LiFi Project

Vorlesungsskript Digitalisierung und Programmierung

Prof. Dr. Nicolas Meseth

Invalid Date

Inhaltsverzeichnis

1 Problemlösung **2**
1.1 Eingabe-Verarbeitung-Ausgabe . . . 2
1.2 Strategien . . . 2
1.2.1 Divide and Conquer . . . 2
1.2.2 Distribute and Parallelize . . . 2

2 Algorithmen **2**
2.1 Suchen . . . 4
2.2 Sortieren . . . 4
2.3 Optimieren . . . 4

3 Informationsrepräsentation **4**
3.1 Information in der Informatik . . . 4
3.1.1 Zahlenraten . . . 4
3.1.2 Bit für Bit . . . 5
3.1.3 Unsicherheit . . . 8
3.1.4 Information . . . 10
3.1.5 Unwahrscheinliche Antworten . . . 10
3.1.6 Mehr als zwei Antworten . . . 12
3.2 Zahlensysteme . . . 13
3.3 Codesysteme . . . 13
3.4 Speicher . . . 13

4 Informationsverarbeitung **13**
4.1 Logik . . . 13
4.2 Substratunabhängigkeit . . . 13
4.3 Addition . . . 13
4.3.1 Binäre Addition . . . 13
4.4 Subtraktion . . . 13

4.5 Andere Informationsverarbeitung 13

5 Informationsübermittlung 13
5.1 Signale 13
5.2 Protokolle 13
5.2.1 Peer-To-Peer 13
5.2.2 Handshake 13
5.2.3 TCP/IP & HTTPs 13
5.3 Kompression 13
5.3.1 Verlustfreie Kompression 13
5.3.2 Verlustbehaftete Kompression 13
5.4 Sicherheit 13
5.4.1 Verschlüsselung 13
5.4.2 Digitale Signaturen 14

1 Problemlösung

1.1 Eingabe-Verarbeitung-Ausgabe

1.2 Strategien

1.2.1 Divide and Conquer

1.2.2 Distribute and Parallelize

2 Algorithmen

Der Begriff "**Algorithmus**" stammt vom Namen des persischen Mathematikers Muhammad al-Khwarizmi, der um das Jahr 780 n. Chr. geboren wurde. Al-Khwarizmi war ein bedeutender Gelehrter am Hofe des Kalifen al-Mamun und verfasste dort Schriften, die den Gebrauch der indischen Zahlzeichen erklärten. Diese Schriften wurden im 12. Jahrhundert ins Lateinische übersetzt, wobei der Titel "Algoritmi de numero Indorum" verwendet wurde. Im Laufe der Zeit wurde der Name al-Khwarizmi zur Bezeichnung für die von ihm beschriebenen Rechenverfahren und entwickelte sich schließlich zum modernen Begriff "Algorithmus".

Heute bezeichnet ein **Algorithmus** eine präzise Abfolge von Anweisungen, die ein bestimmtes Problem lösen oder eine Aufgabe erfüllen sollen. Im Alltag begegnen uns Algorithmen ständig, oft, ohne dass wir es merken: beim Kochen, bei der Wegbeschreibung oder beim Aufbau eines IKEA-Regals.

Um besser zu verstehen, was ein Algorithmus ist, grenzen wir die Begriffe Algorithmus, Programm und Prozess voneinander ab:

- **Algorithmus**: Ein allgemeiner Plan oder eine Methode zur Problemlösung, bestehend aus einer endlichen Sequenz von Anweisungen.

- **Programm**: Die konkrete Umsetzung eines oder mehrerer Algorithmen in einer Programmiersprache, die ein Computer ausführen kann.
- **Prozess**: Die Ausführung eines Programms durch einen Computer.

Algorithmen haben also nicht zwangsläufig etwas mit Computern zu tun. Sie beschreiben lediglich, wie man ein Problem lösen kann. Ob das ein Computer macht oder ein Mensch ist dabei offen. Überlege etwa, wie du eine IKEA-Aufbauanleitung befolgst oder wie du ein Rezept in der Küche kochst. Diese alltäglichen Abläufe folgen alle Algorithmen, ohne dass ein Computer beteiligt ist:

- Kochen: Ein Rezept ist ein Algorithmus für die Zubereitung eines Gerichts.
- Wegbeschreibung: Eine Schritt-für-Schritt-Anleitung, um von Punkt A nach Punkt B zu gelangen.
- Bastelanleitung: Die Anweisungen, um ein Modellflugzeug zusammenzubauen.

Manche Algorithmen können wir einem Computer beibringen und ihn diesen ausführen lassen. Dazu müssen wir eine präzise Übersetzung des Algorithmus in eine Programmiersprache erstellen, die der Computer versteht und ausführen kann. Wir sprechen dann vom Programmieren und das Ergebnis ist ein Programm.

Algorithmen lassen sich basierend auf ihrer Funktion und Anwendung in verschiedene Kategorien einteilen. Hier sind einige wichtige Klassen von Algorithmen:

- Suchalgorithmen
- Sortieralgorithmen
- Optimierungsalgorithmen
- Graphenalgorithmen
- Verschlüsselungsalgorithmen
- Maschinelle Lernalgorithmen

Die oben aufgeführten Klassen sind nicht vollständig und nicht gegenseitig ausschließend. Das bedeutet, dass die Klassen Überlappungen aufweisen und sich manche Algorithmen zwei oder mehreren Klassen zuordnen lassen. So ist der Dijkstra-Algorithmus für das Finden des kürzesten Weges zwischen zwei Orten ein Graphenalgorithmus, weil er auf einer Datenstruktur arbeitet, die wir als Graphen bezeichnen. Gleichzeitig ist er ein Optimierungsalgorithmus, weil er eine optimale Lösung für ein Problem ermittelt: den kürzesten Weg aus der Menge aller möglichen Wege zu finden.

2.1 Suchen

2.2 Sortieren

2.3 Optimieren

3 Informationsrepräsentation

3.1 Information in der Informatik

Hast du dich jemals gefragt, was eigentlich Information genau ist? Jeder hat eine intuitive Vorstellung davon, was wir mit Information meinen. Aber was ist die genaue Definition? Und wie ist der Begriff *Information* im Zusammenhang mit digtialen Computern gemeint?

Hier schon einmal eine Definition, die wir in diesem Kapitel anhand von einigen Beispielen genauer verstehen wollen.

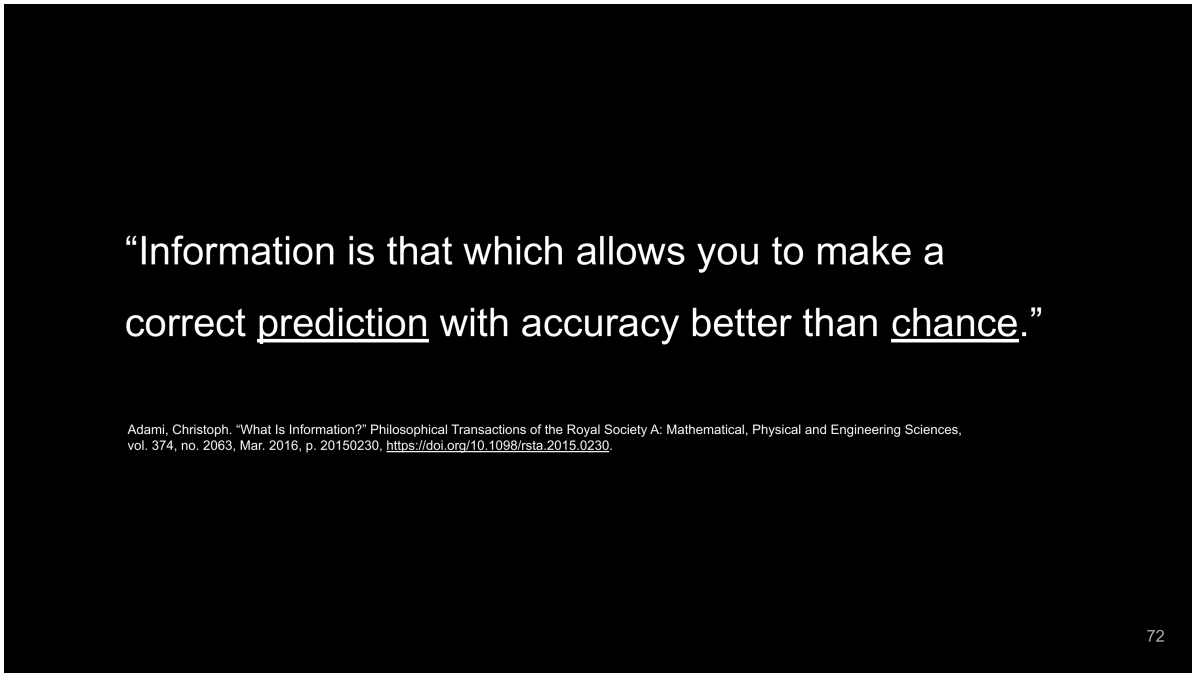

Abbildung 1: Image

3.1.1 Zahlenraten

Um der Information auf die Spur zu kommen beginnen wir mit einem Gedankenexperiment. Stell dir vor, wir spielen ein Zahlenratespiel. Ich denke an eine Zahl zwischen 1 und 16, und dein Ziel ist es, sie zu erraten. Der Haken ist, dass du nur einen Versuch hast, die richtige Zahl zu erraten, aber du kannst die Möglichkeiten vorher durch Fragen der Form “Ist die Zahl

größer als X?" eingrenzen. Für jede Frage werde ich dir mit "ja" oder "nein" antworten und damit die verbleibenden Optionen reduzieren.

Mit jeder Antwort, die du erhältst, wächst dein Wissen über meine Zahl, was bedeutet, dass deine **Unsicherheit** bezüglich der gesuchten Zahl abnimmt. Du kannst einige mögliche Zahlen ausschließen, wenn du eine neue Antwort erhältst.

Wir halten zunächst fest, dass das Eingrenzen der verbleibenden Möglichkeiten die Unsicherheit reduziert. Wir untersuchen nun, wie diese Idee die Grundlage der Informationstheorie bildet. Durch die Reduzierung der Unsicherheit sammeln wir mit jedem Versuch mehr **Information**. Aber wie viel Information erhältst du mit jeder Antwort? Und wie können wir das messen?

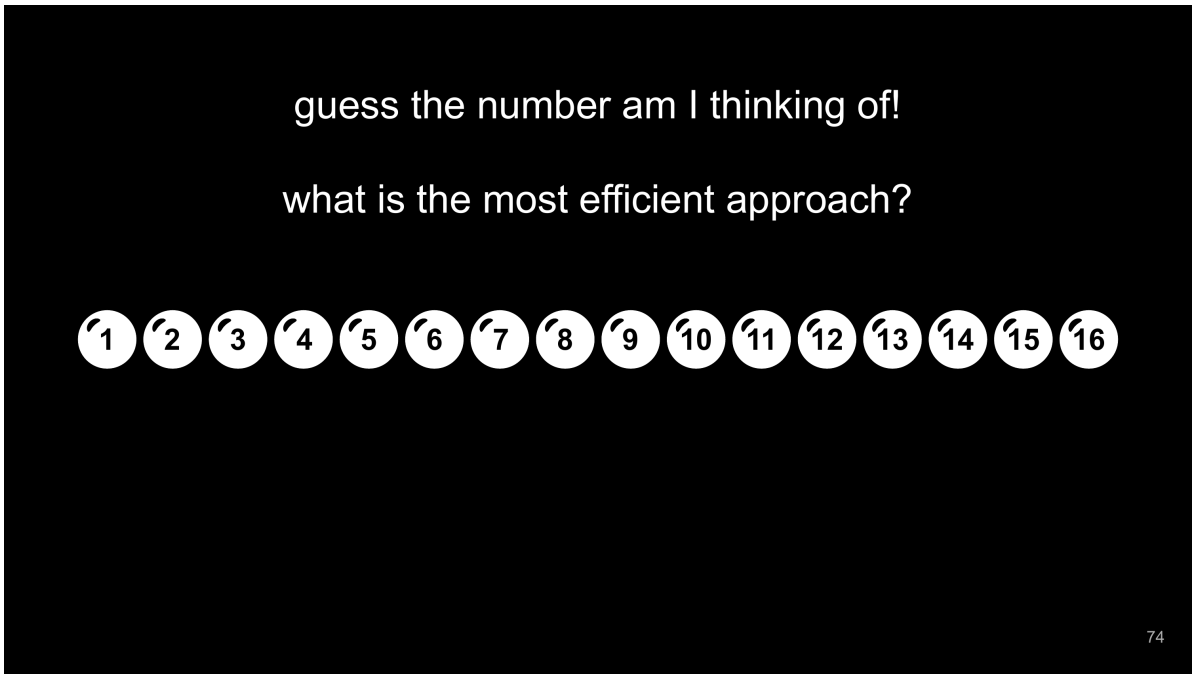

Abbildung 2: Image

3.1.2 Bit für Bit

In der Informatik kann Information definiert werden als "das, was es ermöglicht, eine korrekte Vorhersage mit besserer Genauigkeit als der Zufall zu treffen" [CITE]. Einfacher ausgedrückt bedeutet dies, die Unsicherheit durch Eingrenzung der möglichen Optionen zu reduzieren.

Vor diesem Hintergrund wollen wir unser Zahlenratespiel noch einmal genauer betrachten. Wie oben beschrieben, versuchst du meine Zahl zu erraten, und jede Antwort, die du von mir erhältst, grenzt den Bereich der möglichen Zahlen ein. Diese Reduzierung der Unsicherheit kann mit einer Einheit namens **Bit** gemessen werden. Ein Bit, was für **binary digit** (Binärziffer) steht, ist die grundlegende Einheit der Information. Es zeigt eine Halbierung der Unsicherheit an. Einfacher ausgedrückt: Wenn eine neue Antwort nur noch halb so viele

Optionen wie zuvor übrig lässt, liefert sie uns genau ein Bit an Information.

Allerdings werden nicht alle Fragen und deren Antworten genau ein Bit Information liefern. Wenn zum Beispiel deine erste Frage im Ratespiel lautet: “Ist deine Zahl größer als 12?” und die Antwort “nein” ist, bleiben die Zahlen 1 bis 12 übrig. Das bedeutet, du hast noch 12 Optionen von ursprünglich 16, was die Unsicherheit nicht halbiert. Es werden nur 4 statt der nötigen 8 Möglichkeiten entfernt. Lautet die Antwort dagegen “ja”, so kannst du insgesamt 12 Möglichkeiten streichen, was mehr als der Hälfte entspricht und der Informationsgehalt der Antwort wäre größer als ein Bit.

Um genau ein Bit Information zu erhalten, solltest du darauf abzielen, mit jeder Frage genau die Hälfte der möglichen Zahlen auszuschließen. Wenn du zum Beispiel fragst “Ist deine Zahl größer als 8?”, stellst du sicher, dass du - egal ob die Antwort “ja” oder “nein” ist - in beiden Fällen 8 mögliche Zahlen übrig behältst. Ist die Antwort “ja”, bleiben die Zahlen 9 bis 16 übrig. Ist die Antwort “nein”, bleiben die Zahlen 1 bis 8. In beiden Szenarien wird deine Unsicherheit um die Hälfte reduziert, also um ein Bit.

Indem du deine Fragen sorgfältig so wählst, dass sie die verbleibenden Optionen jedes Mal halbieren, reduzierst du deine Unsicherheit Bit für Bit und machst es dir leichter und schneller, die richtige Zahl zu finden.

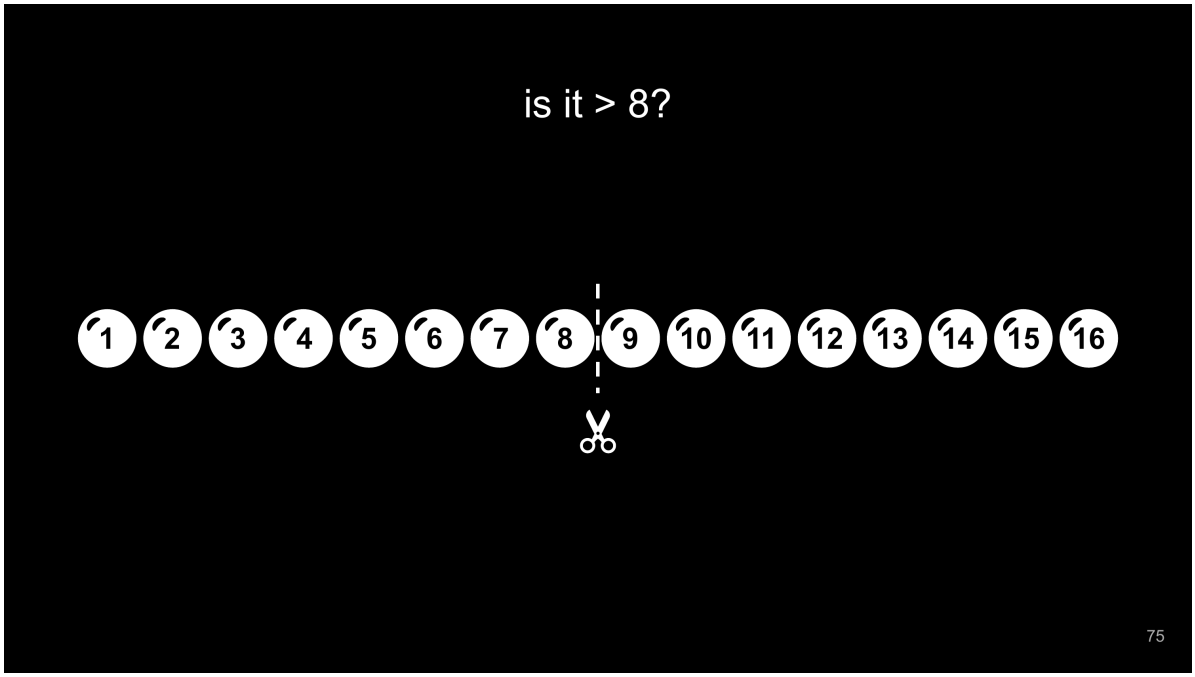

Abbildung 3: Image

Angenommen, die Antwort auf deine erste Frage “Ist deine Zahl größer als 8?” war “nein”. Was sollte deine nächste Frage sein, um die Unsicherheit weiterhin effektiv zu reduzieren? Die beste Strategie ist es zu fragen: “Ist sie größer als 4?”. Diese Vorgehensweise stellt sicher, dass dir immer nur vier mögliche Zahlen bleiben: entweder 1 bis 4, wenn die Antwort “nein” ist,

oder 5 bis 8, wenn die Antwort “ja” ist. Erneut halbiert dies die verbleibenden Optionen und liefert genau ein Bit an Information.

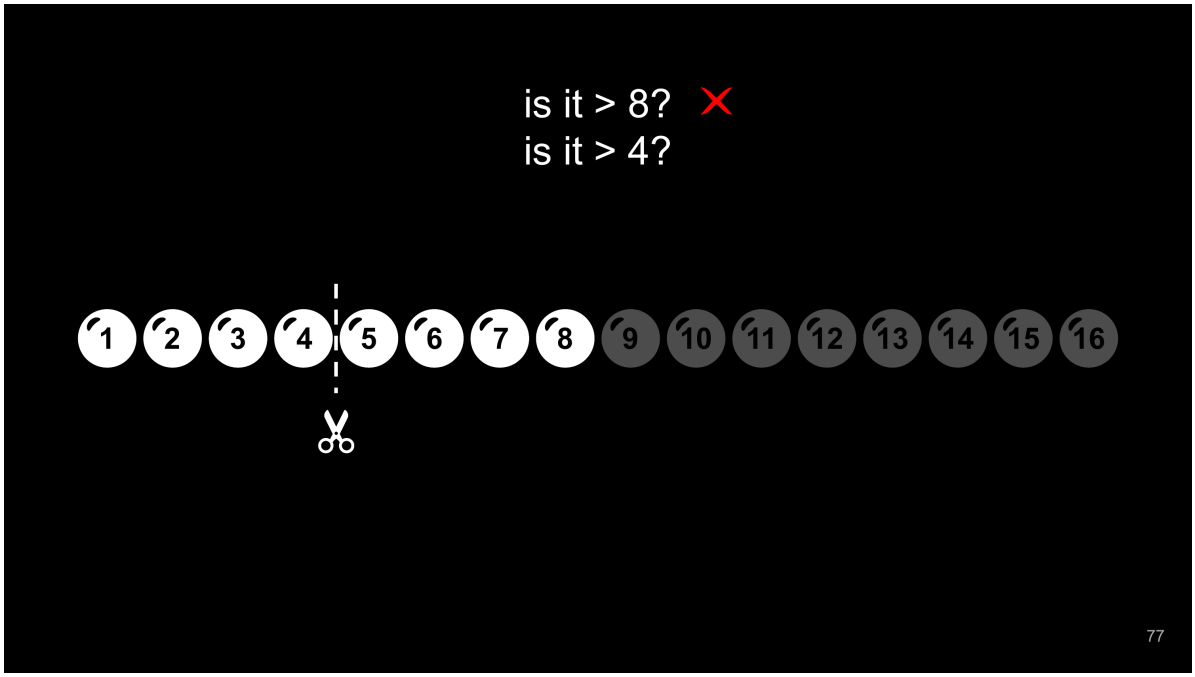

Abbildung 4: Image

Lass uns mit dieser Methode fortfahren. Angenommen, deine zweite Frage “Ist sie größer als 4?” erhält ein “nein” als Antwort. Deine neue Auswahl an Zahlen ist auf 1, 2, 3 und 4 begrenzt. Um die Unsicherheit weiter zu reduzieren, wäre die nächste logische Frage: “Ist sie größer als 2?” Dies lässt dir entweder die Zahlen 1 und 2 oder 3 und 4, abhängig von der Antwort.

Indem du weiterhin Fragen stellst, die systematisch die verbleibenden Optionen halbieren, kannst du sehen, wie wir schrittweise die Unsicherheit Bit für Bit reduzieren. Schließlich wirst du nach nur vier Fragen die Zahl auf genau eine eingrenzen, da keine anderen Möglichkeiten mehr übrig sind. Das bedeutet, du hast die Unsicherheit auf null reduziert.

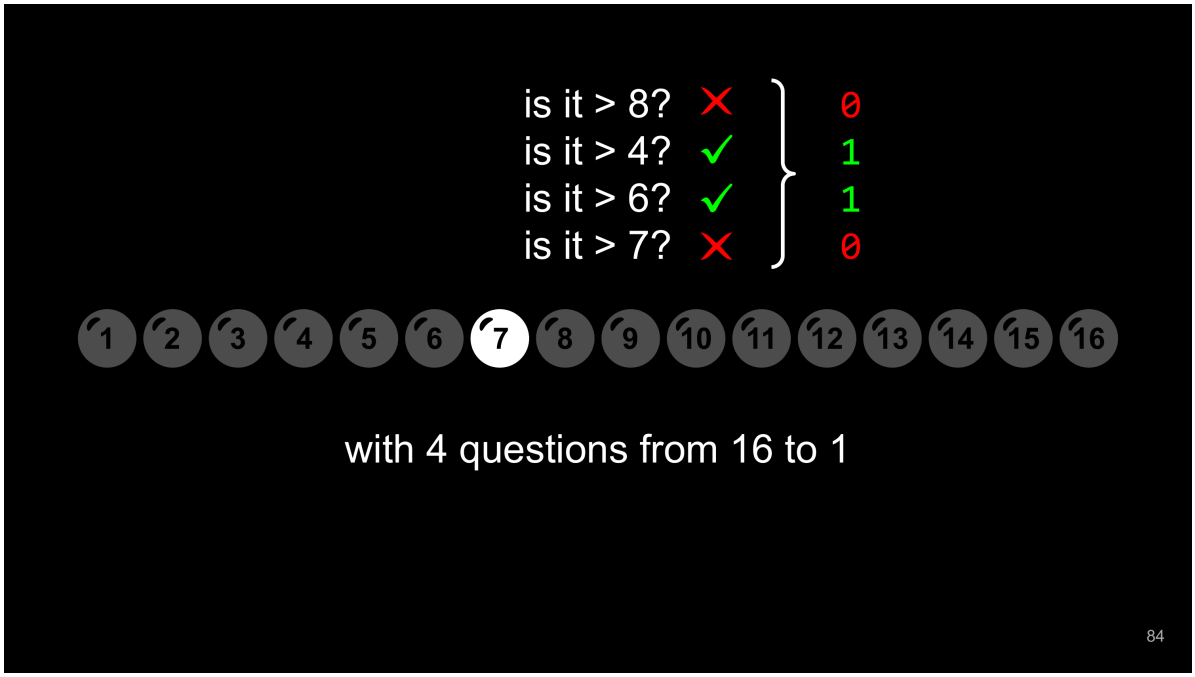

Abbildung 5: Image

3.1.3 Unsicherheit

Wir haben gerade gelernt, dass wir mit jeder Ja/Nein-Frage, die die Hälfte der Möglichkeiten eliminiert, ein Bit an Information gewinnen. Aber wie können wir dieses Konzept der Reduktion von Unsicherheit mathematisch quantifizieren? Wir gehen das Schritt für Schritt durch.

Stell dir vor, du bist am Anfang unseres Zahlenratespiels. Es gibt 16 mögliche Zahlen, an die ich denken könnte, daher ist deine Wahrscheinlichkeit, beim ersten Versuch die richtige Zahl zu erraten, ziemlich gering:

$$P_{correct} = \frac{1}{16}$$

Das entspricht einer Wahrscheinlichkeit von nur 0,0625 - definitiv nicht zu deinen Gunsten. Angenommen, du stellst eine Frage, die die Möglichkeiten auf die Hälfte reduziert. Jetzt verbessert sich deine Chance, richtig zu raten:

$$P_{correct} = \frac{1}{8}$$

Mit jeder weiteren Frage verdoppelt sich diese Wahrscheinlichkeit, während sich deine Unsicherheit halbiert. Ist Wahrscheinlichkeit die beste Methode, um Unsicherheit zu messen? Wenn wir möchten, dass unser Maß für Unsicherheit abnimmt, während wir mehr Informationen sammeln, ist es es sinnvoll, den Kehrwert der Wahrscheinlichkeit zu verwenden. Das entspräche der Anzahl an Möglichkeiten, die noch im Rennen sind.

Zu Beginn gibt es 16 Möglichkeiten, also begänne die Unsicherheit bei 16. Nach der ersten Frage fiele sie auf 8, dann auf 4, 2 und schließlich 1. Allerdings würde dieses Maß uns selbst dann, wenn nur noch eine Option übrig ist, eine gewisse Unsicherheit, nämlich 1, anzeigen, was nicht ganz unserem intuitiven Verständnis entspricht. Wenn wir sicher sind, dann sollte die Unsicherheit 0 sein.

Claude Shannon, der Vater der Informationstheorie, schlug einen anderen Ansatz vor. Er empfahl, die Unsicherheit mithilfe des Logarithmus zur Basis 2 der Anzahl der Möglichkeiten zu messen:

$$H = log_2(N)$$

Diese Methode hat mehrere Vorteile. Erstens ist die Unsicherheit gleich null, wenn es nur eine mögliche Antwort gibt (N=1), was mit unserer Intuition übereinstimmt:

$$log_2(1) = 0$$

Ein weiterer Vorteil ist die Einfachheit der Berechnungen, besonders wenn es um mehrere unabhängige Unsicherheiten geht. Nehmen wir an, das Spiel ändert sich: Jetzt denke ich an zwei unabhängige Zahlen zwischen 1 und 16. Bevor du irgendwelche Fragen stellst, beträgt deine Unsicherheit für jede Zahl:

$$log_2(16) = 4\ bits$$

Die Gesamtunsicherheit für beide Zahlen ist einfach die Summe ihrer individuellen Unsicherheiten:

$$log_2(16) + log_2(16) = 8\ bits$$

Wenn wir stattdessen Wahrscheinlichkeiten verwenden, müssten wir die Chancen multiplizieren, um die Wahrscheinlichkeit zu finden, beide Zahlen korrekt zu erraten:

$$P_{correct,correct} = \frac{1}{16} \times \frac{1}{16} = \frac{1}{256}$$

Mit beiden Zahlen gibt es 256 Möglichkeiten. Mit Shannons Methode erhalten wir das gleiche Ergebnis:

$$log_2(256) = 8\ bits$$

As you can see, using logarithms simplifies our calculations, especially when dealing with multiple sources of uncertainty. This clarity and simplicity are why Shannon's approach has become fundamental in the field of information theory.

Wie du siehst, vereinfacht die Verwendung von Logarithmen unsere Berechnungen, besonders wenn es um mehrere Quellen der Unsicherheit geht. Diese Klarheit und Einfachheit sind der Grund, warum Shannons Ansatz fundamental für die Informationstheorie geworden ist.

3.1.4 Information

Um den Begriff *Information* aus dem Begriff der *Unsicherheit* herzuleiten, betrachten wir zwei Zustände: Die ursprüngliche Unsicherheit vor einer Frage, die wir als

$$H_0$$

bezeichnen, und die reduzierte Unsicherheit nach der Antwort, die wir

$$H_1$$

nennen. Die Menge an Information

$$I$$

, die wir durch die Antwort gewinnen, entspricht der Differenz dieser beiden Unsicherheiten:

$$I = H_0 - H_1$$

Information ist also die Reduzierung von Unsicherheit. Mit der obigen Formel können wir die Information

$$I$$

präzise quantifizieren. Dies ermöglicht uns zu berechnen, wie viel Information wir durch jede Frage gewinnen. Betrachten wir ein konkretes Beispiel aus unserem Zahlenratespiel: Angenommen, deine erste Frage ist "Ist deine Zahl größer als 12?" und die Antwort lautet "nein". Dann bleiben die Zahlen 1 bis 12 als Möglichkeiten übrig. Die gewonnene Information berechnet sich wie folgt:

$$I = log_2(16) - log_2(12) \approx 4 - 3.585 \approx 0.415\ bits$$

In diesem Fall lieferte die Antwort etwa 0,415 Bits an Information – also weniger als ein Bit. Das ist folgerichtig, da die Antwort weniger als die Hälfte der Möglichkeiten eliminiert hat.

Wie sähe es aus, wenn die Antwort "ja" gewesen wäre? In diesem Fall blieben nur die Zahlen 13, 14, 15 und 16 übrig, also vier mögliche Optionen:

$$I = log_2(16) - log_2(4) = 4 - 2 = 2\ bits$$

Diese Antwort lieferte mehr als ein Bit, nämlich 2 Bits. Wie lässt sich dieser Unterschied erklären?

3.1.5 Unwahrscheinliche Antworten

Die Menge an Information hängt von der Wahrscheinlichkeit der jeweiligen Antwort ab. Bei einer Frage, die die Möglichkeiten halbiert, hat jede Antwort "Ja" oder "Nein" auf die Frage "Ist die Zahl größer als X?" eine Wahrscheinlichkeit von genau 50%. Diese gleichmäßige Verteilung vereinfacht die Berechnung und liefert genau ein Bit an Information.

Bei der Frage "Ist deine Zahl größer als 12?" sind die beiden Antworten "Ja" und "Nein" allerdings nicht gleich wahrscheinlich. Es gibt zwölf mögliche Zahlen, die ein "Nein" ergeben

und nur vier, die ein “Ja” ergeben, was Wahrscheinlichkeiten von 75% für “Nein” und 25% für “Ja” bedeutet. Da die “Ja”-Antwort weniger wahrscheinlich ist, ist sie überraschender – und liefert deshalb mehr Information. Wie wir berechnet haben, gibt uns ein “Ja” 2 Bits an Information, während ein “Nein” nur 0,415 Bits liefert.

Dieser Zusammenhang ist ein Kernprinzip der Informationstheorie: Je unwahrscheinlicher ein Ereignis ist, desto mehr Information enthält sein Auftreten. Umgekehrt liefert eine sehr wahrscheinliche Antwort, wie das “Nein” im obigen Beispiel, weniger Information, da sie weniger überraschend ist.

Warum zielen wir darauf ab, Fragen zu stellen, die die Möglichkeiten gleichmäßig halbieren? Man könnte auch wild raten in der Hoffnung, durch eine unwahrscheinlichere Antwort mehr Informationen zu bekommen – auch wenn man seltener richtig liegt. Die Antwort ist einfach: Eine Frage, die den Raum der Möglichkeiten halbiert, maximiert unseren *erwarteten* Informationsgewinn. Zwar können wir mit einer riskanten Frage wie “Ist die Zahl größer 15?” unser Glück herausfordern und im Erfolgsfall die Unsicherheit stark verringern. Dies ist jedoch keine nachhaltige Strategie. Wenn wir das Spiel nicht nur einmal, sondern 10, 100 oder 1000 Mal spielen, nähern wir uns im Mittel dem Erwartungswert für die gewonnene Information

$$E[I]$$

. Und dieser Erwartungswert favorisiert eindeutig Fragen, die die Hälfte der Möglichkeiten eliminieren.

Um dies zu verdeutlichen, berechnen wir zunächst den Erwartungswert für die extreme Frage “Ist die Zahl größer 15?”:

$$E[I] = \left(\frac{1}{16} \times 4\right) + \left(\frac{15}{16} \times 0.09311\right) = 0.3373$$

Der Erwartungswert berechnet sich durch den mit der Wahrscheinlichkeit gewichteten Informationsgewinn für jede mögliche Antwort. Bei der Frage “Ist die Zahl größer als 15?” gibt es ein “Ja” nur dann, wenn die Zahl 16 ist. Von den sechzehn möglichen Zahlen erzeugt also nur eine diese Antwort. Die Wahrscheinlichkeit beträgt somit

$$\frac{1}{16}$$

. In diesem Fall reduziert sich unsere Unsicherheit sofort auf Null, da nur noch eine einzige Möglichkeit übrig bleibt. Die ursprüngliche Unsicherheit betrug 4 Bits:

$$H_0 = log_2(16) = 4\ bits$$

Und wenn die Unsicherheit nach der Antwort “Ja” auf Null reduziert wird:

$$H_1 = log_2(1) = 0\ bits$$

Daraus ergibt sich ein Informationsgehalt von 4 Bits:

$$I = H_0 - H_1 = 4 - 0 = 4\ bits$$

Dies erklärt den ersten Teil der Erwartungswertgleichung. Der zweite Teil folgt demselben Prinzip, allerdings mit einem wichtigen Unterschied: Die Wahrscheinlichkeit für die Antwort "Nein" ist mit

$$\frac{15}{16}$$

deutlich höher. Gleichzeitig verbleibt eine große Restunsicherheit

$$H_1$$

, da wir lediglich eine einzige Zahl ausschließen konnten:

$$H_1 = log_2(15) \approx 3.9069 \; bits$$

Die gewonnene Information ist somit sehr klein:

$$I = 4 - 3.9069 \approx 0.0931 \; bits$$

Anhand der Erwartungswertformel konnten wir zeigen, dass eine extreme Frage, die mit viel Glück direkt zum Ziel führt, in unserem Ratespiel einen erwarteten Informationsgehalt von etwa einem Drittel Bit (0,3373) hat. Prüfen wir nun, ob der Erwartungswert für Fragen, die die Hälfte der Möglichkeiten eliminieren, tatsächlich höher ist:

$$E[I] = (0.5 \times 1) + (0.5 \times 1) = 0.5 + 0.5 = 1 \; bit$$

Wie erwartet beträgt der Erwartungswert genau 1 Bit, da bei jeder Antwort – ob "Ja" oder "Nein" – jeweils die Hälfte der Optionen eliminiert wird.

3.1.6 Mehr als zwei Antworten

Bisher haben wir in unserem Zahlenratespiel nur Fragen mit zwei Antwortmöglichkeiten betrachtet. Informationen, also Antworten auf Fragen, können jedoch vielfältiger sein. Wie berechnen wir die Information, wenn es mehr als zwei Antwortmöglichkeiten gibt?

Als Beispiel betrachten wir eine Urne mit farbigen Kugeln in drei Farben: Rot, Grün und Blau. Aus dieser Urne ziehen wir zwei Kugeln, wobei wir nach jedem Zug die gezogene Kugel zurück in die Urne legen. Dieses statistische Experiment bezeichnet man als "Ziehen mit Zurücklegen".

3.2 Zahlensysteme

3.3 Codesysteme

3.4 Speicher

4 Informationsverarbeitung

4.1 Logik

4.2 Substratunabhängigkeit

4.3 Addition

4.3.1 Binäre Addition

4.4 Subtraktion

4.5 Andere Informationsverarbeitung

5 Informationsübermittlung

5.1 Signale

5.2 Protokolle

5.2.1 Peer-To-Peer

5.2.2 Handshake

5.2.3 TCP/IP & HTTPs

5.3 Kompression

5.3.1 Verlustfreie Kompression

- Huffman

5.3.2 Verlustbehaftete Kompression

- JPEG

5.4 Sicherheit

5.4.1 Verschlüsselung

Caesar Cipher

5.4.2 Digitale Signaturen

- **How is information represented?**
- **How can information be shared?**
- **How can information be processed?**