FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Friedrich-Schiller-Universität Jena
Fakultät Mathematik und Informatik
Modul Molekulare Algorithmen Sommersemester 2020

Projekt 11

Chemisches Digitalcomputermodell für den Vergleich zweier zweistelliger Zahlen im Ternärsystem auf $\leq$, das mit möglichst wenigen Spezien und Reaktionen auskommt

bearbeitet von: Alexander Henoch
Martikel-Nr.: 182709

Jena, den 06.07.2020

Inhaltsverzeichnis

1 Einleitung

1.1 Chemische Analogcomputermodelle

Ein Reaktionssystem, inklusive seiner Reaktionskinetiken, lässt sich im informatischen Sinne als Analogcomputer auffassen. Durch die entsprechenden Stoffkonzentrationen der Reaktanden und Katalysatoren lassen sich dabei die Eingabe, generierte Zwischenergebnisse und die Ausgabe kodieren. Die vollzogenen Reaktionen, die mögliche Veränderungen der Konzentrationswerte nach sich ziehen, werden dabei als Berechnungsprozesse interpretiert (Hinze 2013).
Eine mögliche Operation, die durch ein solches Reaktionssystem ausgeführt werden kann, ist die Addition. Hierbei repräsentieren die Eingabespezies und ihre dazugehörigen Anfangskonzentrationen die Operanden. Durch die Ausführung vorgegebener chemischer Reaktionen ergeben sich, im stationären Zustand, die Ausgabespezies. Diese stellen das Ergebnis der Berechnung dar (Hinze 2013).

$$\begin{aligned} X_1 &\xrightarrow{k_1} X_1 + Y \\ X_2 &\xrightarrow{k_2} X_2 + Y \\ Y &\xrightarrow{k_3} \emptyset \end{aligned}$$

Gl. 1.1 Chemische Reaktionen für eine nicht-negative mathematische Addition. Dieses Beispiel zeigt die nötigen chemischen Reaktionen für eine nicht-negative einfachen mathematische Addition. Die gezeigten Formeln wurden aus Hinze 2013 übernommen.

Im Falle einer einfachen nicht-negativen Addition werden mindestens drei Reaktionen benötigt (siehe Gl. 1.1). Die Anfangskonzentrationen von X_1 und X_2 bauen dabei unabhängig voneinander die Konzentration von Y auf, die sich am Anfang der Ausführung bei $0\frac{\text{mol}}{\text{l}}$ befindet. Für das Beispiel der Berechnung

$$3 + 2 = 5$$

werden dann die entsprechenden Anfangskonzentrationen von X_1 als $3\frac{\text{mol}}{\text{l}}$ und von X_2 als $2\frac{\text{mol}}{\text{l}}$ festgelegt. Die Konzentration von Y im stationären Zustand beträgt dann $5\frac{\text{mol}}{\text{l}}$ (siehe Gl. 1.2) (Hinze 2013).

$$\begin{array}{llll} Y[\infty] & = X_1[0] & + X_2[0] \\ 5 & = 3 & + 2 \end{array}$$

Gl. 1.2 Ergebnis einer beispielhaften nicht-negativen Additon. Dieses Beispiel zeigt das Ergebnis der Durchführung einer Addition basierend auf den Reaktionen aus Gl. 1.1. Das gezeigte Beispiel basiert auf Hinze 2013.

1.2 Chemische Digitalcomputermodelle

In der heutigen Zeit werden Analogcomputer auf Grund der geringen Störungssicherheit nicht mehr häufig verwendet. An ihre Stelle sind die Digitalcomputer getreten. Das chemische Prinzip der Analogcomputermodelle kann ebenfalls auf diese übertragen und erweitert werden und bildet damit die so genannten Digitalcomputermodelle. Diese werden ähnlich wie die Analogcomputermodelle über chemische Reaktionen und die genutzten Reaktanden und Katalysatoren betrieben. Hierbei können DNA, RNA und Proteine als Spezies genutzt werden. Dabei können über ihre komplexen Raumstrukturen stabile Mechanismen gebildet werden und somit eine klare Umschaltung durch katalysierung erreicht werden (Hinze 2013).

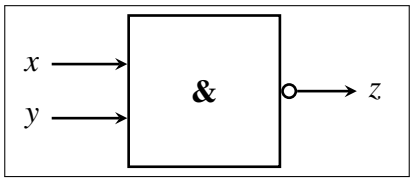

Fig. 1.1 Simples NAND-Gatter. Gezeigt ist ein simples NAND-Gatter, welches zwei Eingaben fordert und diese auf den selben Wert untersucht. Sind die Werte gleich ist die Ausgabe der gegenteilige Wert der Eingaben. Andernfalls wird der größere Eingabewert ausgegeben.

Besonders gut eignen sich zu diesem Zweck Proteine, da sie im Vergleich zu DNA und RNA aus einer größeren Anzahl an Bausteinen bestehen. Bei RNA und DNA können nur vier Nukleotide, bei Proteinen dagegen mindestens 22 proteinbildende Aminosäuren als monomere Bausteine genutzt werden. Die Möglichkeit an interagierenden Spezies steigt somit durch die höhere Anzahl an Bausteinen und damit einhergehend höhere Anzahl an tertiären Strukturen stark an. Die Interaktionen der Spezies werden dabei dann als Schaltung zwischen den boolischen Zuständen wahr ≡ $True$ und falsch ≡ $False$ genutzt (Hinze 2013).

Tab. 1.1 Logische Übertragungstabelle. Gezeigt ist die Übertragungstabelle, die sich aus dem NAND-Gatter in Fig. 1.1 ergibt.

x	x_{Bool}	y	y_{Bool}	z	z_{Bool}
0	$False$	0	$False$	1	$True$
0	$False$	1	$True$	1	$True$
1	$True$	0	$False$	1	$True$
1	$True$	1	$True$	0	$False$

Über so genannte logische Übergangstabellen werden dabei die nötigen Schaltungen festgelegt, um einen Mechanismus z.B. ein logisches NAND-Gatter (siehe Fig. 1.1) zu modellieren (siehe Tab. 1.1). Die Zustände von z könnten dabei als $True$ und $False$ einer bestimmten Fragestellung betreffend gedeutet werden. Die nötigen Reaktionen für das Digitalcomputermodell, sowie die Anzahl nötiger Proteine, kann dann anschließend häufig noch durch verschiedene Algorithmen minimiert werden. Für Tab. 1.1 würden beispielsweise vier Reaktionen und insgesamt sechs

Spezies benötigt werden. Jeweils eine Spezies für jede Variable multipliziert mit jedem Zustand (siehe Gl. 1.3 (Hinze 2013).

$$\begin{aligned} xF + yF + zF &\longrightarrow xF + yF + zT \\ xF + yT + zF &\longrightarrow xF + yT + zT \\ xT + yF + zF &\longrightarrow xT + yF + zT \\ xT + yT + zT &\longrightarrow xT + yT + zF \end{aligned}$$

Gl. 1.3 Reaktionen der Übertragungstabelle. Die aus der Übertragungstabelle Tab. 1.1 gewonnenen chemischen Reaktionen.

1.3 Ternärsystem

Im Gegensatz zu dem Binärsystem, in dem Zahlen als $\{0_2, 1_2\}^*$ dargestellt werden und in dem logische Schaltvorgänge $\{False, True\}$ ausgeführt werden, stellt das Ternärsystem natürliche Zahlen mit der Basis drei dar $\{0_3, 1_3, 2_3\}^*$. Auswertungslogiken in dieser Form werden z.B. in der MSR-Technik verwendet, da hierbei drei verschiedene Zustände bei der Auswertung genutzt werden (Brusentsov und Ramil Alvarez 2011).

Tab. 1.2 Übersetzungstabelle Decimal zu Ternär. Zum besseren Verständnis ternärer Zahlen ist die Übersetzung der Decimalzahlen von eins bis zehn in ihre ternären Äquivalente gezeigt. In der vorliegenden Projektarbeit sind nur die zweistelligen Ternärzahlen (eins bis acht) von Bedeutung.

x_{Dec}	x_{Tern}
0_{10}	00_3
1_{10}	01_3
2_{10}	02_3
3_{10}	10_3
4_{10}	11_3
5_{10}	12_3
6_{10}	20_3
7_{10}	21_3
8_{10}	22_3
9_{10}	100_3
...	...

In der vorliegenden Projektarbeit wird ein chemisches Digitalcomputermodell in Copasi erstellt, dass zwei zweistellige Ternärzahlen x und y über den Operator $\leq$ miteinander vergleicht und das Ergebnis logisch als $True$ oder $False$ ausgiebt. Da sich Ternärzahlen nicht, ohne komplexe dreiwertige Logiken, mit den erwähnten logischen Gattern vergleichen lassen, wird eine andere Methode gesucht, um eine entsprechende Schalttabelle zu erstellen, die möglichst wenig Reaktionen und Spezies enthält. Anschließend wird eine Fallstudie mit zwei Ternärzahlen in Copasi durchgeführt (Brusentsov und Ramil Alvarez 2011).

2 Material und Methoden

Zum besseren Verständnis der verwendeten Methodik, wird das verwendete Skript im folgenden in geringerer Komplexität und anhand eines vereinfachten Beispiels erläutert. Für den Vergleich zweier einstelliger Ternärzahlen, anstelle von zwei zweistelligen Ternärzahlen, würde sich die folgende Übertragungstabelle ergeben (siehe Tab. 2.1). Da nur einstellige Ternärzahlen in diesem Beispiel verglichen werden, müssen x und y im Gegensatz zum vorgestellten Projekt, nicht in x_0, x_1, y_0 und y_1 aufgeteilt werden.

Tab. 2.1 Maximale Übertragungstabelle des Erläuterungsbeispiels. Die dargestellte Tabelle repräsentiert die neun möglichen Vergleiche von zwei einstelligen Ternärzahlen x und y, sowie deren dazugehörige Spezies.

x	x_{Tern}	x_{Dec}	y	y_{Tern}	y_{Dec}	z	z_{Bin}	z_{Bool}
L	0_3	0_{10}	L	0_3	0_{10}	H	1_2	$True$
L	0_3	0_{10}	M	1_3	1_{10}	H	1_2	$True$
L	0_3	0_{10}	H	2_3	2_{10}	H	1_2	$True$
M	1_3	1_{10}	L	0_3	0_{10}	L	0_2	$False$
M	1_3	1_{10}	M	1_3	1_{10}	H	1_2	$True$
M	1_3	1_{10}	H	2_3	2_{10}	H	1_2	$True$
H	2_3	2_{10}	L	0_3	0_{10}	L	0_2	$False$
H	2_3	2_{10}	M	1_3	1_{10}	L	0_2	$False$
H	2_3	2_{10}	H	2_3	2_{10}	H	1_2	$True$

Aus der maximalen Übertragungstabelle ohne Optimierung durch löschen von Redundanzen Tab. 2.1 könnte nun die maximale Anzahl an Reaktionen übertragen werden (siehe Gl. 2.1).

$$
\begin{aligned}
xL + yL + zL &\longrightarrow xL + yL + zH \\
xL + yM + zL &\longrightarrow xL + yM + zH \\
xL + yH + zL &\longrightarrow xL + yH + zH \\
xM + yL + zH &\longrightarrow xM + yL + zL \\
xM + yM + zL &\longrightarrow xM + yM + zH \\
xM + yH + zL &\longrightarrow xM + yH + zH \\
xH + yL + zH &\longrightarrow xH + yL + zL \\
xH + yM + zH &\longrightarrow xH + yM + zL \\
xH + yH + zL &\longrightarrow xH + yH + zH
\end{aligned}
$$

Gl. 2.1 Reaktionen der Übertragungstabelle. Aus der maximalen Übertragungstabelle Tab. 2.1 können die abgebildeten neun chemischen Reaktionen gewonnen werden.

Um nun zu erläutern, wie die Optimierung grundlegend durchgeführt wurde, ist hier eine vereinfachte Version des in der Projektarbeit verwendeten und beigefügten Skripts dargestellt.

```
import numpy as np

#Maximale Uebertragungstabelle
Reaction = np.array([
    [0, 0, 1],
    [0, 1, 1],
    [0, 2, 1],
    [1, 0, 0],
    [1, 1, 1],
    [1, 2, 1],
    [2, 0, 0],
    [2, 1, 0],
    [2, 2, 1]
])
Reaction2 = np.copy(Reaction)

#Anzahl geschachtelter {0,1,2}-Loop = 1
for i in range(0,3):

    #Anzahl der Listen = 2
    Listx = []
    Listy = []

    #Anzahl der if-Abfragen in der Zeilen-Loop = 2
    for row in Reaction:
        if row[0] == i:
            Listx.append(row[2])
        if row[1] == i:
            Listy.append(row[2])

    #Anzahl der Listenlänge if-Abfragen = 2
    if len(set(Listx)) == 1:
        z = list(set(Listx))[0]
        for j in range(0,3):
            if np.count_nonzero(Reaction2[np.where((Reaction == (i,
                j, z)).all(axis=1)), :] == 3) <= 1:
                Reaction2[np.where((Reaction == (i, j,
                    z)).all(axis=1)), :] = [i, 3, z]

    if len(set(Listy)) == 1:
        z = list(set(Listy))[0]
        for j in range(0,3):
            if np.count_nonzero(Reaction2[np.where((Reaction == (j,
                i, z)).all(axis=1)), :] == 3) <= 1:
                Reaction2[np.where((Reaction == (j, i,
                    z)).all(axis=1)), :] = [3, i, z]
```

```
43
44 Reaction3 = np.unique(Reaction2, axis=0)
45
46 #Minimale Uebertragungstabelle
47 Reaction4 = np.where(Reaction3 == 3, '*', Reaction3)
48 np.savetxt("NAME.csv", Reaction4, delimiter=",", fmt='%s')
```

Dieses teilt sich in mehrere Bereiche auf. Zuerst wird die maximale Übertragungstabelle als Array durch das Paket **NumPy Version 1.17.4** in **Python Version 3.8.2** eingepflegt (Oliphant 2006; Van Rossum und Drake Jr 1995).
Für die weitere Erläuterung wird eine Variable n nun als Anzahl der einstelligen Ternärzahlen der Übertragungstabelle ohne die Ausgabe z festgelegt.

$$n = 2$$

Die Tiefe der Schachtelung der Schleife der Länge drei (Iteration über 0_3, 1_3 und 2_3) ist nun

$$n - 1 = 1$$

also ist in dem vereinfachten Beispiel eine Schleife genug. Bei der Durchführung der Projektarbeit mussten dann also drei geschachtelte Schleifen der Länge drei in das Skript eingefügt werden, da vier einstellige Ternärzahlen genutzt werden mussten.
Die Anzahl der Listen, der If-Abfragen in der Iteration über die Anzahl der Tabellenzeilen und der Listenlänge bezogenen If-Abfragen wird berechnet durch

$$\sum_{k=1}^{n-1} \binom{n}{k} = 2$$

und stellt damit die Anzahl alle Kombinationen dar, über die sich die einstelligen Ternärzahlen miteinander verbinden können, so dass mindestens eine einstellige Ternärzahl ausgeschlossen wird, welche dann iteriert werden kann. In der vorgestellten Projektarbeit mussten dadurch z.B. 14 Listen angelegt werden.
In diesem Beispiel muss wie bereits erwähnt, nur eine Schleife der Länge drei eingefügt werden und somit werden nur genau drei Durchläufe durch die Tabelle durchgeführt. In jedem Durchlauf werden Listen erstellt, die das Ergebnis z jeder Zeile enthält, in der die einstellige ternäre Zahl x oder y jeweils einem spezifischen Wert (dargestellt durch die Iteration der übergreifenden Schleife) entspricht, während die andere nicht-festgelegte ternäre Zahl iteriert werden kann.
Anschließend wird jede Liste auf die Anzahl verschiedener Ausgaben $\{0_2, 1_2\}$ von z überprüft. Ergibt diese Überprüfung den Wert eins, so sind die Zeilen, die zu den Ausgaben in dieser Liste gehören redundant, da sie alle die gleiche Ausgabe enthalten, obwohl mindestens eine ternäre Zahl iteriert wird. Diese Zeilen können dann zusammengefasst werden. Zum besseren Verständnis sind die ersten Durchläufe des Skripts in Tab. 2.2 dargestellt. Die erste *Listex* stellt

Tab. 2.2 Beispielhafter Programmdurchlauf. Zur besseren Erläuterung ist in dieser Tabelle der beispielhafte Programmdurchlauf verkürzt dargestellt. Die Iteration der Schleife wird über die Variable i verdeutlicht. Die Ausgaben von z werden entsprechend $x = i$ und $y = i$ jeweils in Listen zusammengefasst. Die rote Schriftfarbe verdeutlicht die Zusammensetzung der *Listex*, die bei der ersten Iteration $i = 0_3$ redundant ist und somit auf eine Zeile reduziert werden kann, da alle Ausgaben von $z = 1_2$ entsprechen (siehe Tab. 2.3).

i_{Tern}	x_{Tern}	y_{Tern}	z_{Bin}	*Listx*	*Listy*
0_3	0_3	0_3	1_2		
0_3	0_3	1_3	1_2		
0_3	1_3	0_3	0_2	1_2,1_2,1_2	1_2,0_2,0_2
0_3	0_3	2_3	1_2		
0_3	2_3	0_3	0_2		
1_3	1_3	0_3	0_2		
1_3	0_3	1_3	1_2		
...	...	...	...	...	...

Tab. 2.3 Minimale Übertragungstabelle des Erläuterungsbeispiels. Die dargestellte Tabelle repräsentiert die 6 mindestens notwendigen Zeilen um zwei einstellige Ternärzahlen x und y zu vergleichen, sowie die dazugehörigen Spezies von x und y. Das auftreten von * symbolisiert, dass an dieser Stelle jede Spezies der Variable stehen kann.

x	y	z
L	$*$	H
M	L	L
M	M	H
H	L	L
H	M	L
$*$	H	H

hier dann einen redundanten Fall dar, da alle Ausgaben von z in der Liste den binären Wert 1_2 haben. Die 3 Zeilen, in denen x den ternären Wert 0_3 annimmt, entspricht der Spezies xL, können also zu einer Zeile zusammengefasst werden (siehe Tab. 2.3).

$$
\begin{aligned}
xL &+ zL \longrightarrow xL &+ zH \\
xM + yL &+ zH \longrightarrow xM + yL &+ zL \\
xM + yM + zL &\longrightarrow xM + yM + zH \\
xH + yL &+ zH \longrightarrow xH + yL &+ zL \\
xH + yM &+ zH \longrightarrow xH + yM + zL \\
yH &+ zL \longrightarrow &yH + zH
\end{aligned}
$$

Gl. 2.2 Reaktionen der Übertragungstabelle. Aus der minimalen Übertragungstabelle Tab. 2.3 können nun die hier gezeigten 6 Reaktionen gewonnen werden. Die Zellen mit * wurden bei der Übersetzung in die Reaktionen ausgelassen, da die betreffenden Spezies in den Reaktionen als Katalysatoren nicht benötigt werden.

Bei der Zusammenfassung wird hierarchisch vorgegangen. Zeilen in denen z.B. die Ausgabe identisch bleibt, obwohl gleichzeitig zwei ternäre Zahlen iteriert werden, haben eine höhere

Priorität, als wenn nur eine ternäre Zahl iteriert wird.
Schlussendlich wird die optimierte minimale Tabelle als CSV-Tabelle ausgegeben und kann in die entsprechenden mindestens nötigen chemischen Reaktionen umgeformt werden (siehe Tab. 2.3).
Die chemischen Reaktionen (siehe Gl. 2.2) können dann in Copasi eingepflegt werden, um das Digitalcomputermodell zu testen. Für die Durchführung der Fallstudie in Unterabs. 3.2 wurde **CopasiUI Linux Version 4.28 (Build 226)** genutzt (Hoops et al. 2006)).

3 Ergebnisse

3.1 Optimierung der Übertragungstabelle

Da beide Ternärzahlen x und y jeweils zweistellig sind und Dezimalwerte von null bis acht annehmen können, ergab sich eine Übertragungstabelle mit 81 Zeilen. Die Ausgabe z wurde dabei als binär festgelegt, da es nur zwei Ausgaben gibt ($True$ und $False$) und eine ternäre Ausgabe die Anzahl der Zeilen auf 162 erhöht hätte. Des Weiteren wurden die zweistelligen Ternärzahlen x und y in die einstelligen Ternärzahlen x_0, x_1, y_0 und y_1 übertragen. Diese Teilen sich jeweils in drei Aktivierungszustände auf $\{L, M, H\}$, diese entsprechen $\{0_3, 1_3, 2_3\}$. Die zweistelligen Ternärzahlen x und y werden also in die einstelligen Ternärzahlen x_0 und x_1, sowie y_0 und y_1 aufgeteilt. Diese entsprechen wiederum abhängig davon welche ihrer zugeschriebenen Spezies in erhöhter Konzentration auftritt, den Werten 0_3, 1_3 oder 2_3. Wenn bei der Eingabe der zweistelligen ternären Zahlen z.B. x als 01_3 festgelegt wird, so wird gleichzeitig die Konzentration der Spezies x_0L ($x_0 = 0_3$) und x_1M ($x_1 = 1_3$) auf $1\frac{\text{mol}}{\text{l}}$ angehoben und die Konzentrationen der Spezies x_0M, x_0H, x_1L und x_1H auf $0\frac{\text{mol}}{\text{l}}$ abgesenkt (siehe Tab. 3.1).

Tab. 3.1 Maximale Übertragungstabelle der Vergleiche. Die dargestellte Tabelle repräsentiert gekürzt die 81 möglichen Vergleiche von zwei zweistelligen Ternärzahlen x und y, sowie die dazugehörigen einstelligen Ternärzahlen von x_0, x_1, y_0 und y_1 und deren entsprechende Spezies.

x_0	x_1	$x_{1_{Tern}}$	$x_{2_{Tern}}$	x_{Tern}	y_0	y_1	$y_{0_{Tern}}$	$y_{1_{Tern}}$	y_{Tern}	z	z_{Bool}
L	L	0_3	0_3	00_3	L	L	0_3	0_3	00_3	H	$True$
L	L	0_3	0_3	00_3	L	M	0_3	1_3	01_3	H	$True$
L	L	0_3	0_3	00_3	L	H	0_3	2_3	02_3	H	$True$
L	L	0_3	0_3	00_3	M	L	1_3	0_3	10_3	H	$True$
L	L	0_3	0_3	00_3	M	M	1_3	1_3	11_3	H	$True$
L	L	0_3	0_3	00_3	M	H	1_3	2_3	12_3	H	$True$
L	L	0_3	0_3	00_3	H	L	2_3	0_3	20_3	H	$True$
L	L	0_3	0_3	00_3	H	M	2_3	1_3	21_3	H	$True$
L	L	0_3	0_3	00_3	H	H	2_3	2_3	22_3	H	$True$
L	M	0_3	1_3	01_3	L	L	0_3	0_3	00_3	L	$False$
...	...	...	...	...	...	...	...	...	...	...	...
H	M	2_3	1_3	21_3	H	H	2_3	2_3	22_3	H	$True$
H	H	2_3	2_3	22_3	L	L	0_3	0_3	00_3	L	$False$
H	H	2_3	2_3	22_3	L	M	0_3	1_3	01_3	L	$False$
H	H	2_3	2_3	22_3	L	H	0_3	2_3	02_3	L	$False$
H	H	2_3	2_3	22_3	M	L	1_3	0_3	10_3	L	$False$
H	H	2_3	2_3	22_3	M	M	1_3	1_3	11_3	L	$False$
H	H	2_3	2_3	22_3	M	H	1_3	2_3	12_3	L	$False$
H	H	2_3	2_3	22_3	H	L	2_3	0_3	20_3	L	$False$
H	H	2_3	2_3	22_3	H	M	2_3	1_3	21_3	L	$False$
H	H	2_3	2_3	22_3	H	H	2_3	2_3	22_3	H	$True$

Um diese Tabelle nun weiter zu optimieren, sowie Spezies und Reaktionen einzusparen, wurde

das im Anhang beigefügte Skript genutzt. Somit konnten z.B. bereits die ersten acht Zeilen von Tab. 3.1 eingespart werden, denn wenn x_0 und x_1 die einstellige ternäre Zahl 0_3 annehmen, ist es nicht mehr von Bedeutung, welchen ternären Wert y_0 und y_1 annehmen, bzw. welche Spezies von y_0 und y_1 mit erhöhter Konzentration vorhanden sind. Denn die Konzentration von zH wird durch das Vorhandensein der Spezies x_0L und x_1L immer auf $1\frac{\text{mol}}{\text{l}}$ ansteigen. Dies lässt sich dadurch erklären, dass jede nicht-negative dezimale Zahl ≥ der dezimalen Zahl null ist.

$$[x_0L] = 1\frac{\text{mol}}{\text{l}};\ [x_1L] = 1\frac{\text{mol}}{\text{l}} \equiv x_0 = 0_3;\ x_1 = 0_3 \equiv x = 00_3 \equiv 0_{10}$$

Tab. 3.2 Minimale Übertragungstabelle. Die dargestellte Tabelle repräsentiert die 24 mindestens notwendigen Zeilen um zwei zweistellige Ternärzahlen x und y zu vergleichen, sowie die Spezies von den dazugehörigen einstelligen Ternärzahlen x_0, x_1, y_0 und y_1. Das auftreten von * symbolisiert, dass an dieser Stelle jede Spezies von z.B. x_0 stehen kann.

x_0	x_1	y_0	y_1	z
L	L	$*$	$*$	H
L	M	$*$	M	H
L	$*$	M	$*$	H
L	$*$	H	$*$	H
L	$*$	$*$	H	H
M	L	M	$*$	H
M	M	M	L	L
M	M	M	M	H
M	H	M	L	L
M	H	M	M	L
M	$*$	L	$*$	L
M	$*$	M	H	H
M	$*$	H	$*$	H
H	M	$*$	L	L
H	H	$*$	L	L
H	H	$*$	M	L
H	$*$	L	$*$	L
H	$*$	M	$*$	L
$*$	L	H	$*$	H
$*$	M	L	L	L
$*$	M	H	M	H
$*$	H	L	L	L
$*$	H	L	M	L
$*$	$*$	H	H	H

Die generierte neue Übertragungstabelle (siehe Tab. 3.2) besteht nun nur noch aus 24 Zeilen, wodurch also mindestens 24 chemische Reaktionen und 14 verschiedene Spezies benötigt werden, um die volle Funktionalität zu gewährleisten. Die verschiedenen benötigten Reaktionen sind in Gl. 3.1 zusammengefasst dargestellt. Diese 24 Reaktionen und 14 Spezies wurden für die Fallstudie in Copasi eingetragen.

$$
\begin{array}{|llllllllllll|}
\hline
x_0L & +x_1L & & & +zL & \longrightarrow & x_0L & +x_1L & & & & +zH \\
x_0L & +x_1M & & +y_1M & +zL & \longrightarrow & x_0L & +x_1M & & +y_1M & +zH \\
x_0L & & +y_0M & & +zL & \longrightarrow & x_0L & & +y_0M & & +zH \\
x_0L & & +y_0H & & +zL & \longrightarrow & x_0L & & +y_0H & & +zH \\
x_0L & & & +y_1H & +zL & \longrightarrow & x_0L & & & +y_1H & +zH \\
x_0M & +x_1L & +y_0M & & +zL & \longrightarrow & x_0M & +x_1L & +y_0M & & +zH \\
x_0M & +x_1M & +y_0M & +y_1L & +zH & \longrightarrow & x_0M & +x_1M & +y_0M & +y_1L & +zL \\
x_0M & +x_1M & +y_0M & +y_1M & +zL & \longrightarrow & x_0M & +x_1M & +y_0M & +y_1M & +zH \\
x_0M & +x_1H & +y_0M & +y_1L & +zH & \longrightarrow & x_0M & +x_1H & +y_0M & +y_1L & +zL \\
x_0M & +x_1H & +y_0M & +y_1M & +zH & \longrightarrow & x_0M & +x_1H & +y_0M & +y_1M & +zL \\
x_0M & & +y_0L & & +zH & \longrightarrow & x_0M & & +y_0L & & +zL \\
x_0M & & +y_0M & +y_1H & +zL & \longrightarrow & x_0M & & +y_0M & +y_1H & +zH \\
x_0M & & +y_0H & & +zL & \longrightarrow & x_0M & & +y_0H & & +zH \\
x_0H & +x_1M & & +y_1L & +zH & \longrightarrow & x_0H & +x_1M & & +y_1L & +zL \\
x_0H & +x_1H & & +y_1L & +zH & \longrightarrow & x_0H & +x_1H & & +y_1L & +zL \\
x_0H & +x_1H & & +y_1M & +zH & \longrightarrow & x_0H & +x_1H & & +y_1M & +zL \\
x_0H & & +y_0L & & +zH & \longrightarrow & x_0H & & +y_0L & & +zL \\
x_0H & & +y_0M & & +zH & \longrightarrow & x_0H & & +y_0M & & +zL \\
 & x_1L & +y_0H & & +zL & \longrightarrow & & x_1L & +y_0H & & +zH \\
 & x_1M & +y_0L & +y_1L & +zH & \longrightarrow & & x_1M & +y_0L & +y_1L & +zL \\
 & x_1M & +y_0H & +y_1M & +zL & \longrightarrow & & x_1M & +y_0H & +y_1M & +zH \\
 & x_1H & +y_0L & +y_1L & +zH & \longrightarrow & & x_1H & +y_0L & +y_1L & +zL \\
 & x_1H & +y_0L & +y_1M & +zH & \longrightarrow & & x_1H & +y_0L & +y_1M & +zL \\
 & & y_0H & +y_1H & +zL & \longrightarrow & & & y_0H & +y_1H & +zH \\
\hline
\end{array}
$$

Gl. 3.1 Reaktionen der Übertragungstabelle. Aus der Übertragungstabelle Tab. 3.2 können nun die hier gezeigten 24 Reaktionen gewonnen werden. Die Zellen mit * wurden bei der Übersetzung in die Reaktionen ausgelassen, da die betreffenden Spezies in den Reaktionen als Katalysatoren nicht benötigt werden.

3.2 Fallstudie

Für die Fallstudie wurde ein Vergleich von zwei zweistelligen Ternärzahlen durch das vorgestellte Digitalcomputermodell durchgeführt. Der Vergleich

$$12_3 \leq 21_3 \text{ oder } 5_{10} \leq 7_{10}$$

wurde zu diesem Zweck gewählt. Dafür wurden die Anfangskonzentrationen $[x_0M]$, $[x_1H]$, $[y_0H]$, $[y_1M]$ und $[zL]$ auf $1\frac{\text{mol}}{\text{l}}$ angehoben und die Konzentrationen aller anderen Spezies auf $0\frac{\text{mol}}{\text{l}}$ herabgesetzt, da

$$[x_0M] = 1\frac{\text{mol}}{\text{l}};\ [x_1H] = 1\frac{\text{mol}}{\text{l}} \equiv x_0 = 1_3;\ x_1 = 2_3 \equiv x = 12_3 \equiv 5_{10}$$

und

$$[y_0H] = 1\frac{\text{mol}}{\text{l}};\ [y_1M] = 1\frac{\text{mol}}{\text{l}} \equiv y_0 = 2_3;\ y_1 = 1_3 \equiv y = 21_3 \equiv 7_{10}$$

Die Startkonzentration $[zL]$ wurde im Gegensatz zu $[zH]$ auf $1\frac{\text{mol}}{\text{l}}$ festgelegt, da dies einen intuitiveren Startzustand repräsentiert. Sollte der Vergleich die Ausgabe $True$ erzeugen, sinkt die Konzentration von zL wieder und im gleichen Zug steigt die von zH. Wenn der Vergleich dagegen die Ausgabe $False$ erzeugt, bleiben die Konzentrationen $[zL]$ und $[zH]$ unverändert. Bei diesem Vergleich wurde

$$x_0M + y_0H + zL \longrightarrow x_0M + y_0H + zH$$

von den 24 in Copasi eingepflegten chemischen Reaktionen verwendet. Die Konzentration $[zH]$ stieg im laufe der Berechnung, während in gleichem Maße die Konzentration $[zL]$ fiel (siehe Tab. A.1 und Fig. 3.1).

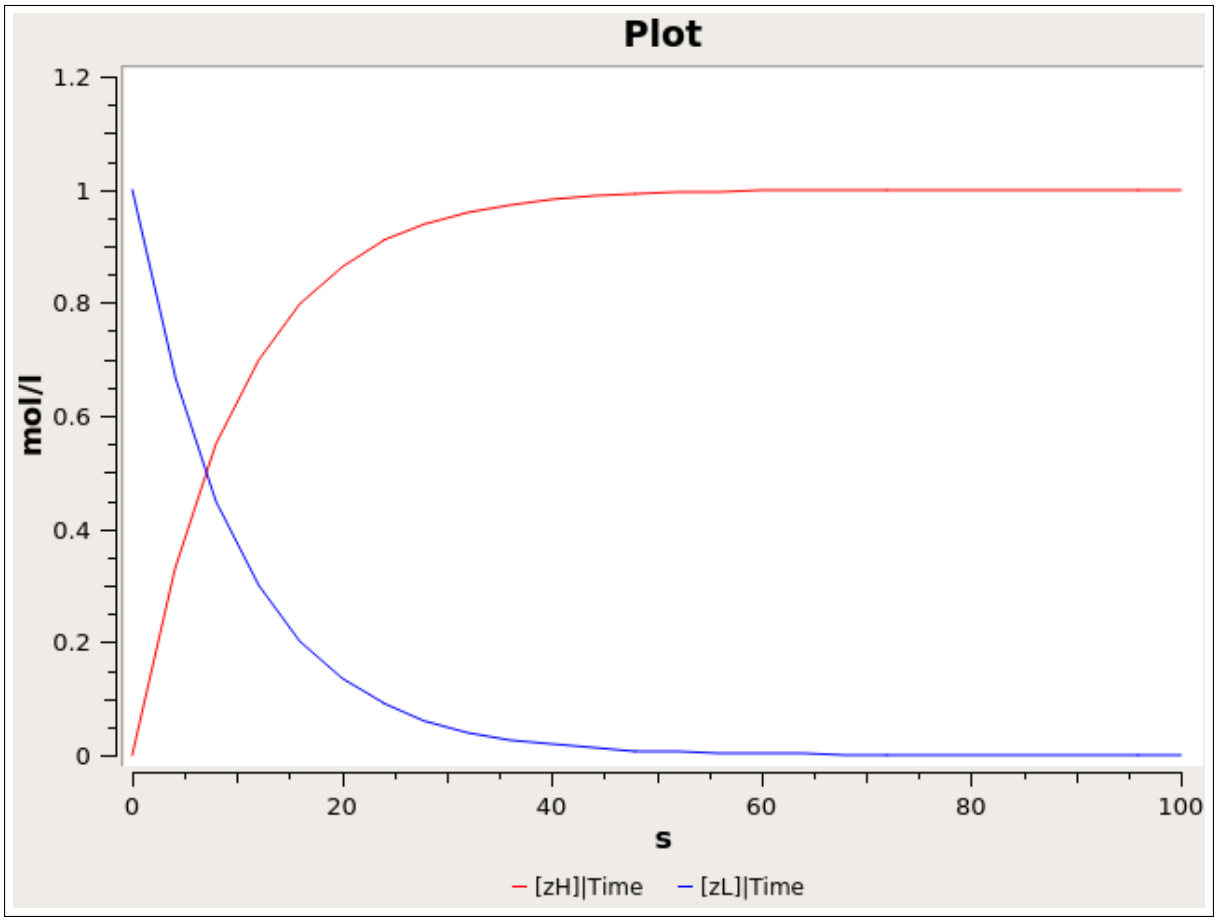

Fig. 3.1 Änderung der Konzentrationen $[zH]$ und $[zL]$. Bei diesem Plot wurde die Zeit $t[s]$ gegen die Konzentration von zH und zL aufgetragen. Während die blaue Linie, die $[zL]$ repräsentiert abfällt, steigt in gleichem Maße die rote Linie von $[zH]$

4 Diskussion

Bei der Fallstudie in Unterabs. 3.2 wurde der Vergleich

$$12_3 \leq 21_3$$

durch Simulation der 24 chemischen Reaktion in Copasi durchgeführt. Für den Vergleich wurde anhand der vorhandenen Spezies die Reaktion

$$x_0M + y_0H + zL \longrightarrow x_0M + y_0H + zH$$

genutzt, da die einstelligen Ternärzahlen $x_0 = 1_3$ und $y_0 = 2_3$ sind, gibt es somit keine Kombination der Aktivierungszustände von x_1 und y_1 die nicht zu einer Änderung des Aktivierungszustands von zL zu zH führen würde. Die Spezies x_0M und x_0H wirken dabei als Katalysatoren der Reaktion und behalten ihren eigenen Aktivierungszustand bei.
Da am Ende der Reaktion

$$[zH] \gg [zL]$$

und damit $True$ ausgegeben wurde, konnte auf eine erfolgreiche Anwendung des Digitalcomputermodells mit dem gewünschten Ergebnis geschlossen werden.
Da die 24 Reaktionen aus der Ausgabetabelle des Skripts allerdings noch eigenständig in Copasi übertragen wurden, kann hier durch Erweiterung des Skripts in eine Copasi gerechte Ausgabe, noch eine weitere Optimierung ausgeschöpft werden. Somit könnten mögliche Fehler bei dem aufwendigen Kopiervorgang der Reaktionen verhindert werden.
Außerdem könnte durch Anlegen entsprechender Klassen in Python eine Erweiterung des Skripts erfolgen, die die Auswertung sämtlicher Kombinationen von Ternärzahlen mit einschließt und nicht nur auf zweistellige ternäre Zahlen ausgelegt ist.
Auch wäre eine Durchführung des Experiments unter Laborbedingungen, anstelle einer Simulation, von großem Interesse, um dort möglicherweise auftretende Herausforderungen und Erfolge näher zu beleuchten.
Zu guter Letzt wäre auch eine bessere Kombination von Python und Copasi von größerem Interesse, da hierbei alle 81 Reaktionen in einer entsprechenden Schleife nacheinander untersucht werden könnten, um die vollständige Richtigkeit der minimalen Übertragungstabelle zu testen.

Literaturverzeichnis

Brusentsov, NP und Ramil Alvarez, J (2011). „Ternary Computers: The Setun and the Setun 70“. In: *Perspectives on Soviet and Russian Computing*. Hrsg. von J Impagliazzo und E Proydakov. Bd. 357. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 74–80. DOI: 10.1007/978-3-642-22816-2_10.

Hinze, T (März 2013). *Computer der Natur*. Friedrich-Schiller-Universität Jena und Brandenburgische Technische Universität Cottbus.

Hoops, S, Sahle, S, Gauges, R, Lee, C, Pahle, J, Simus, N, Singhal, M, Xu, L, Mendes, P und Kummer, U (Dez. 2006). „COPASI–a COmplex PAthway SImulator“. In: *Bioinformatics* 22.24, S. 3067–3074. DOI: 10.1093/bioinformatics/btl485.

Oliphant, TE (Dez. 2006). *A guide to NumPy*. Trelgol Publishing USA.

Van Rossum, G und Drake Jr, FL (Mai 1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam.

A Anhang

Tab. A.1 Datenpunkte Copasi

Time	$[zH]$	$[zL]$
0	0	1
4	0.32968	0.67032
8	0.550671	0.449329
12	0.698806	0.301194
16	0.798104	0.201896
20	0.864665	0.135335
24	0.909282	0.0907178
28	0.93919	0.06081
32	0.959238	0.0407621
36	0.972676	0.0273237
40	0.981684	0.0183156
44	0.987723	0.0122773
48	0.99177	0.00822972
52	0.994483	0.00551654
56	0.996302	0.00369785
60	0.997521	0.00247874
64	0.998338	0.00166155
68	0.998886	0.00111377
72	0.999253	0.000746582
76	0.9995	0.000500449
80	0.999665	0.000335461
84	0.999775	0.000224866
88	0.999849	0.000150732
92	0.999899	0.000101039
96	0.999932	$6.772 \cdot 10^{-5}$
100	0.999955	$4.539 \cdot 10^{-5}$