ergebnis in der Amplitude gespeichert wird. Das ist ein Unterschied zum Bit-Orakel  $U_f$ , welches für die Ausgabe ein zusätzliches Qubit benötigt und deswegen auf  $n + 1$  arbeitet.

Auf den ersten Blick scheint das Phasen-Orakel nicht viel zu machen, da es bei Anwendung auf einem Basiszustand nur ein allgemeines Vorzeichen einfügt, wobei wir aus Übungsaufgabe 2.7 wissen, dass dieses nicht gemessen werden kann. Wenn wir es aber auf einen Zustand

in superposition anwenden, kann das Phasen-Orakel ein relatives Vorzeichen einfügen, was zu interessanten Ergebnissen führen kann. Wenden wir beispielsweise für  $n = 2$  Qubits das Phasen-Orakel auf einen allgemeinen Zwei-Qubit-Zustand

$$|\psi\rangle = \psi_{00} |00\rangle + \psi_{01} |01\rangle + \psi_{10} |10\rangle + \psi_{11} |11\rangle$$

an, erhalten wir

$$O_f |\psi\rangle = (-1)^{f(0,0)} \psi_{00} |00\rangle + (-1)^{f(0,1)} \psi_{01} |01\rangle + (-1)^{f(1,0)} \psi_{10} |10\rangle + (-1)^{f(1,1)} \psi_{11} |11\rangle.$$

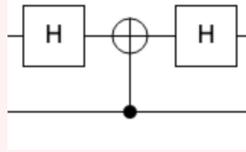
Es stellt sich heraus, dass das Phasen-Orakel  $O_f$  nützlicher und meist einfacher in Quantenalgorithmen anzuwenden ist als das Bit-Orakel  $U_f$ , weswegen wir das Bit-Orakel nicht mehr weiter benutzen.

### Übungsaufgabe 5.2: Phasen-Orakel für eine Ein-Qubit-Funktion

Erinnere dich an Übungsaufgabe 5.1, wo die vier Funktionen  $f : \{0, 1\} \rightarrow \{0, 1\}$  mit einem Eingabe- und Ausgabe-Bit vorgestellt wurden. Kannst du das Phasen-Orakel  $O_f$  für jede davon in QUIRKY implementieren?

### Hausaufgabe 5.1: Bestimme die Funktion anhand ihres Phasen-Orakels

Betrachte den folgenden Zwei-Qubit-Schaltkreis (wie üblich ist der untere Draht das erste Qubit):



Für welche Funktion  $f: \{0,1\}^2 \rightarrow \{0,1\}$  stellt der Schaltkreis das Phasen-Orakel dar?

**Hinweis:** Benutze, dass  $H \text{NOT} H = Z$ , was aus Übungsaufgabe 4.5 folgt.

Lass uns kurz zusammenfassen, was wir bisher erreicht haben: Mit Bit-Orakeln können wir Quantencomputer eine Funktion  $f: \{0,1\}^n \rightarrow \{0,1\}$  genau so auswerten lassen, wie mit einem klassischen Computern, der umkehrbar agiert (vergleiche Gl. (5.4) und (5.5)). Das ist wichtig, da wir also nicht Äpfel mit Birnen vergleichen wenn wir zählen, wie viele Fragen man das Bit-Orakel  $U_f$  fragen muss, um eine Eigenschaft über  $f$  zu lernen, im Vergleich zu wie oft man  $f$  auf einem klassischen Computer auswerten muss, um die gleiche Eigenschaft zu lernen. Und da wir gerade gezeigt haben, dass man das Phasen-Orakel  $O_f$  immer mit einem Bit-Orakel  $U_f$  bauen kann, macht es keinen Unterschied wenn wir stattdessen das Phasen-Orakel  $O_f$  fragen

## 5.2 Quantenalgorithmen

In diesem Abschnitt werden wir uns verschiedene Quantenalgorithmen anschauen, die ein Rechenproblem viel schneller als klassische Computer lösen können. Solche Verbesserungen der Laufzeit sind sehr unerwartet, da Quantenmechanik auf den ersten Blick nichts mit der eigentlichen Berechnung zu tun haben scheint. Trotzdem können Quantenmechanische Phänomene wie Interferenz beeindruckende Beschleunigungen erzielen. Wie wir bereits im Voraus erklärt hatten, zählen wir dabei nur die Anzahl der Fragen, die an ein Orakel gestellt werden. Angenommen wir können also die Funktion  $f$  auf jeder Eingabe evaluieren, auf wie vielen Eingaben müssen wir die Funktion auswerten um eine Eigenschaft von  $f$  zu bestimmen. Oder anders gesagt, wie oft müssen wir das Orakel  $O_f$ , welches  $f$  auswerten kann, benutzen, um die Eigenschaft von  $f$  zu bestimmen.

### 5.2.1 Der Algorithmus von Deutsch

Es ist ein später Sonntagabend. Alice und Bob haben gerade eben ihre Hausaufgaben für den Quantencomputerkurs gemacht und wollen jetzt einen 3D-Film schauen. Als sie ihren holografischen Fernseher anschalten, stellen sie fest, dass der Film wegen unerwarteter dramatischer Neuigkeiten von der Internationalen Transgalaktischen Station verschoben wurde. Es gab einen schrecklichen Unfall: ein Modul mit den zwei Crew-Mitgliedern Hila und Iman hat sich vom Hauptschiff getrennt. Die letzte empfangene Nachricht vom Modul war, dass Iman verletzt wurde und blutete – er braucht dringend eine Bluttransfusion. Leider stehen sowohl Hila als auch Iman unter Schock und haben ihren eigenen Blutgruppen vergessen – sie können sich nur daran erinnern, dass sie jeweils entweder Blutgruppe A oder B hatten. Die Moderation der Sendung appelliert an alle Zuschauende, Vorschläge zu machen, wie Hila und Iman herausfinden könnten, ob sie die selbe Blutgruppe haben, dann könnte Hila nämlich ihr Blut an Iman übertragen um sein Leben zu retten. An Bord befindet sich nämlich ein Lympho-Transcoder, der die beiden Blutgruppen in den jeweils anderen umwandeln kann. Selbst wenn sie also nicht die gleiche Blutgruppe haben, könnte Hila den Transcoder nutzen um ihr Blut zum richtigen Typ umzuwandeln.

Nachdem sie diese Nachrichten gehört haben, entscheiden sich Alice und Bob zu überlegen, wie man Hila und Iman helfen könnte, anstelle den Film zu schauen. Die Nachrichtensendung setzt fort mit weiteren Informationen. Glücklicherweise hat das Modul einen Datenbank-Chip, auf dem Hila und Imans Blutgruppe gespeichert ist. Wir können diesen durch eine Funktion  $f: \{0, 1\} \rightarrow \{0, 1\}$  modellieren, wobei

$$f(0) = \begin{cases} 0 & \text{'} \\ 1 & \text{' .} \end{cases}$$

$$f(1) = \begin{cases} 0 & \text{'} \\ 1 & \text{' .} \end{cases}$$

Nun muss herausgefunden werden, ob  $f(0) = f(1)$  gilt oder nicht!

Die Lösung scheint einfach: Hila und Iman fragen die Datenbank einfach zweimal ab um ihre jeweiligen Blutgruppen  $f(0)$  und  $f(1)$  herauszufinden und vergleichen anschließend die Werte. Unglücklicherweise wurde der Chip bei dem Unfall beschädigt und die Nachrichtensendung berichtet, dass die Datenbank höchstwahrscheinlich nach einer einzigen Abfrage völlig zerstört sein würde.

Unsere beiden Protagonisten befinden sich in einer Pattsituation. Offensichtlich muss jeder klassische Algorithmus  $f$  genau zweimal auswerten um herauszufinden, ob  $f(0) = f(1)$ . Wenn wir den Wert von  $f(0)$  kennen hängt  $f(0) = f(1)$  immer noch von  $f(1)$  ab und lässt sich nicht bestimmen, außer man berechnet  $f(1)$ . Genauso lässt sich ein bekanntes  $f(1)$  nicht mit einem unbekanntem  $f(0)$  vergleichen. Egal welchen Ansatz man verfolgt, man muss sowohl den Wert von  $f(0)$  als auch von  $f(1)$  kennen, um  $f(0) = f(1)$  zu bestimmen. Gibt es da wirklich keinen Ausweg?

Nachdem Bob ein paar Bedienungsanleitungen durchblättert hat, stellt Bob fest, dass der Datenbank-Chip auch einen *Quantenmodus* besitzt. Wenn dieser aktiviert ist, wertet die Datenbank die Funktion nicht mehr klassisch aus, sondern nutzt stattdessen das Phasen-Orakel  $O_f$ . Könnte das vielleicht dabei helfen, das Problem zu lösen? Alice überlegt eine Weile und stellt dann überrascht fest, dass es genau für dieses Problem den **Algorithmus von Deutsch** gibt! Die beiden prüfen mit ein paar Berechnungen noch schnell, dass alles funktioniert und senden dann eine intergalaktische E-Mail mit Anweisungen zum lösen des Dilemmas an Hila und Iman. Ihre Instruktionen sind die folgenden:

1. Bereitet ein Qubit im Zustand  $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$  vor.
2. Nutzt den Datenbank-Chip im Quantenmodus um die Operation  $O_f$  auf das Qubit anzuwenden.
3. Wendet das Hadamard-Gatter  $H$  auf das Ausgabe-Qubit an und misst anschließend.
4. Wenn das Messergebnis 0 ist, haben Hila und Iman die gleiche Blutgruppe, ansonsten haben sie unterschiedliche.

Beachte, dass bei dieser Prozedur der Datenbank-Chip nur *einmal* abgefragt wird, um zu bestimmen, ob sie die gleiche Blutgruppe haben. Hier ist eine Implementierung des Algorithmus in QUIRKY:



Das Bild zeigt, dass das Ergebnis 1 ist, also haben Hila und Iman unterschiedliche Blutgruppen.

Aber wieso funktioniert der Algorithmus von Deutsch? Lass uns den Algorithmus Schritt für Schritt analysieren. Das erste Hadamard-Gatter erstellt den Zustand  $|+\rangle = H|0\rangle$ . Anschließend wenden wir das Phasen-Orakel  $O_f$  an, was zum folgenden Zustand führt

$$\begin{aligned} O_f|+\rangle &= \frac{1}{\sqrt{2}}O_f|0\rangle + \frac{1}{\sqrt{2}}O_f|1\rangle \\ &= \frac{1}{\sqrt{2}}(-1)^{f(0)}|0\rangle + \frac{1}{\sqrt{2}}(-1)^{f(1)}|1\rangle. \end{aligned}$$

Nach dem anwenden des zweiten Hadamard-Gatters, erhalten wir den Zustand:

$$\begin{aligned} HO_f|+\rangle &= \frac{1}{\sqrt{2}}(-1)^{f(0)}H|0\rangle + \frac{1}{\sqrt{2}}(-1)^{f(1)}H|1\rangle \\ &= \frac{1}{\sqrt{2}}(-1)^{f(0)}|+\rangle + \frac{1}{\sqrt{2}}(-1)^{f(1)}|-\rangle \\ &= \frac{1}{2}(-1)^{f(0)}(|0\rangle + |1\rangle) + \frac{1}{2}(-1)^{f(1)}(|0\rangle - |1\rangle) \\ &= \frac{(-1)^{f(0)} + (-1)^{f(1)}}{2}|0\rangle + \frac{(-1)^{f(0)} - (-1)^{f(1)}}{2}|1\rangle. \end{aligned} \tag{5.7}$$

Beachte, dass die beiden Vorzeichen  $(-1)^{f(0)}$  und  $(-1)^{f(1)}$  in der ersten Amplitude *addiert*, in der zweiten aber *subtrahiert* werden. Je nach den Werten von  $f(0)$  und  $f(1)$  sehen wir für jede Amplitude entweder konstruktive oder destruktive Interferenz (siehe §2.6.1). Tatsächlich bestimmt sich nur daran, ob  $f(0)$  und  $f(1)$  gleich sind oder nicht, welche Amplitude übrig bleibt:

$$\begin{aligned} f(0) = f(1) : \quad HO_f|+\rangle &= \pm|0\rangle, \\ f(0) \neq f(1) : \quad HO_f|+\rangle &= \pm|1\rangle. \end{aligned} \tag{5.8}$$

Es ist eine gute Übung, dies explizit zu verifizieren:

### Übungsaufgabe 5.3: Den Algorithmus von Deutsch verifizieren

Erinnere dich aus Übungsaufgabe 5.1 daran, dass es vier Funktionen  $f : \{0, 1\} \rightarrow \{0, 1\}$  gibt. Berechne für jede Funktion den Zustand  $HO_f|+\rangle$  mit Gl. (5.7).

Gl. (5.8) zeigt, dass die abschließende Messung nur dann das Ergebnis 0 ergibt, wenn  $f(0) = f(1)$  gilt. Also bestimmt der Algorithmus ob  $f(0) = f(1)$ . Dabei evaluiert der Algorithmus die Funktion  $f : \{0, 1\} \rightarrow \{0, 1\}$  nur ein einziges Mal mit dem Phasen-Orakel. Im Gegensatz dazu hatten wir gesehen, dass ein klassischer Algorithmus notwendigerweise beide Funktionswerte  $f(0)$  und  $f(1)$  getrennt auswerten muss.

Eine weitere Interpretation des Algorithmus von Deutsch ist, dass er die Summe der beiden Bits  $f(0)$  und  $f(1)$  modulo zwei berechnet. Das liegt daran, dass  $f(0) \oplus f(1) = 0$  nur genau dann gilt, wenn  $f(0) = f(1)$ . Aus Gl. (3.20) erinnern wir uns an die Definition von Addition modulo zwei:

$x_1$	$x_2$	$x_1 \oplus x_2$	
0	0	0	
0	1	1	
1	0	1	
1	1	0	(5.9)

Aus diesem Grund ist die Summe modulo zwei auch als XOR (engl. Abkürzung für "exklusives Oder") der beiden Bits, da sie 1 genau dann ergibt, wenn nur eines der beiden Bits gesetzt ist.

## 5.2.2 Die Hadamard-Transformation und Interferenz

Der Algorithmus von Deutsch ist zwar sehr überraschend, erreicht aber nur eine geringe Verbesserung gegenüber klassischen Algorithmen, nämlich 1 anstelle von 2 Ausführungen von  $f$ . Das könnte natürlich nützlich sein, wenn das Evaluieren von  $f$  sehr lange dauert, beispielsweise ein Jahr. Aber wenn es sich nur um Millisekunden handelt, macht es für die meisten keinen Unterschied eine Millisekunde länger zu warten (dafür aber viel weniger zu zahlen, da man keinen Quantencomputer bräuchte)! Um den Nutzen von Quantencomputern zu zeigen, würden wir gerne eine Berechnung um mehr als nur den Faktor 2 beschleunigen.

Wenn wir uns nur Funktionen mit einem Bit anschauen, können wir eine stärkere Beschleunigung vergessen, da diese mit nur zwei Auswertungen voll bestimmt werden können:  $f(0)$  und  $f(1)$ . Daher schauen wir uns allgemeinere Funktionen  $f: \{0,1\}^n \rightarrow \{0,1\}$  mit  $n$  Eingabe-Bits an, da es davon viel mehr gibt (tatsächlich gibt es  $2^{2^n}$  viele). Bedenke, unser Ziel ist es *nicht*, die Funktion auszuwerten (das erreichen wir schon mit nur einer Evaluation des Orakels) sondern eine interessante Eigenschaft der Funktion zu bestimmen, die Funktionswerte auf verschiedenen Eingaben zueinander in Relation stellt. Gibt es Eigenschaften von  $n$ -Bit Funktionen, die wir effizient mit cleveren Quantenalgorithmien in Erfahrung bringen können?

Um den Algorithmus von Deutsch zu verallgemeinern, bemerken wir, dass es ein Schlüsselbestandteil war, ein Hadamard-Gatter vor und nach dem Phasen-Orakel einzufügen. In diesem Fall ist das Phasen-Orakel  $O_f$  eine Quantenoperation auf  $n$  Qubit, also könnten wir auf all diesen Qubits ein Hadamard-Gatter vor und nach dem Orakel einfügen. Die Quantenoperation, die parallel auf allen  $n$  Qubits eine Hadamard-Operation anwendet nennt sich **Hadamard-Transformation**. Erinnerung dich aus §3.2.3, dass wir sie so schreiben können:

$$H \otimes \cdots \otimes H.$$

Lass uns zunächst herausfinden, wenn wir die Hadamard-Transformation auf einen Basiszustand anwenden. Auf  $|0 \dots 0\rangle$  ergibt sich einfach die gleichverteilte Superposition über alle Basiszustände:

$$\begin{aligned} (H \otimes \cdots \otimes H) |0 \dots 0\rangle &= H|0\rangle \otimes \cdots \otimes H|0\rangle \\ &= \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes \cdots \otimes \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \\ &= \frac{1}{\sqrt{2^n}} (|0 \dots 00\rangle + |0 \dots 01\rangle + \dots + |1 \dots 11\rangle). \end{aligned}$$

Dieser Zustand ist eine Superposition von  $2^n$  Basiszuständen. Kompakter könnte man auch schreiben:

$$(H \otimes \cdots \otimes H) |0 \dots 0\rangle = \frac{1}{\sqrt{2^n}} \sum_{y_1, \dots, y_n \in \{0,1\}} |y_1, \dots, y_n\rangle \quad (5.10)$$

Die Notation  $\sum_{y_1, \dots, y_n \in \{0,1\}}$  bedeutet, dass die Summe von  $|y_1, \dots, y_n\rangle$  über alle möglichen Kombinationen von den Bits  $y_1, \dots, y_n$  gebildet wird.

Allgemeiner gesagt ist die Ausgabe auf einem beliebigen Basiszustand immer eine Superposition von allen  $2^n$  Basiszuständen mit gleichen Amplituden, bis auf Vorzeichen. Diese Vorzeichen genauer zu bestimmen kann manchmal schwierig sein.

### Übungsaufgabe 5.4: Zwei Hadamards

In Gl. (2.20) hatten wir gesehen, dass  $H|x_1\rangle = (|0\rangle + (-1)^{x_1}|1\rangle)/\sqrt{2}$  für alle  $x_1 \in \{0,1\}$  gilt.

1. Schreibe den Zustand  $H|x_1\rangle$  für beliebige  $x_1 \in \{0, 1\}$  wie folgt:

$$H|x_1\rangle = \frac{1}{\sqrt{2}} \sum_{y_1 \in \{0,1\}} (-1)^{\boxed{???}} |y_1\rangle,$$

wobei  $\boxed{???}$  ein Ausdruck ist, der  $x_1, y_1 \in \{0, 1\}$  enthält. Finde diesen Ausdruck.

2. Schreibe den Zustand  $(H \otimes H)|x_1, x_2\rangle$  für beliebige  $x_1, x_2 \in \{0, 1\}$  wie folgt:

$$(H \otimes H)|x_1, x_2\rangle = \frac{1}{2} \sum_{y_1, y_2 \in \{0,1\}} (-1)^{\boxed{???}} |y_1, y_2\rangle,$$

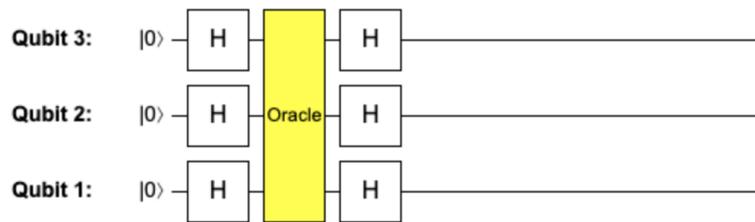
wobei  $\boxed{???}$  ein Ausdruck ist, der  $x_1, x_2, y_1, y_2 \in \{0, 1\}$  enthält.  
Findest du auch diesen Ausdruck?

Hast du die Lösung gefunden? Gut, dann darfst du weiter lesen!

Im Allgemeinen kann man sich die folgende Formel herleiten, die die Vorzeichen der Amplituden beschreibt, die man erhält, wenn man die Hadamard-Transformation auf einen beliebigen  $n$ -Qubit-Basiszustand anwendet: Für alle  $x_1, \dots, x_n \in \{0, 1\}$ ,

$$(H \otimes \dots \otimes H) |x_1, \dots, x_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{y_1, \dots, y_n \in \{0, 1\}} (-1)^{x_1 y_1 + \dots + x_n y_n} |y_1, \dots, y_n\rangle. \quad (5.11)$$

Nun können wir den Algorithmus von Deutsch recht einfach verallgemeinern. Wir starten mit einem  $n$ -Qubit-Basiszustand  $|0 \dots 0\rangle$ , wenden zunächst die Hadamard-Transformation an und anschließend das Phasen-Orakel  $O_f$  der Funktion  $f: \{0, 1\}^n \rightarrow \{0, 1\}$ , die wir untersuchen wollen, und abschließend eine weitere Hadamard-Transformation. Für beispielsweise  $n = 3$  ergibt sich der folgende Schaltkreis in QUIRKY:



Für allgemeine  $n$  lässt sich der finale  $n$ -Qubit-Zustand wie folgt beschreiben:

$$(H \otimes \dots \otimes H) O_f (H \otimes \dots \otimes H) |0 \dots 0\rangle. \quad (5.12)$$

Können wir das auch expliziter schreiben? Wie schon zuvor, rechnen wir Schritt für Schritt. Zunächst wenden wir die Hadamard-Transformation auf den Null-Basiszustand an. Nach Gl. (5.10) erhalten wir die uniforme Überlagerung aller Basiszustände:

$$(H \otimes \dots \otimes H) |0 \dots 0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x_1, \dots, x_n \in \{0, 1\}} |x_1, \dots, x_n\rangle.$$

Anschließend wenden wir das Phasen-Orakel  $O_f$  an:

$$O_f (H \otimes \dots \otimes H) |0 \dots 0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x_1, \dots, x_n \in \{0, 1\}} (-1)^{f(x_1, \dots, x_n)} |x_1, \dots, x_n\rangle.$$

Was erreicht die abschließende Hadamard-Transformation? Durch Linearität können wir Gl. (5.11) auf jeden Basisvektor anwenden, also erhalten wir den folgenden Ausdruck für den Zustand in Gl. (5.12):

$$\begin{aligned} & (H \otimes \dots \otimes H) O_f (H \otimes \dots \otimes H) |0 \dots 0\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{x_1, \dots, x_n \in \{0, 1\}} (-1)^{f(x_1, \dots, x_n)} \frac{1}{\sqrt{2^n}} \sum_{y_1, \dots, y_n \in \{0, 1\}} (-1)^{x_1 y_1 + \dots + x_n y_n} |y_1, \dots, y_n\rangle \\ &= \sum_{y_1, \dots, y_n \in \{0, 1\}} \frac{1}{2^n} \left( \sum_{x_1, \dots, x_n \in \{0, 1\}} (-1)^{x_1 y_1 + \dots + x_n y_n} (-1)^{f(x_1, \dots, x_n)} \right) |y_1, \dots, y_n\rangle, \end{aligned} \quad (5.13)$$

wobei die zwei Summen im letzten Schritt getauscht wurden. Für  $n = 1$  entspricht das genau Gl. (5.7). Im Allgemeinen sieht dieser Ausdruck aber kompliziert und schwer zu interpretieren aus – es scheint, der Schaltkreis berechnet eine Superposition der Basiszustände, wobei die Amplitude eine seltsame Summe von Plus- und Minuszeichen ist!

Wir können versuchen ein intuitives Verständnis zu erreichen, indem wir uns die Amplitude von  $|0 \dots 0\rangle$  in Gl. (5.13) anschauen. Das Quadrat dieser Zahl ist die Wahrscheinlichkeit, dass

beim messen der  $n$  Qubits alle Ergebnisse Null sind. Da diese Amplitude  $y_1 = \dots = y_n = 0$  entspricht, ist sie gegeben durch

$$\frac{1}{2^n} \sum_{x_1, \dots, x_n \in \{0,1\}} (-1)^{f(x_1, \dots, x_n)}.$$

Aber was bedeutet das? Nehmen wir an, es gibt  $N_f$  Eingabe-Bitstrings für die  $f$  Null ergibt und  $2^n - N_f$  Bitstrings für die  $f$  Eins ergibt. Dann,

$$\frac{1}{2^n} \sum_{x_1, \dots, x_n \in \{0,1\}} (-1)^{f(x_1, \dots, x_n)} = \frac{N_f - (2^n - N_f)}{2^n} = \frac{2N_f - 2^n}{2^n}.$$

Dabei gibt es zwei interessante Extremfälle: <sup>13</sup>

- Wenn  $f$  eine *konstante Funktion* ist, gilt entweder  $N_f = 0$  (für die Alles-Null-Funktion) oder  $N_f = 2^n$  (für die Alles-Eins-Funktion). In jedem Fall ist die Amplitude  $\pm 2^n / 2^n = \pm 1$ . Da die Summe der Quadrate der Amplituden genau 1 ergeben muss, können wir Schlussfolgern, dass alle anderen Amplituden in Gl. (5.13) genau 0 sein müssen. Mit anderen Worten, wenn  $f$  konstant ist, muss der Zustand in Gl. (5.13) einfach  $\pm |0 \dots 0\rangle$ . Wenn wir alle  $n$  Qubits dieses Zustands messen, müssen alle Ergebnisse Null sein.
- Wenn  $f$  eine *balanced Funktion*<sup>14</sup> ist, also das Ergebnis der Funktion genauso oft Null wie Eins ist, dann ist  $N_f = 2^n / 2$  und die Amplitude ist 0. Wenn man also den Zustand in Gl. (5.13) misst, werden niemals alle Ergebnisse Null sein. Also wird für eine balanced Funktion immer mindestens ein Messergebnis Eins sein.

Bemerke, dass diese zwei Fälle sehr verschiedenen Interferenzmustern entsprechen (siehe §2.6.1). Im ersten Fall wird die Amplitude von  $|0 \dots 0\rangle$  durch stark fokussierte *konstruktive* Interferenz zu  $\pm 1$  verstärkt, wobei gleichzeitig alle anderen Amplituden durch verbreitete *destruktive* Interferenz verschwinden. Im zweiten Fall erfährt  $|0 \dots 0\rangle$  dagegen *destruktive* Interferenz, wodurch überall sonst Nicht-Null-Amplituden auftauchen. Die Hadamard-Transformation zeigt auf wie wichtig Interferenz für die Quanteninformatik ist. In den nächsten zwei Abschnitten wird sie eine entscheidende Rolle für das Design von Quantenalgorithmien auf vielen Qubits spielen.

### 5.2.3 Der Deutsch-Jozsa-Algorithmus

Sind die obigen Beobachtungen für irgendwas nützlich? Ja, sind sie! Sei  $f: \{0,1\}^n \rightarrow \{0,1\}$  eine unbekannte Funktion, die entweder konstant oder balanced ist. Dann gibt es einen einfachen Quantenalgorithmus, welcher feststellen kann, welcher der beiden Fälle zutrifft, der  $f$  nur ein einziges mal auswertet (also einmal  $O_f$  anwendet).

Dieser Algorithmus heißt **Deutsch-Jozsa-Algorithmus** und funktioniert in fünf einfachen Schritten:

1. Starte mit  $|0 \dots 0\rangle$ .
2. Wende die Hadamard-Transformation  $H \otimes \dots \otimes H$  an.
3. Wende das Phasen-Orakel  $O_f$  der entsprechenden Funktion  $f$  an.
4. Wende wieder die Hadamard-Transformation  $H \otimes \dots \otimes H$  an.

<sup>13</sup>Für  $n = 1$  ist jede Funktion entweder konstant oder balanced. Für  $n > 1$  ist das nicht der Fall (z.B. ist die AND-Funktion weder konstant noch balanced).

<sup>14</sup>(engl. für "gleichmäßig/balanciert")

5. Messe alle Qubits. Wenn alle Ergebnisse Null sind, ist die Funktion  $f$  konstant. Ansonsten muss sie balanced sein.

Diesen Algorithmus können wir mit klassischen Algorithmen vergleichen, wobei wir bemerken müssen, dass jeder klassische Algorithmus  $f$  im schlimmsten Fall mindestens  $\frac{2^n}{2} + 1$  mal evaluieren muss. Angenommen, wir kennen die Funktion auf der Hälfte ihrer Eingabe-Bitstrings (also  $2^{n-1}$  Eingaben) und erhalten in jedem Fall die gleiche Ausgabe. Dann können wir immer noch keine Aussage darüber treffen, ob die Funktion konstant ist, da die Funktion auf der anderen Hälfte der Eingabe-Bitstrings immer das andere Ergebnis ausgibt und somit balanced ist. Das würde also dem Worst-Case (engl. für "schlimmsten Fall") entsprechen. In diesem Szenario hätten wir  $2^{n-1}$  Fragen verschwendet, ohne dabei etwas sinnvolles herauszufinden. Um sicher zu gehen, ob  $f$  konstant oder balanced ist, müssen wir die Funktion auf einer weiteren Eingabe evaluieren. Das ergibt insgesamt  $\frac{2^n}{2} + 1$  Auswertungen von  $f$  im Vergleich zu 1 Auswertung im Quantenalgorithmus. Bemerke, dass der Unterschied auch für recht kleine Werte wie  $n = 100$  schon so dramatisch ist, dass man  $f$  nicht mit einem vertretbaren Zeitaufwand so häufig auswerten könnte (tatsächlich würde vorher die Sonne vollständig ausbrennen und wir müssten in ein neues Sonnensystem gezogen sein).

Zusammengefasst: Wenn  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  eine Funktion ist, die entweder konstant oder balanced ist, dann kann der Deutsch-Jozsa-Algorithmus mit *nur einer Auswertung* von  $f$  feststellen, welche der beiden Eigenschaften zutrifft. Das ist exponentiell schneller als jeder klassische Algorithmus, welcher im schlimmsten Fall die Funktion auf  $\frac{2^n}{2} + 1$  Eingaben auswerten müsste.

### Hausaufgabe 5.2: Deutsch-Jozsa ausführen

Die gelbe **Oracle** Box in QUIRKY implementiert das Phasen-Orakel für eine Funktion, die entweder konstant oder balanced ist. Implementiere den Deutsch-Jozsa-Algorithmus in QUIRKY und finde damit heraus, welcher der beiden Fälle zutrifft.

#### 5.2.4 Bernstein-Vazirani-Algorithmus

In den vorigen Abschnitten haben wir mithilfe der Hadamard-Transformation interessante Probleme gelöst. Gegeben einer Funktion  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  und dem Versprechen, dass  $f$  entweder konstant oder balanced ist, kann der Deutsch-Jozsa-Algorithmus herausfinden, welche der beiden Fälle zutrifft.

In diesem Abschnitt werden wir ein anderes interessantes Problem besprechen, welches mit einer kleinen Veränderung des selben Algorithmus gelöst werden kann. Wie auch zuvor starten wir mit einem Versprechen über eine unbekannte Funktion  $f$ . In diesem Fall werden wir aber nicht annehmen, dass die Funktion entweder konstant oder balanced ist, stattdessen nehmen wir, an dass sie die folgende spezielle Form hat:

$$f(x_1, \dots, x_n) = x_1 a_1 \oplus \dots \oplus x_n a_n, \quad (5.14)$$

wobei  $a_1, \dots, a_n \in \{0, 1\}$  ein fester Bitstring ist, der die Funktion definiert.

Wenn  $n = 1$ , gibt es nur zwei solche Funktionen:

- $f(x_1) = 0$  wenn  $a_1 = 0$  sowie
- $f(x_1) = x_1$  wenn  $a_1 = 1$ .

Wenn  $n = 2$ , dann gibt es schon vier verschiedene solcher Funktionen:

- $f(x_1, x_2) = 0$ , wenn  $[a_1, a_2] = [0, 0]$ ,
- $f(x_1, x_2) = x_2$ , wenn  $[a_1, a_2] = [0, 1]$ ,

- 
- 

Beachte, dass jede dieser Funktionen eine Summe modulo Zwei über eine *Teilmenge* der Variablen  $x_i$  berechnet. Welche Teilmenge? Wenn  $a_i = 1$  gilt, dann ist die Variable  $x_i$  in der Teilmenge enthalten.

Im Allgemeinen gibt es  $2^n$  mögliche Bitstrings  $a_1, \dots, a_n$  und somit  $2^n$  Funktionen  $f$  der speziellen Form (5.14). Tatsächlich können wir uns Gl. (5.14) als Methode zum Verstecken des Bitstrings in der Funktion  $f$  vorstellen. Wie oft muss man die Funktion auswerten um den Bitstring offenzulegen? Da

$$\begin{aligned} a_1 &= f(1, 0, \dots, 0, 0), \\ a_2 &= f(0, 1, \dots, 0, 0), \\ &\vdots \\ a_n &= f(0, 0, \dots, 0, 1), \end{aligned}$$

können wir schlussfolgern, dass  $n$  Auswertungen von  $f$  auf jeden Fall ausreichen. Für jeden klassischen Algorithmus entspricht das auch dem Optimum: für jede Auswertung lernt man ein Bit an Information. Da die unbekannt Bits  $a_1, \dots, a_n$  komplett unabhängig und beliebig sind und es genau  $n$  gibt, muss man die Funktion mindestens  $n$ -mal auswerten um alle Bits zu lernen.

Gleich werden wir sehen, dass das Problem mit einem Quantenalgorithmus viel besser gelöst werden kann. Zu Beginn berechnen wir erst einmal, wie sich das Phasen-Orakel für die Funktion aus Gl. (5.14) auf Basiszustände auswirkt:

$$\begin{aligned} O_f |x_1, \dots, x_n\rangle &= (-1)^{x_1 a_1 \oplus \dots \oplus x_n a_n} |x_1, \dots, x_n\rangle \\ &= (-1)^{x_1 a_1 + \dots + x_n a_n} |x_1, \dots, x_n\rangle. \end{aligned} \quad (5.15)$$

Im zweiten Schritt haben wir ausgenutzt, dass  $(-1)^a$  nur von  $a$  modulo Zwei (also ob  $a$  gerade oder ungerade ist) abhängt, weswegen es keinen Unterschied macht, ob wir die Addition modulo 2 ( $\oplus$ ) oder die normale Addition benutzen. Wie können wir dieses Phasen-Orakel implementiert werden? Eigentlich interessiert uns das nicht, da der Algorithmus das Orakel wie eine Black-Box benutzen wird. Aber da es eine schöne Übung ist, kannst du das in der folgenden Übungsaufgabe herausfinden.

### Übungsaufgabe 5.5: Das Phasen-Orakel implementieren (Optional)

In dieser Aufgabe kannst du das Phasen-Orakel für Funktionen der Form (5.14) implementieren.

1. Wenn  $n = 2$  ist, gibt es vier solcher Funktionen, wie oben besprochen. Kannst du für jede von ihnen einen QUIRKY-Schaltkreis bauen, der die Phasen-Orakel implementiert?
2. Erkläre in Worten oder mit Bildern, wie man das Phasen-Orakel im Allgemeinen bauen kann (also wenn  $n \geq 1$  und die Bits  $a_1, \dots, a_n \in \{0, 1\}$  unbekannt sind).

Nun stellen wir den **Bernstein-Vazirani-Algorithmus** vor, welcher den versteckten Bitstring  $[a_1, \dots, a_n]$  mit nur eines Aufrufs des Phasen-Orakels für  $f$  aufdecken kann:

1. Starte mit  $|0 \dots 0\rangle$ .
2. Wende die Hadamard-Transformation  $H \otimes \dots \otimes H$  an.

3. Wende das Phasen-Orakel  $O_f$  der Funktion  $f$  an.
4. Wende wieder die Hadamard-Transformation  $H \otimes \dots \otimes H$  an.
5. Messe zum Schluss alle Qubits. Das Messergebnis entspricht genau dem Bitstring  $[a_1, \dots, a_n]$ .

Der Algorithmus ist identisch zu dem Deutsch-Jozsa-Algorithmus aus §5.2.3, bis auf den letzten Schritt, welcher noch einfacher ist.

### Hausaufgabe 5.3: Bernstein-Vazirani ausführen

Die orangene **Orakel** Box in QUIRKY implementiert das Phasen-Orakel für eine Funktion der Form (5.14) mit  $n = 4$ . Implementiere den Bernstein-Vazirani-Algorithmus in QUIRKY und nutze ihn um den versteckten Bitstring  $[a_1, a_2, a_3, a_4]$  aufzudecken.

Warum funktioniert dieser Algorithmus? Jetzt bist du mit der Analyse dran!

### Übungsaufgabe 5.6: Bernstein-Vazirani überprüfen

Die Funktion  $f(x_1, x_2) = x_2$  entspricht dem Bitstring  $[a_1, a_2] = [0, 1]$ , wie wir oben gesehen hatten.

Zeige, dass bei dem Ausführen des Bernstein-Vazirani-Algorithmus auf dieser Funktion das Messergebnis tatsächlich immer  $[a_1, a_2] = [0, 1]$  ist. Überprüfe das nicht einfach mit QUIRKY sondern schreibe den Zustand nach jedem Schritt selbst auf.

In der folgenden Hausaufgabe kannst du den allgemeinen Fall analysieren:

### Hausaufgabe 5.4:

Zeige, dass das Ausführen von Bernstein-Vazirani auf einer Funktion der Form (5.14) mit 100% Wahrscheinlichkeit zu dem Messergebnis  $[a_1, \dots, a_n]$  führt.

**Hinweis:** Da die ersten vier Schritte des Algorithmus identisch zu dem Deutsch-Jozsa-Algorithmus sind, ist der Zustand direkt vor der Messung gegeben in Gl. (5.13).

## 5.3 Suchen mit Grover

Nachdem Hila und Iman wieder sicher auf der Erde gelandet sind, wurden sie gute Freunde von Alice und Bob. Die vier entscheiden sich, Silvester zusammen zu verbringen. Zufällig ist am selben Tag auch die Ziehung des jährlichen Quanten-Lottos! Sie wissen, dass nur eine Person der vier den großen Gewinn absahnen kann, aber wer wird es sein? Wenn wir unsere vier Hauptfiguren mit Bitstrings benennen, zum Beispiel so,

	$x_1$	$x_2$
Alice	0	0
Bob	0	1
Hila	1	0
Iman	1	1

dann können wir die Lotterie durch eine Funktion  $f: \{0, 1\}^2 \rightarrow \{0, 1\}$  modellieren, die nur für die gewinnende Person 1 ergibt. Wenn zum Beispiel  $f(1, 0) = 1$  ist, dann hat Hila dieses Jahr gewonnen.

Wie können unsere Freunde herausfinden, wer gewonnen hat? Mit einem klassischen Algorithmus müssen sie die Funktion bis zu drei mal ausführen um herauszufinden, wer ausgelost wurde. Angenommen, sie wüssten, dass  $f(0, 1) = 0$  und  $f(1, 0) = 0$  – dann ist immer noch nicht klar ob Alice oder Iman gewonnen haben! Aus längst vergessenen Gründen besagen die Regeln der Lotterie aber, dass die Funktion nur einmal ausgewertet werden darf.

Die vier setzen sich zusammen und fangen an zu überlegen. Nach einer Weile wird Bob ungeduldig und schlägt vor: *“Lasst uns doch einfach den ersten Schritten des Deutsch-Jozsa- und Bernstein-Vazirani-Algorithmus folgen – bestimmt klappt der selbe Trick nochmal...”* Da die anderen keine bessere Idee haben, bereiten sie den folgenden Zustand vor

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

Anschließend geben sie diesen Zustand der Quanten-Lotterie, welche das Phasen-Orakel  $O_f$  anwendet und den Zustand zurück gibt. Seien  $a$  und  $b$  die beiden Bits, die die gewinnende Person festlegen, also  $f(a, b) = 1$  und alle anderen Funktionswerte sind Null. Dann erhalten sie von der Quanten-Lotterie den folgenden Zwei-Qubit-Zustand:

$$\begin{aligned} & \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle - 2|a, b\rangle) \\ &= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) - |a, b\rangle \\ &= |+\rangle \otimes |+\rangle - |a\rangle \otimes |b\rangle, \end{aligned}$$

wobei die  $-2|a, b\rangle$  in der ersten Zeile eins der Plus-Zeichen durch ein Minus ersetzt. Nach dem anwenden einer Hadamard-Transformation erhalten sie den folgenden Zustand:

$$\begin{aligned} & |0\rangle \otimes |0\rangle - H|a\rangle \otimes H|b\rangle \\ &= |00\rangle - \frac{1}{2}(|0\rangle + (-1)^a|1\rangle) \otimes (|0\rangle + (-1)^b|1\rangle) \\ &= -\frac{1}{2}(-|00\rangle + (-1)^a|10\rangle + (-1)^b|01\rangle + (-1)^{a+b}|11\rangle) \end{aligned}$$

Jetzt sind sie verwirrt und wissen nicht ganz weiter. Hila hat eine Idee: *“Mich stört das Minuszeichen vor  $|00\rangle$ . Warum wenden wir nicht eine Quantenoperation an, die so funktioniert?”*

$$\begin{aligned} |00\rangle &\mapsto -|00\rangle \\ |01\rangle &\mapsto |01\rangle \\ |10\rangle &\mapsto |10\rangle \\ |11\rangle &\mapsto |11\rangle \end{aligned} \tag{5.16}$$

Iman fällt ein: *“Ich glaube ich weiß wie wir diese Operation mit einer kontrollierten-Z-Operation und ein paar NOT-Operationen bauen könnenn...”* Nach kurzer Zeit erhält die Gruppe also den folgenden Zustand:

$$-\frac{1}{2}(|00\rangle + (-1)^a|10\rangle + (-1)^b|01\rangle + (-1)^{a+b}|11\rangle)$$

Nach einem kurzen Moment bemerkt Alice: *“Den Zwei-Qubit-Zustand können wir als Tensorprodukt schreiben!”*

$$-\frac{1}{2}(|0\rangle + (-1)^a|1\rangle) \otimes (|0\rangle + (-1)^b|1\rangle) = -(H \otimes H)|a, b\rangle.$$

Bob freut sich riesig: *“Aha! Wir müssen nur noch eine weitere Hadamard-Transformation anwenden und können dann beide Qubits messen...”* Kannst du herausfinden, wer die diesjährige Quanten-Lotterie gewinnen wird?

### Hausaufgabe 5.5: Quanten-Lotterie

1. Schreibe die Quantenoperation aus Gl. (5.16) mit einer kontrollierten-Z-Operation und ein paar NOT-Operationen.

**Hinweis:** Erinnerung dich, dass die kontrollierte-Z-Operation  $CZ_{1 \rightarrow 2}$  auf einem Basiszustand  $|x, y\rangle$  sich wie folgt verhält: Wenn  $x = 0$  tut sie nichts. Wenn  $x = 1$ , verhält sie sich wie Z auf dem zweiten Qubit. Letzte Woche hast du gelernt, sie in QUIRKY zu bauen.

2. Baue den Quantenalgorithmus, den Alice, Bob, Hila und Iman sich ausgedacht haben in QUIRKY und finde heraus, wer dieses Jahr die Quanten-Lotterie gewinnt.

Der Quantenalgorithmus, den die Gruppe entdeckt haben ist ein Spezialfall von **Grovers Algorithmus**. Grovers Algorithmus löst das folgende Problem: Gegeben eines Orakels für die Funktion  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  findet er  $x_1, \dots, x_n \in \{0, 1\}$ , sodass  $f(x_1, \dots, x_n) = 1$ . Wir können uns das Problem vorstellen wie das finden einer gewinnenden Person aus  $2^n$  Teilnehmenden, oder anders gesagt, ein Element, welches eine bestimmte Eigenschaft besitzt aus einer ungeordneten Datenbank (womit wir meinen, dass die Elemente in der Datenbank nicht sortiert sind) zu finden. Mit dem selben Argument wie zuvor, muss jeder klassische Algorithmus schlimmstenfalls alle bis auf einen der  $2^n$  Einträge anschauen, bevor er fertig ist (durchschnittlich muss die Hälfte der Einträge angeschaut werden). Im Gegensatz dazu braucht Grovers Algorithmus nur proportional zu  $\sqrt{2^n}$  viele Abfragen, was langsamer wächst in  $n$ ! Grovers Algorithmus ist ein sehr vielseitiges Werkzeug, welches ein Quadratwurzel-Speedup für viele Berechnungsprobleme liefert.

#### 5.3.1 Winkelverstärkung

Der letzte Quantenalgorithmus den wir besprechen werden ist ein sehr wichtiges Unterprogramm, welches von vielen anderen Quantenalgorithmen genutzt wird (zum Beispiel ist es das Herz von Grovers Algorithmus für Funktionen mit mehr als  $n = 2$  Eingabebits).

Das Problem, das das Unterprogramm löst mag auf den ersten Blick seltsam wirken. Tatsächlich ist es ein reines Quantenproblem in dem Sinne, dass es kein sinnvolles klassisches Äquivalent gibt. Trotzdem taucht es auf natürliche Art in anderen Algorithmen auf, was das Unterprogramm nützlich macht. Das unterstreicht noch einmal, wie anders die Ideen hinter Quantenalgorithmen sind und das neuartige Ansätze für das Erfinden neuer Quantenalgorithmen notwendig sind!

Weiterhin ist das Problem ein Beispiel, wo mehr als eine Orakel-Anfrage notwendig sind, um zu einer Lösung zu gelangen. Das ist ein Unterschied zu den Quantenalgorithmen, die wir bisher in dieser Quest kennengelernt haben, welche das Orakel nur genau ein mal benötigen.

### Hausaufgabe 5.6: Mit welchem Winkel? (schwierig)

Alice und Bob sind verzweifelt. "Es tut uns leid," erzählen sie dir. "Wir haben diese schöne Reflektion  $V(\theta)$  mit dem Winkel

$$\theta = +\frac{\pi}{4k} \quad \text{oder} \quad \theta = -\frac{\pi}{4k},$$

gebaut, aber wir können uns einfach nicht mehr erinnern welcher von beiden es war – wir wissen nur noch, dass  $k \geq 1$  war!"<sup>a</sup> Was das Problem noch schlimmer macht: Das Gatter wird sich selbst zerstören, wenn es mehr als  $k$  mal benutzt wird.

Kannst du ihnen helfen? Deine Aufgabe, wenn du sie annimmst, ist es, herauszufinden, welcher der beiden Winkel der richtige ist, indem du das Gatter  $V(\theta)$  maximal  $k$  mal

benutzt.

**Hinweis:** Zwei aufeinanderfolgende Reflexionen ergeben eine Rotation. Was erhältst du, wenn du NOT und  $V(\theta)$  kombinierst? (Du brauchst die allgemeine Formel aus Übungsaufgabe 4.6 nicht um diese Frage zu beantworten).

<sup>a</sup>Wenn du möchtest, kannst du dir die Reflektion wie ein Orakel vorstellen, das einen der beiden Winkel seltsam versteckt.

## 5.4 Deine Quanten-Reise

Mit diesem letzten Problem ist *The Quantum Quest* abgeschlossen. Wow, so schnell sind fünf Wochen vorbeigeflogen – dieser Kurs war eine ganz schöne Reise! Wir hoffen wirklich, dass du in den letzten Wochen viel Spaß hattest und viel Interessantes über Mathematik gelernt hast.

Wenn du nicht genug von der Quanteninformatik bekommen kannst, brauchst du nicht zu verzweifeln. Mittlerweile bist du schon ein:e erfahrene:r Magier:in und kannst genug um ein fortgeschrittenes Buch selbst zu studieren. Warum schaust du nicht mal, ob die nächste Bücherei nicht eine Kopie des Buchs '*Quantum Computation and Quantum Information*' von Michael Nielsen und Isaac Chuang hat?



## 5.5 Lösungen der Übungsaufgaben

### Lösung Übungsaufgabe 5.1

Die anderen drei Funktionen sind  $f = \text{NOT}$  sowie die zwei konstanten Funktionen  $f(0) = f(1) = 0$  und  $f(0) = f(1) = 1$ .

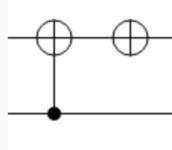
- Für die NOT-Funktion gilt:

$$U_{\text{NOT}} |a, b\rangle = |a, b \oplus \text{NOT}(a)\rangle = |a, b \oplus a \oplus 1\rangle = |a, \text{NOT}(b \oplus a)\rangle,$$

was wie als Zusammensetzung einer kontrollierten-NOT-Operation und einer NOT-Operation auf dem zweiten Qubit geschrieben werden kann, also,

$$U_{\text{NOT}} = (I \otimes \text{NOT}) \text{CNOT}_{1 \rightarrow 2}.$$

In QUIRKY sieht das dann so aus:



- Für die Alles-Null-Funktion  $f(0) = f(1) = 0$  gilt:

$$U_f |a, b\rangle = |a, b\rangle,$$

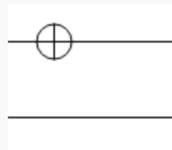
also müssen wir gar nichts machen:



- Für die Alles-Eins-Funktion  $f(0) = f(1) = 1$  gilt:

$$U_f |a, b\rangle = |a, b \oplus 1\rangle = |a, \text{NOT}(b)\rangle,$$

also müssen wir nur das zweite Qubit invertieren:



## Lösung Übungsaufgabe 5.2

Es gibt vier Funktionen: die 'Identitäts'-Funktion  $f(x) = x$ , die NOT-Funktion, die Alles-Null-Funktion und die Alles-Eins-Funktion.

- Für die Identitäts-Funktion  $f(x) = x$  kann Gl. (5.6) gelesen werden als

$$O_f |x\rangle = (-1)^x |x\rangle,$$

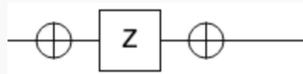
also ist das genau das Z-Gatter:



- Für die NOT-Funktion  $f(x) = \text{NOT}(x)$  soll gelten, dass

$$O_f |x\rangle = (-1)^{\text{NOT}(x)} |x\rangle = \text{NOT Z NOT } |x\rangle,$$

was folgender Reihe an Operationen entspricht:



- Für die Alles-Null-Funktion  $f(0) = f(1) = 0$  gilt:

$$O_f |x\rangle = |x\rangle,$$

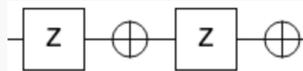
daher müssen wir nichts tun:



- Für die Alles-Eins-Funktion  $f(0) = f(1) = 1$  gilt:

$$O_f |x\rangle = -|x\rangle,$$

was wir erreichen, indem wir die ersten beiden Orakel nacheinander schalten:



Tatsächlich fügt das erste Orakel ein Minuszeichen hinzu, wenn  $x = 1$  ist, während das zweite Orakel ein Minuszeichen hinzuzufügt, wenn  $x = 0$  ist, also erhalten wir immer eines:

$$\text{NOT Z NOT Z } |0\rangle = \text{NOT Z NOT } |0\rangle = -|0\rangle,$$

$$\text{NOT Z NOT Z } |1\rangle = \text{NOT Z NOT}(-|1\rangle) = -\text{NOT Z NOT } |1\rangle = -|1\rangle.$$

Im vorletzten Schritt haben wir dabei die Linearität ausgenutzt, um das Minuszeichen nach vorne zu bringen.

### Lösung Übungsaufgabe 5.3

Wir evaluieren Gl. (5.7) für alle vier Funktionen.

- Für  $f(x) = x$ :

$$HO_f |+\rangle = \frac{1+(-1)}{2} |0\rangle + \frac{1-(-1)}{2} |1\rangle = |1\rangle.$$

- Für  $f(x) = \text{NOT}(x)$ :

$$HO_f |+\rangle = \frac{-1+1}{2} |0\rangle + \frac{-1-1}{2} |1\rangle = -|1\rangle.$$

- Für die Alles-Null-Funktion:

$$HO_f |+\rangle = \frac{1+1}{2} |0\rangle + \frac{1-1}{2} |1\rangle = |0\rangle.$$

- Für die Alles-Eins-Funktion

$$HO_f |+\rangle = \frac{(-1)+(-1)}{2} |0\rangle + \frac{(-1)-(-1)}{2} |1\rangle = -|0\rangle.$$

### Lösung Übungsaufgabe 5.4

1. Hier steht  für  $x_1 y_1$ .

2. Für  $n = 2$ ,

$$\begin{aligned}(H \otimes H) |x_1, x_2\rangle &= (H |x_1\rangle) \otimes (H |x_2\rangle) \\ &= \left( \frac{1}{\sqrt{2}} \sum_{y_1 \in \{0,1\}} (-1)^{x_1 y_1} |y_1\rangle \right) \otimes \left( \frac{1}{\sqrt{2}} \sum_{y_2 \in \{0,1\}} (-1)^{x_2 y_2} |y_2\rangle \right) \\ &= \frac{1}{2} \sum_{y_1, y_2 \in \{0,1\}} (-1)^{x_1 y_1} (-1)^{x_2 y_2} |y_1\rangle \otimes |y_2\rangle \\ &= \frac{1}{2} \sum_{y_1, y_2 \in \{0,1\}} (-1)^{x_1 y_1 + x_2 y_2} |y_1, y_2\rangle,\end{aligned}$$

also steht  für  $x_1 y_1 + x_2 y_2$ .

## Lösung Übungsaufgabe 5.5

1. Das sind die vier Funktionen und ihre Phasen-Orakel:

- Für  $[a_1, a_2] = [0, 0]$ ,  $O_f |x_1, x_2\rangle = |x_1, x_2\rangle$ , also macht das Phasen-Orakel nichts.
- Für  $[a_1, a_2] = [0, 1]$ ,  $O_f |x_1, x_2\rangle = (-1)^{x_2} |x_1, x_2\rangle$ , was  $I \otimes Z$  entspricht.
- Für  $[a_1, a_2] = [1, 0]$ ,  $O_f |x_1, x_2\rangle = (-1)^{x_1} |x_1, x_2\rangle$ , was das gleiche ist wie  $Z \otimes I$ .
- Für  $[a_1, a_2] = [1, 1]$ ,  $O_f |x_1, x_2\rangle = (-1)^{x_1+x_2} |x_1, x_2\rangle = (-1)^{x_1} (-1)^{x_2} |x_1, x_2\rangle$ , was  $Z \otimes Z$  entspricht.

Es ist klar, wie diese vier Operationen in QUIRKY aussehen.

2. Das allgemeine Muster ist damit klar. Für eine beliebige Funktion der Form (5.14) gilt:

$$O_f |x_1, \dots, x_n\rangle = (-1)^{x_1 a_1 + \dots + x_n a_n} |x_1, \dots, x_n\rangle = (Z^{a_1} \otimes \dots \otimes Z^{a_n}) |x_1, \dots, x_n\rangle,$$

wobei wir  $Z^1 = Z$  und  $Z^0 = I$  schreiben. Mit anderen Worten, wir wenden ein Z-Gatter auf das  $j$ -te Qubit nur genau dann an, wenn  $a_j = 1$  ist.

## Lösung Übungsaufgabe 5.6

In Schritt 1 starten wir mit:

$$|00\rangle$$

In Schritt 2 wenden wir  $H \otimes H$  an und erhalten:

$$\begin{aligned} & \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \\ &= \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) \end{aligned}$$

Im 3. Schritt wenden wir das Phasen-Orakel  $O_f$  für  $f(x_1, x_2) = x_2$  an:

$$\frac{1}{2} (|00\rangle - |01\rangle + |10\rangle - |11\rangle)$$

In Schritt 4 wenden wir wieder  $H \otimes H$  an und erhalten:

$$\begin{aligned} & \frac{1}{2} \left( (H \otimes H) |00\rangle - (H \otimes H) |01\rangle + (H \otimes H) |10\rangle - (H \otimes H) |11\rangle \right) \\ &= \frac{1}{4} \left( (|00\rangle + |01\rangle + |10\rangle + |11\rangle) - (|00\rangle - |01\rangle + |10\rangle - |11\rangle) \right. \\ & \quad \left. + (|00\rangle + |01\rangle - |10\rangle - |11\rangle) - (|00\rangle - |01\rangle - |10\rangle + |11\rangle) \right) \\ &= |01\rangle. \end{aligned}$$

In Schritt 5 messen wir schlussendlich beide Qubits und erhalten immer das Ergebnis  $[0, 1]$ .