Aufgabe 4: Nandu

Team-ID: 00988

Team-Name: BitShifter

Bearbeiter dieser Aufgabe:
Filip Zelinskyi

21. November 2023

Inhaltsverzeichnis

1 Abstract

Die Lösung beruht auf der Simulation von Logik-Gattern. Die Suche nach der Antwort beginnt nicht am Anfang bei den Lichtquellen, sondern am Ende bei den Lichtempfängern.

In diesem Dokument wird der Lösungsansatz beschrieben, die Umsetzung auf JavaScript erläutert, die Lösungen für die Beispiele und die interaktive Web-Anwendung als Zusatzaufgabe vorgestellt.

2 Lösungsidee

Jeder Baustein kann als ein Logikgatter dargestellt werden. Der blaue Baustein: $\neg(a \wedge b)$, der rote: $\neg a$ und der blaue einfach als ein Draht, der das Signal nicht beeinflusst. Um das Ergebnis am Empfänger L_i für den aktuellen Zustand der Lichtquellen zu bestimmen, muss festgestellt werden, von welchen Logikgattern und welchen Lichtquellen die Antwort beeinflusst wird. Dadurch kann eine Funktion erstellt werden, die bei Eingabe des Zustands relevanter Quellen sofort die Antwort liefert. Dies wird durch das Durchlaufen der oberhalb liegenden Bausteine vom Lichtempfänger aus erreicht, bis der Zeiger auf Vakuum oder eine Lichtquelle stößt. Eine solche Funktion wird für alle Empfänger auf der Karte erstellt. Die Ausgabe wird für alle möglichen Kombinationen der Lichtquellen ($2^{|Q|}$) berechnet.

3 Umsetzung

Zunächst erfolgt die Initialisierung globaler Variablen zur Verwaltung von Lichtquellen und Lichtempfängern. In diesem Abschnitt werden auch die Funktionen für die Simulation der Logikgatter definiert.

Algorithm 1 Initialisierung globaler Variablen und Funktionen für die Simulation von Logikgattern

```
Require: input[ROW][COL]                          ▷Größe des Feldes aus der Eingabe
  QuelleStatus ← HashMap(name, bool)
  Empfaenger ← HashMap(name, SolveFunction)
  function WeissBaustein(a: bool, b: bool)
      return ¬(a ∧ b)
  end function
  function RotBaustein(a: bool)
      return ¬a
  end function
```

Da es vorkommen kann, dass für einen weißen Baustein beide vorherigen Elemente von Bedeutung sind, benötigen wir 2 x-Werte, an denen sich der Baustein $matrix[row][col]$ befindet. Gleiches gilt, wenn beispielsweise bei einem roten Baustein die Lichtquelle vom Sensor gefunden werden muss. Hier wird ein Zähler $blockIndex$ initialisiert, der modulo 2 berechnet wird, exklusive Vakuum.

Algorithm 2 Suche den x-Wert des geeigneten zweiten Baustein-Teils

```
  function GetBlockPair(matrix[ROW][COL], col, row)
      blockIndex ← 0
      for x ← 0 to col do
          if matrix[row][x] = X then
              blockIndex ← 0
          else
              blockIndex ← (blockIndex + 1)  mod 2
          end if
      end for
      if blockIndex > 0 then
          return col − 1
      else
          return col + 1
      end if
  end function
```

Die Lösungsfunktion sollte rekursiv bis zur Lichtquelle vordringen und eine Sammlung von relevanten Logikgattern, die die Antwort beeinflussen, in Form einer Funktion ausgeben.

```
  function Solve(matrix[ROW][COL], x, y)
      if x < 0  or y < 0  or matrix[y][x] = X then
          return false;
      end if
      cell ← matrix[y][x]
      if cell beginnt mit Q then
          return QuelleStatus[cell]
      else if cell = B then
          return Solve(matrix, x, y − 1)
      else if cell = R then
          return RotBaustein(Solve(matrix, x, y − 1))
      else if cell = r then
          return Solve(matrix, GetBlockPair(matrix, x, y), y − 1)
      else if cell = W then
          return WeissBaustein(Solve(matrix, x, y−1), Solve(matrix, GetBlockPair(matrix, x, y), y−
  1))
      end if
  end function
```

4 Beispiele

Nandu 1

```
cat examples/nandu1.txt | node nandu.js
Q1      Q2      L1      L2
Aus     Aus     Ein     Ein
Ein     Aus     Ein     Ein
Aus     Ein     Ein     Ein
Ein     Ein     Aus     Aus
```

Nandu 2

```
cat examples/nandu2.txt | node nandu.js
Q1      Q2      L1      L2
Aus     Aus     Aus     Ein
Ein     Aus     Aus     Ein
Aus     Ein     Aus     Ein
Ein     Ein     Ein     Aus
```

Nandu 3

```
cat examples/nandu3.txt | node nandu.js
Q1      Q2      Q3      L1      L2      L3      L4
Aus     Aus     Aus     Ein     Aus     Aus     Ein
Ein     Aus     Aus     Aus     Ein     Aus     Ein
Aus     Ein     Aus     Ein     Aus     Ein     Ein
Ein     Ein     Aus     Aus     Ein     Ein     Ein
Aus     Aus     Ein     Ein     Aus     Aus     Aus
Ein     Aus     Ein     Aus     Ein     Aus     Aus
Aus     Ein     Ein     Ein     Aus     Ein     Aus
Ein     Ein     Ein     Aus     Ein     Ein     Aus
```

Nandu 4

```
cat examples/nandu4.txt | node nandu.js
Q1      Q2      Q3      Q4      L1      L2
Aus     Aus     Aus     Aus     Aus     Aus
Ein     Aus     Aus     Aus     Aus     Aus
Aus     Ein     Aus     Aus     Ein     Aus
Ein     Ein     Aus     Aus     Aus     Aus
Aus     Aus     Ein     Aus     Aus     Ein
Ein     Aus     Ein     Aus     Aus     Ein
Aus     Ein     Ein     Aus     Ein     Ein
Ein     Ein     Ein     Aus     Aus     Ein
Aus     Aus     Aus     Ein     Aus     Aus
Ein     Aus     Aus     Ein     Aus     Aus
Aus     Ein     Aus     Ein     Ein     Aus
Ein     Ein     Aus     Ein     Aus     Aus
Aus     Aus     Ein     Ein     Aus     Aus
Ein     Aus     Ein     Ein     Aus     Aus
Aus     Ein     Ein     Ein     Ein     Aus
Ein     Ein     Ein     Ein     Aus     Aus
```

Nandu 5

```
cat examples/nandu5.txt | node nandu.js
Q1      Q2      Q3      Q4      Q5      Q6      L1      L2      L3      L4      L5
Aus     Aus     Aus     Aus     Aus     Aus     Aus     Aus     Aus     Ein     Aus
Ein     Aus     Aus     Aus     Aus     Aus     Ein     Aus     Aus     Ein     Aus
Aus     Ein     Aus     Aus     Aus     Aus     Aus     Aus     Aus     Ein     Aus
Ein     Ein     Aus     Aus     Aus     Aus     Ein     Aus     Aus     Ein     Aus
Aus     Aus     Ein     Aus     Aus     Aus     Aus     Aus     Aus     Ein     Aus
Ein     Aus     Ein     Aus     Aus     Aus     Ein     Aus     Aus     Ein     Aus
Aus     Ein     Ein     Aus     Aus     Aus     Aus     Aus     Aus     Ein     Aus
Ein     Ein     Ein     Aus     Aus     Aus     Ein     Aus     Aus     Ein     Aus
Aus     Aus     Aus     Ein     Aus     Aus     Aus     Aus     Ein     Aus     Aus
Ein     Aus     Aus     Ein     Aus     Aus     Ein     Aus     Ein     Aus     Aus
Aus     Ein     Aus     Ein     Aus     Aus     Aus     Aus     Ein     Aus     Aus
Ein     Ein     Aus     Ein     Aus     Aus     Ein     Aus     Ein     Aus     Aus
Aus     Aus     Ein     Ein     Aus     Aus     Aus     Aus     Ein     Aus     Aus
Ein     Aus     Ein     Ein     Aus     Aus     Ein     Aus     Ein     Aus     Aus
Aus     Ein     Ein     Ein     Aus     Aus     Aus     Aus     Ein     Aus     Aus
Ein     Ein     Ein     Ein     Aus     Aus     Ein     Aus     Ein     Aus     Aus
Aus     Aus     Aus     Aus     Ein     Aus     Aus     Aus     Aus     Ein     Ein
Ein     Aus     Aus     Aus     Ein     Aus     Ein     Aus     Aus     Ein     Ein
Aus     Ein     Aus     Aus     Ein     Aus     Aus     Aus     Aus     Ein     Ein
Ein     Ein     Aus     Aus     Ein     Aus     Ein     Aus     Aus     Ein     Ein
Aus     Aus     Ein     Aus     Ein     Aus     Aus     Aus     Aus     Ein     Ein
Ein     Aus     Ein     Aus     Ein     Aus     Ein     Aus     Aus     Ein     Ein
Aus     Ein     Ein     Aus     Ein     Aus     Aus     Aus     Aus     Ein     Ein
Ein     Ein     Ein     Aus     Ein     Aus     Ein     Aus     Aus     Ein     Ein
Aus     Aus     Aus     Ein     Ein     Aus     Aus     Aus     Aus     Ein     Ein
Ein     Aus     Aus     Ein     Ein     Aus     Ein     Aus     Aus     Ein     Ein
Aus     Ein     Aus     Ein     Ein     Aus     Aus     Aus     Aus     Ein     Ein
Ein     Ein     Aus     Ein     Ein     Aus     Ein     Aus     Aus     Ein     Ein
Aus     Aus     Ein     Ein     Ein     Aus     Aus     Aus     Aus     Ein     Ein
Ein     Aus     Ein     Ein     Ein     Aus     Ein     Aus     Aus     Ein     Ein
Aus     Ein     Ein     Ein     Ein     Aus     Aus     Aus     Aus     Ein     Ein
Ein     Ein     Ein     Ein     Ein     Aus     Ein     Aus     Aus     Ein     Ein
Aus     Aus     Aus     Aus     Aus     Ein     Aus     Aus     Aus     Ein     Aus
Ein     Aus     Aus     Aus     Aus     Ein     Ein     Aus     Aus     Ein     Aus
Aus     Ein     Aus     Aus     Aus     Ein     Aus     Aus     Aus     Ein     Aus
Ein     Ein     Aus     Aus     Aus     Ein     Ein     Aus     Aus     Ein     Aus
Aus     Aus     Ein     Aus     Aus     Ein     Aus     Aus     Aus     Ein     Aus
Ein     Aus     Ein     Aus     Aus     Ein     Ein     Aus     Aus     Ein     Aus
Aus     Ein     Ein     Aus     Aus     Ein     Aus     Aus     Aus     Ein     Aus
Ein     Ein     Ein     Aus     Aus     Ein     Ein     Aus     Aus     Ein     Aus
Aus     Aus     Aus     Ein     Aus     Ein     Aus     Aus     Ein     Aus     Aus
Ein     Aus     Aus     Ein     Aus     Ein     Ein     Aus     Ein     Aus     Aus
Aus     Ein     Aus     Ein     Aus     Ein     Aus     Aus     Ein     Aus     Aus
Ein     Ein     Aus     Ein     Aus     Ein     Ein     Aus     Ein     Aus     Aus
Aus     Aus     Ein     Ein     Aus     Ein     Aus     Aus     Ein     Aus     Aus
Ein     Aus     Ein     Ein     Aus     Ein     Ein     Aus     Ein     Aus     Aus
Aus     Ein     Ein     Ein     Aus     Ein     Aus     Aus     Ein     Aus     Aus
Ein     Ein     Ein     Ein     Aus     Ein     Ein     Aus     Ein     Aus     Aus
Aus     Aus     Aus     Aus     Ein     Ein     Aus     Aus     Aus     Ein     Ein
Ein     Aus     Aus     Aus     Ein     Ein     Ein     Aus     Aus     Ein     Ein
Aus     Ein     Aus     Aus     Ein     Ein     Aus     Aus     Aus     Ein     Ein
Ein     Ein     Aus     Aus     Ein     Ein     Ein     Aus     Aus     Ein     Ein
Aus     Aus     Ein     Aus     Ein     Ein     Aus     Aus     Aus     Ein     Ein
Ein     Aus     Ein     Aus     Ein     Ein     Ein     Aus     Aus     Ein     Ein
Aus     Ein     Ein     Aus     Ein     Ein     Aus     Aus     Aus     Ein     Ein
Ein     Ein     Ein     Aus     Ein     Ein     Ein     Aus     Aus     Ein     Ein
Aus     Aus     Aus     Ein     Ein     Ein     Aus     Aus     Aus     Ein     Ein
Ein     Aus     Aus     Ein     Ein     Ein     Ein     Aus     Aus     Ein     Ein
Aus     Ein     Aus     Ein     Ein     Ein     Aus     Aus     Aus     Ein     Ein
Ein     Ein     Aus     Ein     Ein     Ein     Ein     Aus     Aus     Ein     Ein
Aus     Aus     Ein     Ein     Ein     Ein     Aus     Aus     Aus     Ein     Ein
Ein     Aus     Ein     Ein     Ein     Ein     Ein     Aus     Aus     Ein     Ein
Aus     Ein     Ein     Ein     Ein     Ein     Aus     Aus     Aus     Ein     Ein
Ein     Ein     Ein     Ein     Ein     Ein     Ein     Aus     Aus     Ein     Ein
```

5 Quellcode

```
const w = (a, b) => !(a && b);
const r = (a) => !a;

const QuelleStatus = {};
const Empfaenger = {}

const getBlockPair = (matrix, col, row) => {
  let blockIndex = 0;
  for (let x = 0; x < col; x++) {
    if (matrix[row][x] === 'X') blockIndex = 0;
    else blockIndex = (blockIndex + 1) % 2;
  }
  return blockIndex ? col - 1 : col + 1;
}

const solve = (matrix, x, y) => {
  if (x < 0 || y < 0 || matrix[y][x] === 'X') return false;
  const cell = matrix[y][x];
  if (cell.startsWith("Q")) return QuelleStatus[cell]
  const Actions = {
    B: () => solve(matrix, x, y - 1),
    R: () => r(solve(matrix, x, y - 1)),
    W: () => w(solve(matrix, x, y - 1), solve(matrix, getBlockPair(matrix, x, y), y - 1)),
    r: () => solve(matrix, getBlockPair(matrix, x, y), y),
  }
  return Actions[cell]();
}

process.stdin.on('data', async data => {
  // Bearbeitung der Eingabe wurde gekuerzt
  // Es wurde die Matrix input[y][x] erstellt

  for (let y = row - 1; y >= 0; y--) {
    for (let x = 0; x < col; x++) {
      const cell = input[y][x];
      if (cell.startsWith("L")) {
        Empfaenger[cell] = () => solve(input, x, y - 1);
      } else if (cell.startsWith("Q")) {
        QuelleStatus[cell] = false;
      }
    }
  }

  // Gekuerzt: Ausgabe fuer alle Kombinationen
})
```

6 Zusatzaufgabe

Meine Webanwendung für die interaktive Visualisierung von Bausteinen wurde komplett in reinem JavaScript, ohne die Verwendung externer Bibliotheken, entwickelt. Dies gewährleistet maximale Flexibilität und Kontrolle über alle Funktionen des Projekts, wobei der Fokus auf der Canvas-Technologie liegt. Die Inspiration für das Design stammt von ähnlichen Webanwendungen wie Figma und GeoGebra.

Im Hintergrund ist ein Koordinatensystem sichtbar, um das sich die Bausteine magnetisch bewegen können. Das Originaldesign des Aufgabenblatts wurde für die Darstellung der Bausteine verwendet. Sie können die Datei index.html problemlos öffnen, um sich das Projekt anzusehen.

Hier sind die wichtigsten Funktionen:

Pan & Zoom

- Leichte Navigation durch Maus- und Tastatursteuerung.
- Funktioniert sowohl auf Tracking-Pads mit Gestenerkennung, ähnlich wie in Figma, als auch mit einer herkömmlichen Maus.

- Shortcuts ermöglichen den Wechsel zwischen Instrumenten:
  - `M/H` - Bewegen der Ansicht
  - `V` - Auswahl von Bausteinen
  - `Strg + Plus` - Hereinzoomen
  - `Strg + Minus` - Herauszoomen

Bewegen von Bausteinen

- Mit dem Auswahlwerkzeug (Taste `V`) können Bausteine auf dem Feld verschoben werden.
- Möglichkeit, mehrere Elemente gleichzeitig auszuwählen und zu bearbeiten, indem ein Rechteck gezogen wird, das die gewünschten Elemente berührt.

Rotieren von roten Bausteinen

- Rote Bausteine können durch doppeltes Klicken verschoben werden.

Ein- und Ausschalten von Lichtquellen

- Taschenlampe kann durch doppeltes Klicken ein- oder ausgeschaltet werden.

Löschen von Elementen

- Ausgewählte Elemente können durch Drücken der `Backspace`-Taste gelöscht werden.

Importieren von Beispielaufgaben

- Textdateien im Format von Beispieldateien können durch Ziehen der Datei auf die Webseite importiert werden.

Einfügen von Elementen

- Ein Element kann durch Klicken an einer beliebigen Stelle auf dem Koordinatenfeld eingefügt werden. Im daraufhin geöffneten Kontextmenü kann ein Element ausgewählt werden.

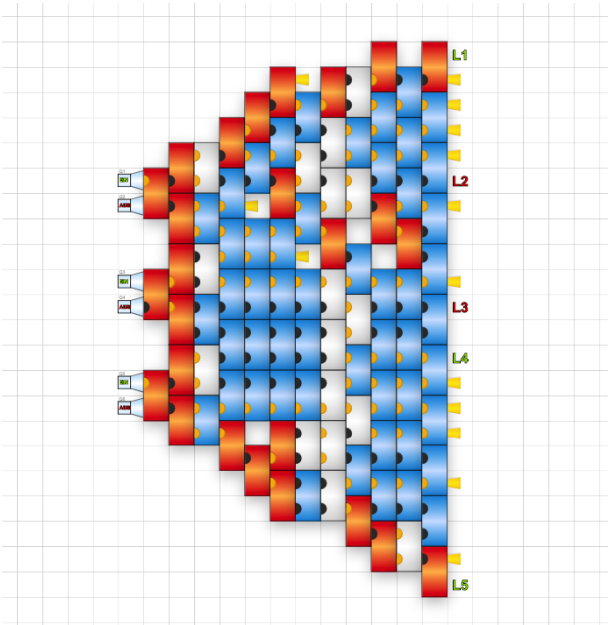

Abbildung 1: Bildschirmaufnahme der Webseite, auf der Beispiel 5 dargestellt ist.