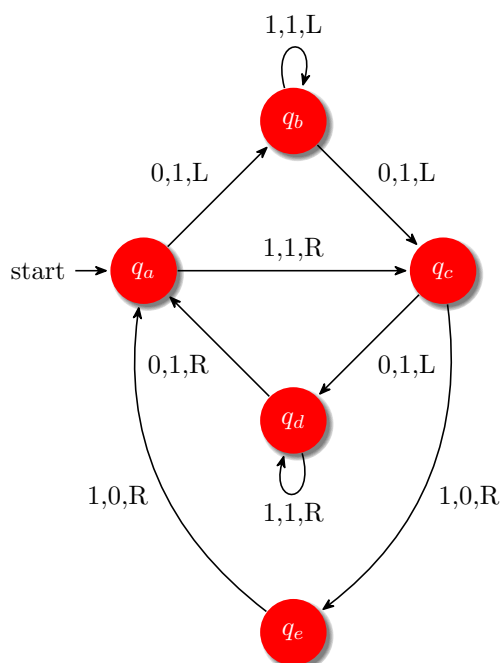


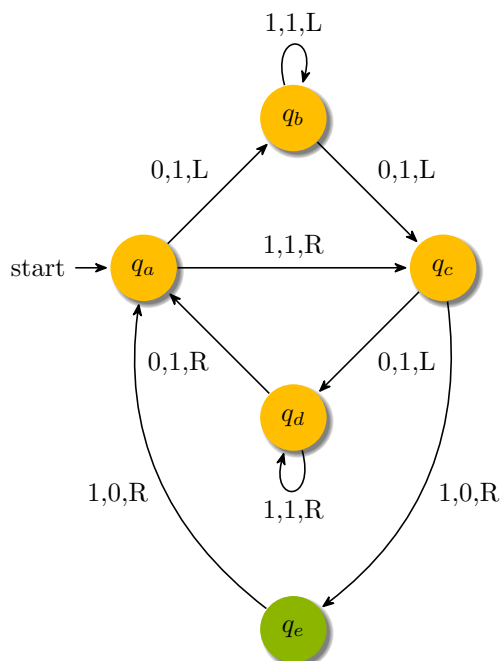
1 Turing machine

1.1 Original busy beaver



The current candidate for the busy beaver for five states. It is presumed that this Turing machine writes a maximum number of 1's before halting among all Turing machines with five states and the tape alphabet $\{0, 1\}$. Proving this conjecture is an open research problem.

1.2 Modified busy beaver



2 Preliminary

2.1 Mathematische Symbole

https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols

2.2 Menge

[https://de.wikipedia.org/wiki/Menge_\(Mathematik\)](https://de.wikipedia.org/wiki/Menge_(Mathematik))

Eine Menge wird über Aufzählung oder Konstruktor - Formel - definiert.

2.2.1 Aufzählung - Enumeration

[https://de.wikipedia.org/wiki/Sinn_\(Wahrnehmung\)](https://de.wikipedia.org/wiki/Sinn_(Wahrnehmung))

S oder Sinne: Hören, Riechen, Schmecken, Sehen, Tasten, Temperatur, Schmerz, Gleichgewicht, Hunger, Durst, Harndrang, ...

$$S = \{H, R, \dots\}$$

oder

x : ist eine Variable von S .

$|$: ist eine Operation und bedeutet: x kann ein Element oder Wert aus S zugewiesen werden.

$$S = \{x \mid x = H \vee x = R, \dots\}$$

2.2.2 Konstruktor - Set builder

https://en.wikipedia.org/wiki/Set-builder_notation

Menge aller natürlichen Zahlen: $\mathbb{N} = \{x \in \mathbb{N} \mid x\}$

oder

$$\mathbb{N} = \{n \mid n \in \mathbb{N}\}$$

Menge der ungeraden Integer: $\{n \in \mathbb{Z} \mid (\exists k [k \in \mathbb{Z} \wedge n = 2k + 1])\}$

oder

$$\{2n + 1 \mid n \in \mathbb{Z}\}$$

und der inverse Konstruktor: $\{2i + 1 \mid i \in \mathbb{Z}\} = \{j \mid (j - 1)/2 \in \mathbb{Z}\}$

2.3 Mengenprodukt

https://de.wikipedia.org/wiki/Kartesisches_Produkt

Das kartesische Produkt $A \times B$ - „A kreuz B“ - zweier Mengen A und B ist die Menge aller geordneten Paare - 2-Tupel oder Liste - (a, b) , in denen a ein Element aus A und b ein Element aus B ist:

$$A \times B := \{(a, b) \mid a \in A, b \in B\}$$

2.3.1 Beispiel

$$A = \{x, y, z\}$$

$$B = \{1, 2, 3\}$$

$$A \times B = \{(x, 1), (x, 2), (x, 3), (y, 1), (y, 2), (y, 3), (z, 1), (z, 2), (z, 3)\}$$

2.4 Funktion

[https://en.wikipedia.org/wiki/Function_\(mathematics\)](https://en.wikipedia.org/wiki/Function_(mathematics))

[https://de.wikipedia.org/wiki/Funktion_\(Mathematik\)](https://de.wikipedia.org/wiki/Funktion_(Mathematik))

Eine Funktion oder Abbildung ist eine Beziehung (Relation) zwischen zwei Mengen, die jedem Element der einen Menge (Funktionsargument: x-Wert) genau ein Element der anderen Menge (Funktionswert: y-Wert) zuordnet.

Eine Funktion f ordnet jedem Element x einer Definitionsmenge D genau ein Element y einer Zielmenge Z zu.

<https://mo.mathematik.uni-stuttgart.de/kurse/kurs44/seite25.html>

$$f : D \rightarrow Z, x \mapsto y \quad \text{oder} \quad f : \begin{cases} D \rightarrow Z \\ x \mapsto y \end{cases}$$

2.4.1 Notation

<https://tex.stackexchange.com/questions/87867/do-i-need-a-specific-package-for-n>

Funktionsgleichung: $f(x) = x^2, x \in \mathbb{N}$

Zuordnungsvorschrift mit D : $x \rightarrow x^2, x \in \mathbb{N}$

Zuordnungsvorschrift mit D und Z : $f : \mathbb{N} \rightarrow \mathbb{N}, x \mapsto x^2$ oder $f : \begin{cases} \mathbb{N} \rightarrow \mathbb{N} \\ x \mapsto x^2 \end{cases}$

Wertetabelle:

x|1|2|3|...

y|1|4|9|...

Relation oder Aufzählung: $f = \{(1,1), (2,4), (3,9), \dots\}$

2.4.2 Ausdruck

[https://de.wikipedia.org/wiki/Funktion_\(Mathematik\)](https://de.wikipedia.org/wiki/Funktion_(Mathematik))

- x wird abgebildet auf f von x
- f von x wird x eindeutig zugeordnet
- y gleich f von x
- y ist das Bild von x unter der Abbildung f

2.4.3 Partielle Funktion

<https://mirror.dogado.de/tex-archive/graphics/pgf/base/doc/pgfmanual.pdf>

https://de.wikipedia.org/wiki/Partielle_Funktion

https://en.wikipedia.org/wiki/Partial_function

<https://latexdraw.com/exploring-tikz-arrows/>

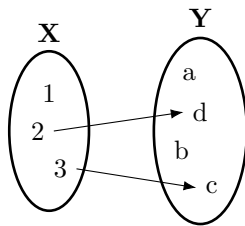
Eine partielle Funktion ordnet einem Element aus der Menge X ein Element aus der Menge Y zu.

Aufzählung

$X = \{1, 2, 3\}$

$Y = \{a, b, c, d\}$

$f: X \rightarrow Y = \{2 \rightarrow d, 3 \rightarrow c\}$



Konstruktion

Ungerade natürliche Zahlen.

$f: \mathbb{N} \rightarrow \mathbb{N}$

$f(x) = 2x + 1$

2.5 Algorithmus

<https://de.wikipedia.org/wiki/Algorithmus>

Ein Algorithmus ist eine Turingmaschine.

2.5.1 Deterministischer Algorithmus

[https://de.wikipedia.org/wiki/Determinismus_\(Algorithmus\)](https://de.wikipedia.org/wiki/Determinismus_(Algorithmus))

Ein Algorithmus ist deterministisch, wenn er für die gleiche Eingabe immer die gleiche Ausgabe erzeugt und die gleichen Zustände durchlaufen werden.

2.6 Deterministische Turingmaschine

<https://de.wikipedia.org/wiki/Turingmaschine>

https://en.wikipedia.org/wiki/Turing_machine

Die Zielmenge der partiellen Funktion ersetze ich Γ durch Σ , um eine Endlosschleife, heutiges Verständnis, zu vermeiden.

DTM ist eine deterministische Turingmaschine: $\langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle$:

- Q ist eine Zustandsmenge definiert über Aufzählung oder Generator, wie z.B. $2k + 1$, weil es sonst zu lange dauert, bis man alle aufgesagt hat, wie Zählen.
- Γ ist eine Menge von Speichersymbolen definiert über den I/O-Test.
- $b \in \Gamma$ ist ein Blank- oder Format-Symbol, das beim Parsen ignoriert wird.
- $\Sigma \subseteq \Gamma \setminus \{b\}$ ist die Menge der Ausgabesymbole.
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Sigma \times \{L, R\}$ ist eine partielle Funktion oder Übergangsfunktion:
 $L := \text{Linksbewegung}$ (left shift)
 $R := \text{Rechtsbewegung}$ (right shift)
 if $\delta \notin Q$ then *stuck*(DTM)
 Die Übergangsfunktion definiert den Folgezustand, der vom aktuellen Zustand abgeleitet wird. Man übersetzt das Eingabesymbol und sucht über eine Tabelle, ob der aktuelle Zustand die Eingabe erlaubt. Wenn das nicht zutrifft, stoppt die DTM und erzeugt eine Kopie von sich selbst - Core-Dump.

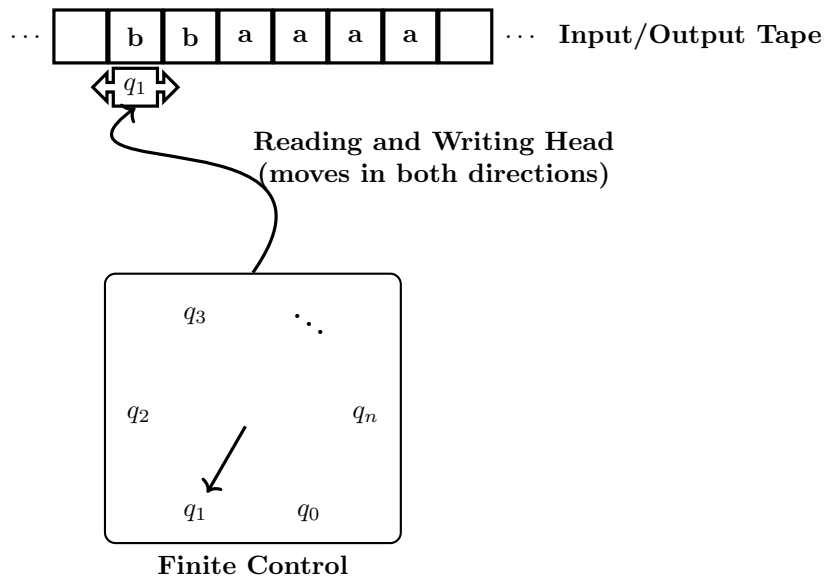
Oder.

Die Übergangsfunktion sagt mir, welchen Nachfolgezustand ich über den aktuellen Zustand erreichen kann.

- $q_0 \in Q$ ist der Anfangszustand.
- $F \subseteq Q$ ist die Menge der Endzustände.

2.6.1 Architecture

Draft



CPU

2.7 Busy beaver

https://en.wikipedia.org/wiki/Turing_machine

Legend

P := Program counter
S := Start or initial state
C := Current state
I := Input
O := Output
A := ALU
N := Next state
M := Memory state

DTM

- $Q = \{A, B, C, H\}$
- $\Gamma = \{0, 1\}$
- $b = 0$
- $\Sigma = \{1\}$
- $q_0 = S = A$
- $F = \{H\}$

2.7.1 States

<http://www.cs.uni.edu/~barr/CS1000/ppt/Chapter09-ComputerOperation.pdf>

https://en.wikipedia.org/wiki/Address_generation_unit

A := Arithmetical logical unit including AGU - address generation unit - with the micro operations: *read*, *calculate*, *write*, *address*

R := Right address

L := Left address

S := Stop or stay on the same address

C	I	O	A	N
S	0	0	S	A
A	0	1	R	B
A	1	1	L	C
B	0	1	L	A
B	1	1	R	B
C	0	1	L	B
C	1	1	R	H
H	1	1	S	H

2.7.2 Trace

Log

P	C	I	O	A	N	M
0	S	0	0	S	A	$\{0, \dots, 0, 0, 0, 0, \mathbf{0}, 0, 0, 0, \dots, 0\}$
1	A	0	1	R	B	$\{0, \dots, 0, 0, 0, 0, 1, \mathbf{0}, 0, 0, \dots, 0\}$
2	B	0	1	L	A	$\{0, \dots, 0, 0, 0, 0, \mathbf{1}, 1, 0, 0, \dots, 0\}$
3	A	1	1	L	C	$\{0, \dots, 0, 0, 0, \mathbf{0}, 1, 1, 0, 0, \dots, 0\}$
4	C	0	1	L	B	$\{0, \dots, 0, 0, \mathbf{0}, 1, 1, 1, 0, 0, \dots, 0\}$
5	B	0	1	L	A	$\{0, \dots, 0, \mathbf{0}, 1, 1, 1, 1, 0, 0, \dots, 0\}$
6	A	0	1	R	B	$\{0, \dots, 0, 1, \mathbf{1}, 1, 1, 1, 0, 0, \dots, 0\}$
7	B	1	1	R	B	$\{0, \dots, 0, 1, 1, \mathbf{1}, 1, 1, 0, 0, \dots, 0\}$
8	B	1	1	R	B	$\{0, \dots, 0, 1, 1, 1, \mathbf{1}, 1, 0, 0, \dots, 0\}$
9	B	1	1	R	B	$\{0, \dots, 0, 1, 1, 1, 1, \mathbf{1}, 0, 0, \dots, 0\}$
10	B	1	1	R	B	$\{0, \dots, 0, 1, 1, 1, 1, 1, \mathbf{0}, 0, \dots, 0\}$
11	B	0	1	L	A	$\{0, \dots, 0, 1, 1, 1, 1, \mathbf{1}, 0, 0, \dots, 0\}$
12	A	1	1	L	C	$\{0, \dots, 0, 1, 1, 1, \mathbf{1}, 1, 0, 0, \dots, 0\}$
13	C	1	1	R	H	$\{0, \dots, 0, 1, 1, \mathbf{1}, 1, 1, 0, 0, \dots, 0\}$
14	H	1	1	S	H	$\{0, \dots, 0, 1, 1, \mathbf{1}, 1, 1, 0, 0, \dots, 0\}$

Beobachtung: In der Berufsschule verwendet man bunte Begriffe, die man 2-dimensionalen mit z.T. bidirektionalen Pfeilen verbindet und zwar unspezifiziert.

Idee: Das ist ein Algorithmus und ein Automat. Als Modell verwende ich deterministische Turing Maschine, damit es nicht zu kompliziert wird.

Spiel: Jeder Mensch, Mathematiker oder Philosoph spielt dauernd mit irgendwelchen Zeichen:

xxx: Um alles zu vereinfachen, beschränke ich mich auf eine deterministische Turing Maschine - DTM - oder einen Computer, der das macht, was - tbd. - er laut Beschreibung kann und zwar sofort - tbd. - ein Ergebnis sofort liefert, das mit der Erwartung übereinstimmt.

xxx: Ziemlich frei habe ich alles undefiniert:

DTM = (C, I, R, B, F, T, S)

C := States of the computer

I := I/O characters (abstract)

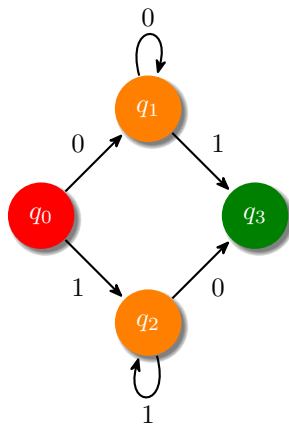
M := Memory

B := Power on or boot phase

F := Functions

T := Timeout actions like powersaveing

S := Power off or shutdown



3 Derivation

3.1 Legend

Applegreen: initial state

Amber: any state

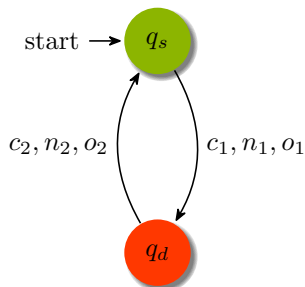
Coquelicot: final state

3.2 Subject - Deadline

1. **Subject - s:** Die Tractatus Dimensionen oder wie alle in jedem Kosmos denken
2. **Deadline - d:** 23.12.2022

Draft:

$s \leftrightarrow d$

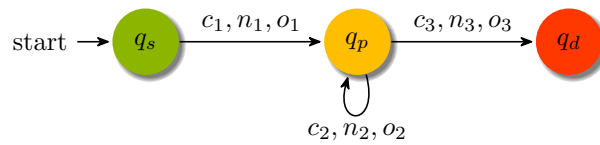


3.3 Proposition

1. **Subject - s:** Die Tractatus Dimensionen oder wie alle in jedem Kosmos denken
2. **Proposition - p:** Literate Programming / LuaTeX
3. **Deadline - p:** 23.12.2022

Draft:

$s \rightarrow l(p) \rightarrow d$



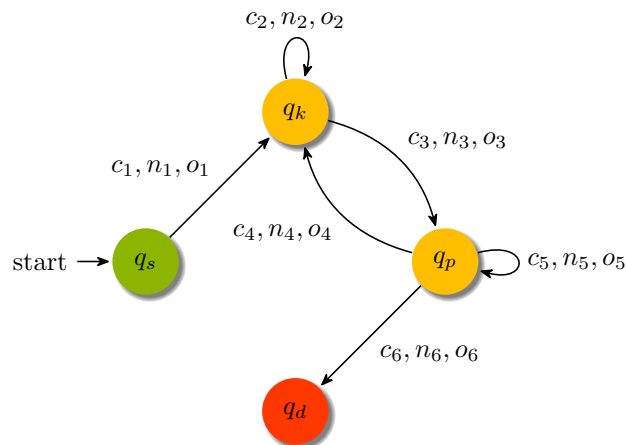
3.4 Knowledge

1. **Subject - s:** Die Tractatus Dimensionen oder wie alle in jedem Kosmos denken
2. **Knowledge - k:** Rezitation des Tractatus
3. **Proposition - p:** Literate Programming / LuaTeX
4. **Deadline - d:** 23.12.2022

Draft:

```

-> l(k) <-
|          |
s          v
          l(p)
          |
d <-
  
```



Contents

1. **Subject - s:** Die Tractatus Dimensionen
oder wie alle im Weltall denken
2. **Knowledge - k:** Rezitation des Tractatus
3. **Proposition - p:** Literate Programming / LuaTeX
4. **Deadline - d:** 23.12.2022
5. **Hypothesis - h:** i. Begriffe sind Koordinatenachsen

Legend:

Eine Landkarte versteht man über eine Übersetzung.

[https://de.wikipedia.org/wiki/Legende_\(Karte\)](https://de.wikipedia.org/wiki/Legende_(Karte))

Colors:

Wenn man die Ampelfarben nicht versteht,
kann man ganz schnell sterben.

<https://de.wikipedia.org/wiki/Ampel>

1. $s \leftrightarrow d$

2. $s \rightarrow l(p) \rightarrow d$

3. $\begin{array}{ccc} & \rightarrow l(k) <- & \\ | & & | \\ s & & v \\ & & l(p) \\ & & | \\ & d <- & \end{array}$

4. $\begin{array}{ccc} & \rightarrow l(k) <- & \\ | & & | \\ s & & v \\ & & l(p) \\ & & | \\ & d <- & \end{array}$