



FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Projekt 2: Chemisches Rechnen

Entwicklung und Simulation eines chemischen Analogcomputermodells zur Eins-aus-Fünf-Auswahl

Jannes Spangenberg, Sandra Triebel

betreut von

PD Dr.-Ing. habil. Thomas HINZE

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

8. Juli 2020

Inhaltsverzeichnis

1	Einleitung	4
2	Materialien und Methoden	5
3	Ergebnisse und Diskussion	11
	Anhang	I
	Literaturverzeichnis	VIII

Abbildungsverzeichnis

1	Erster Teil des Netzwerkes	9
2	Evaluation der Substraktion	10
3	Reaktionsnetzwerk: 1 aus 5 Programm	12
4	Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 0$	I
5	Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 0$	I
6	Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 1$	II
7	Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 1$	II
8	Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 2$	III
9	Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 2$	III
10	Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 3$	IV
11	Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 3$	IV
12	Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 4$	V
13	Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 4$	V
14	Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 5$	VI
15	Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 5$	VI
16	Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 6$	VII
17	Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 6$	VII

Tabellenverzeichnis

1	Konzentrationsänderungen der Spezies R_0, \dots, R_5 je nach Wert des Inputs X	6
2	Konzentrationsänderungen der Spezies A, B, B', \dots, E, E' und $Null$ je nach Wert des Inputs X	7
3	Endkonzentrationen der Spezies A, B, B', \dots, E, E' und $Null$ je nach Wert des Inputs X	8

1 Einleitung

In Lebewesen laufen tausende von chemischen Reaktionen ab, um Stoffe zu bilden, zu verändern oder abzubauen und weitere Reaktionen zu starten. Diese Mechanismen zielen darauf ab, dass der Organismus optimal und effizient unter seinen Umwelteinflüssen und seiner Umwelt arbeitet. Informationsverarbeitung ist also ein Merkmal des Lebens und somit lässt sich sagen „In jedem Lebewesen steckt ein leistungsfähiger Computer“ [1].

Dieses Prinzip möchte man nutzen, um mithilfe geeigneter chemischer Reaktionen oder physikalischer Prozesse Moleküle zielgerichtet zu verändern und auf diese Weise Berechnungen auszuführen. Das junge und interdisziplinäre Fachgebiet *Molecular Computing* beschäftigt sich genau mit dieser Thematik - Informationsverarbeitende Systeme, die ausschließlich oder zu großen Teilen aus Molekülen und molekularen Strukturen bestehen, welche als Datenträger oder Speichermedien betrachtet werden und dessen Weiterverarbeitung dabei algorithmisch beschrieben werden kann. Diese Verarbeitung basiert auf chemischen Reaktionen, Diffusions- und/oder molekularen Transportprozessen [1].

Wir bedienen uns nun des Prinzips der Informationsverarbeitung innerhalb von Reaktionsnetzwerken, wobei die Moleküle die Speichermedien darstellen. In dieser Projektarbeit wird ein chemisches Reaktionsnetzwerk mit Massenwirkungskinetik erstellt, welches das Verhalten des folgenden Java-Programms nachbildet:

```
import java.util.Scanner;

public class EinsAusN {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a = 0, b = 0, c = 0, d = 0, e = 0, x;

        System.out.print("x: ");
        x = Integer.parseInt(sc.next());
        switch(x) {
            case 1: a = 1; break;
            case 2: b = 1; break;
            case 3: c = 1; break;
            case 4: d = 1; break;
            case 5: e = 1; break;
        }
        System.out.println("a: " + a + "b: " + b + "c: " + c + "d: " + d +
            "e: " + e);
    }
}
```

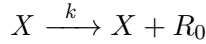
Dabei wird die Spezieskonzentration $[X]$ der Spezies X initial vorgegeben und entspricht so dem Input x des Java-Programms. Die Spezieskonzentrationen $[A]$, $[B]$, $[C]$, $[D]$ und $[E]$ der Spezies A , B , C , D und E sollen sich asymptotisch den entsprechenden Variablenwerten annähern und so dem Output des Programms und den Variablen a , b , c , d und e entsprechen.

2 Materialien und Methoden

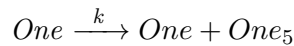
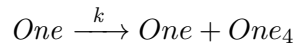
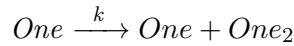
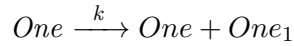
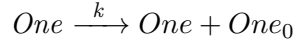
Die erste Überlegung zur Erstellung des Reaktionsnetzwerkes beruhte auf der Idee der nichtnegativen Subtraktion. Dabei werden für zwei Spezies X_1, X_2 Anfangskonzentrationen festgelegt und innerhalb des Reaktionsnetzwerkes wird die Konzentration der Output-Spezies auf $[X_1] - [X_2]$ gesetzt, wobei $[X_1] > [X_2]$ gilt [1].

Diese Idee wurde dann sukzessiv erweitert, um mittels Hilfspezies nach und nach jeweils 1 von der Konzentration der Input-Spezies abzuziehen.

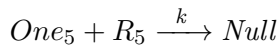
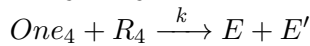
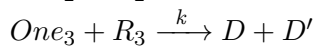
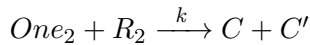
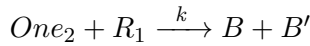
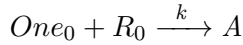
Die Input-Spezies und deren Konzentration wird kopiert und R_0 genannt.



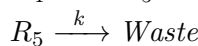
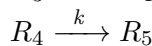
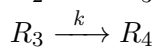
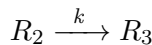
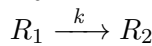
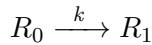
Weiterhin wird eine Spezies mit Konzentration 1 erstellt und mehrmals kopiert, um für jeden Teil des `switch case` eine *One*-Spezies zu haben. Diese 6 Reaktionen werden aufgrund der Trivialität grafisch nicht gezeigt.



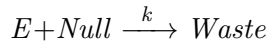
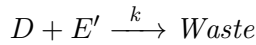
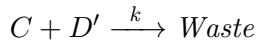
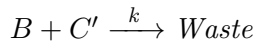
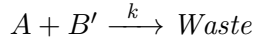
Subtraktion $R_x - 1, \forall x \in \{0, 1, \dots, 5\}$ und Speicherung des Wertes 1 in den Output-Spezies Für die Auswertung des Outputs wird für B, C, D und E eine Kopie erstellt. *Null* steht dabei für die Auswertung, wenn der Input größer ist als 5.



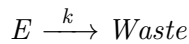
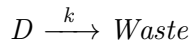
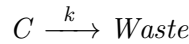
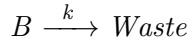
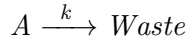
Nach der Subtraktion wird der Rest von R_x in R_{x+1} gespeichert $\forall x \in \{0, 1, \dots, 5\}$, wobei $R_6 = Waste$.



Im Nachhinein wird der Output ausgewertet.



Zum Schluss werden Reaktionsgleichungen in das Netzwerk integriert, um einen Überlauf für unsere Output-Spezies zu vermeiden und einen stationären Zustand zu gewährleisten.



Das komplette Reaktionsnetzwerk ist in Abbildung 3 zu sehen.

Die Idee hinter unserer Implementierung ist, dass wir unseren Input zuerst in sechs verschiedene Werte aufteilen. Diese sind R_0, R_1, \dots, R_5 . R_0 nimmt dabei den Wert vom Input X an. $R_1 = R_0 - 1$, $R_2 = R_1 - 1$ und so weiter. Die Skizze des Netzwerkes kann in Abbildung 1 betrachtet werden. Dies resultiert in den folgenden Konzentrationen für entsprechende Inputs:

Input $[X]$	$[R_0]$	$[R_1]$	$[R_2]$	$[R_3]$	$[R_4]$	$[R_5]$
0	0	0	0	0	0	0
1	1	0	0	0	0	0
2	2	1	0	0	0	0
3	3	2	1	0	0	0
4	4	3	2	1	0	0
5	5	4	3	2	1	0
> 5	> 5	> 4	> 3	> 2	> 1	> 0

Tabelle 1. Konzentrationsänderungen der Spezies R_0, \dots, R_5 je nach Wert des Inputs X

Die Subtrahenden werden verwendet, um eine Inputauswertung auf boolescher Ebene zu ermöglichen. Die Werte $\text{Spezies}'$ ist dabei eine Kopie von Spezies . Für die entsprechenden Werte der Subtrahenden, welche in A, B, B', \dots, E, E' und Null gespeichert werden, ergeben sich folgende Konzentrationswerte:

Input $[X]$	$[A]$	$[B] = [B']$	$[C] = [C']$	$[D] = [D']$	$[E] = [E']$	$[Null]$
0	0	0	0	0	0	0
1	1	0	0	0	0	0
2	1	1	0	0	0	0
3	1	1	1	0	0	0
4	1	1	1	1	0	0
5	1	1	1	1	1	0
> 5	1	1	1	1	1	> 1

Tabelle 2. Konzentrationsänderungen der Spezies A, B, B', \dots, E, E' und $Null$ je nach Wert des Inputs X

Da für unsere Spezies gilt, wie man in der oben stehenden Tabelle sehen kann, wenn für eine $[Spezies] = [Spezies'] = 1$ gilt, dann gilt auch für alle Spezies welche alphabetisch kleiner sind $[.] = 1$. Das gleiche gilt umgekehrt, wenn $[Spezies] = [Spezies'] = 0$, dann gilt für die alphabetisch größeren Spezies, $[.] = 0$.

Für unser Netzwerk gilt dann also:

$$[X] = 0 \Leftrightarrow [A] = [B] = \dots = [E] = [Null] = 0$$

$$[X] = 1 \Leftrightarrow [A] = 1 \wedge [B] = 0$$

$$[X] = 2 \Leftrightarrow [B] = 1 \wedge [C] = 0$$

$$\vdots$$

$$[X] = 5 \Leftrightarrow [D] = 1 \wedge [E] = 0$$

$$[X] > 5 \Leftrightarrow [E] = 1 \wedge [Null] \geq 1$$

Dies lässt sich im Reaktionsnetzwerk implementieren, indem man wieder eine Subtraktion anwendet auf alphabetische aufeinanderfolgende Paare von Spezies. Im Fall $[X] = 1$ rechnet man also $[A] - [B']$. $[A]$ bleibt 1, wenn $[B]$ gleich 0 ist. Das ist nur der Fall, wenn $[X] = 1$. Falls $[X]$ größer ist als 1, ist zwangsläufig $[B] = 1$. Ist $[X]$ gleich 0, dann ist auch $[A] = 0$.

Für den Fall $[X] = 2$, rechnen wir $[B] - [C']$ und $[A] - [B']$. $[A]$ wird dann zu 0 und $[B]$ bleibt 1. Wir können auch die anderen paarweisen Subtraktionen ohne Einschränkungen parallel laufen lassen. Im Fall $[X] = 2$ gilt für $[C] - [D'] \Leftrightarrow 0 - 0$. Es entstehen also keine „bösen“ Seiteneffekte und wir benötigen keine Einschränkungen abhängig vom Input.

Diese Subtraktion wendet man demnach auf alle Paare an. Hierfür brauchen wir die Kopien $[B'], [C'], \dots$. Da wir ansonsten beim doppelten Rechnen mit einer Spezies ohne Kopie Information bzw. Stoffkonzentrationen verlieren würden. Zu guter letzt wird die Spezies $[Null]$ benötigt, um Eingaben $[X] > 5$ abzufangen. Diese werden in $[Null]$ verarbeitet, so dass unsere Ausgabe $[E] = 0$ wird für $[X] > 5$. Das hier beschriebene Vorgehen wird in Abbildung 2 visualisiert. Aus der obigen Tabelle ist dann ersichtlich, dass nach anwenden dieser Subtraktionen folgende Stoffkonzentrationen für unsere Spezies entstehen:

Input $[X]$	$[A]$	$[B] = [B']$	$[C] = [C']$	$[D] = [D']$	$[E] = [E']$	$[Null]$
0	0	0	0	0	0	0
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
> 5	0	0	0	0	0	> 1

Tabelle 3. Endkonzentrationen der Spezies A, B, B', \dots, E, E' und $Null$ je nach Wert des Inputs X

Dies entspricht bereits unserem gewollten Output des vorgestellten Java Programms.

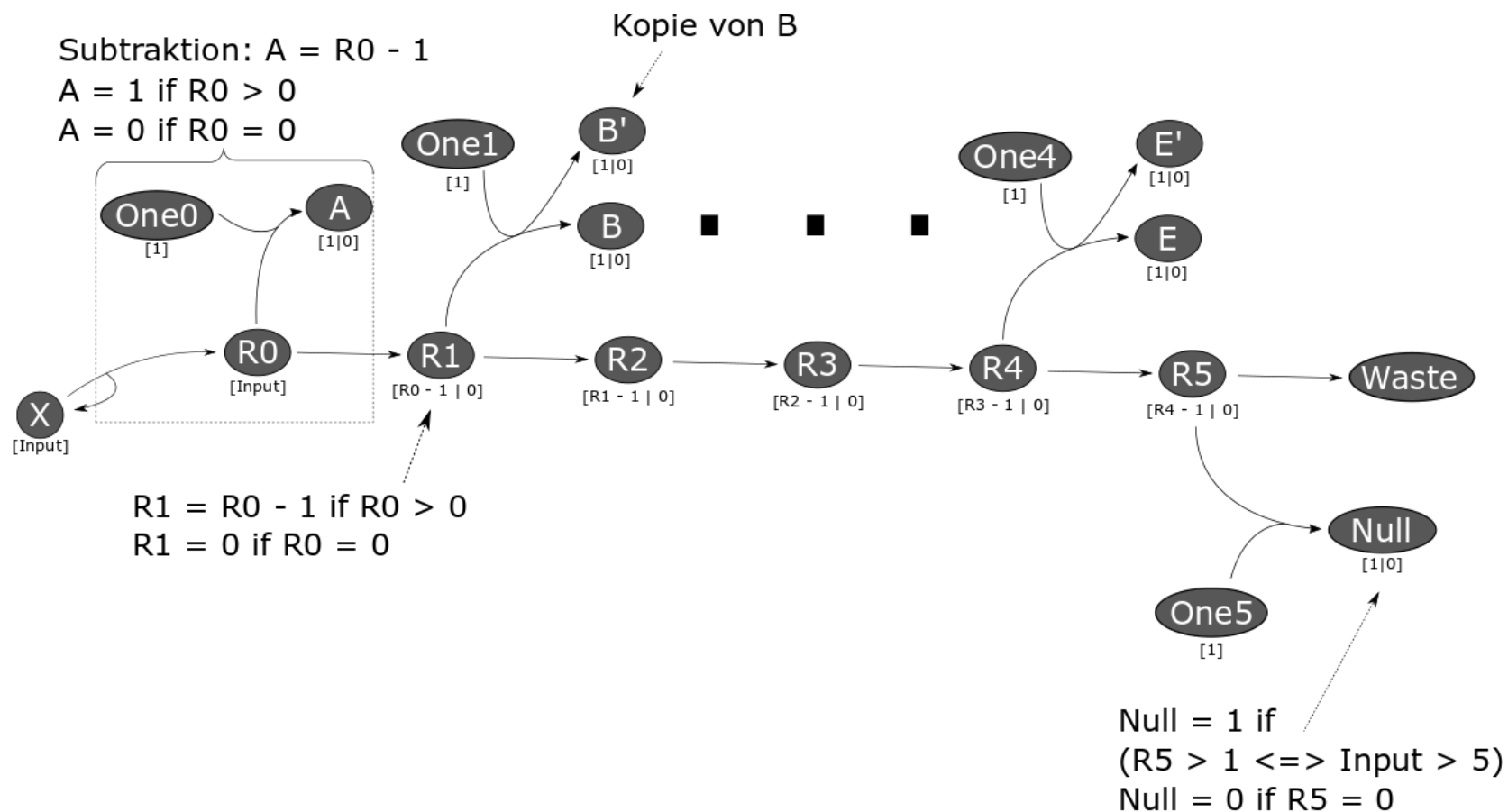


Abbildung 1. Erster Teil des Netzwerkes: Hier wird die Vorbereitung für die Input Auswertung gezeigt. Der Input X wird so verarbeitet, dass sequenziell immer eine Einheit in *One* gespeichert abgezogen wird. Der Subtrahend wird dabei in A, B, B', \dots, E, E' und $Null$ gespeichert. Die Differenz von R_x wird bei uns als „Remainder“ in den R_{x+1} gespeichert $\forall x \in 0, 1, 2, 3, 4$.

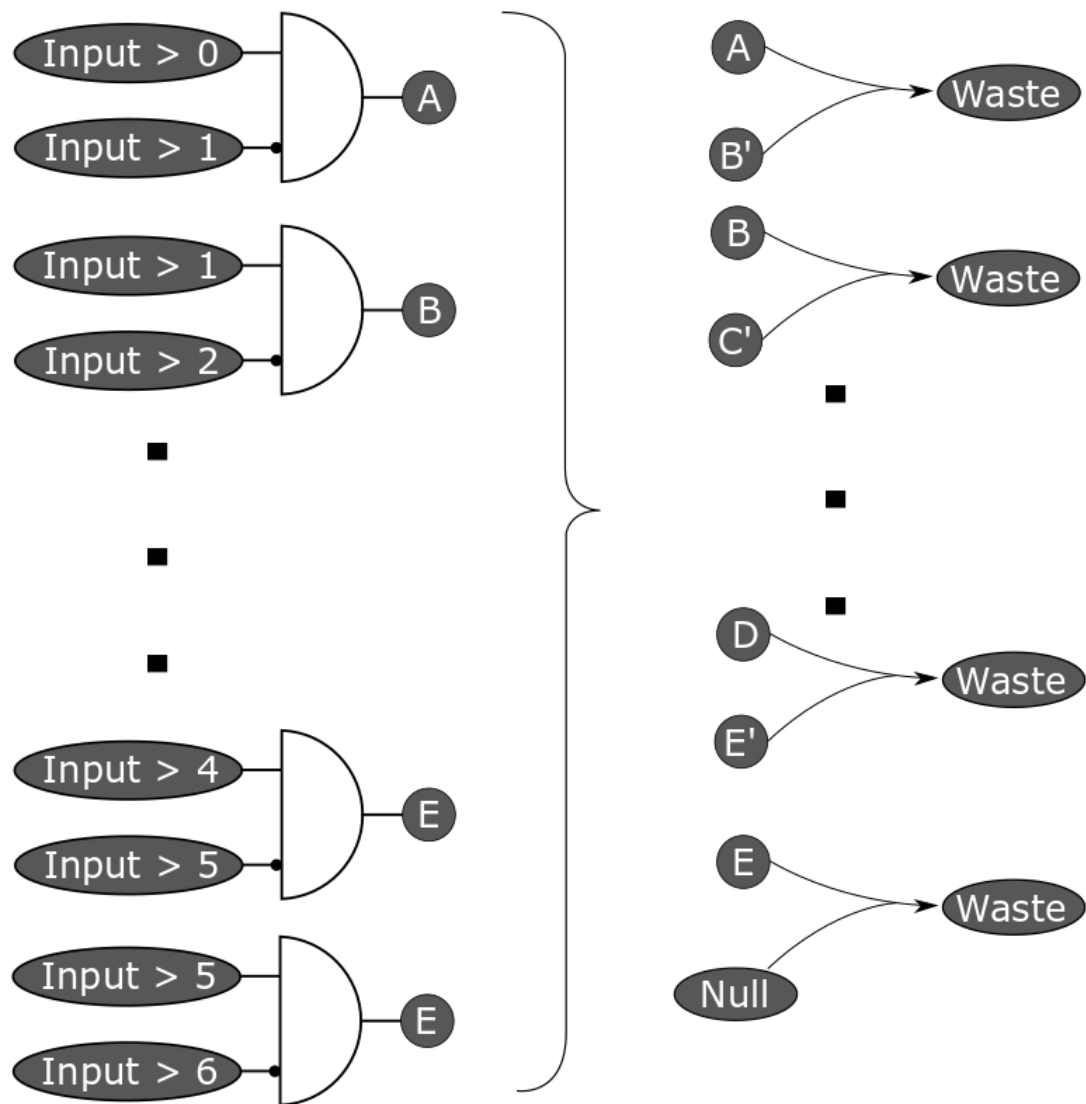


Abbildung 2. Evaluation der Subtraktion: Links ist die Auswertung des Inputs dargestellt mittels Logik Gattern. Diese können erneut leicht in eine Subtraktion umgewandelt werden (rechts), wozu die vorher erstellten Kopien der Spezies B, \dots, E benötigt werden, da ohne Kopien während der Evaluation Stoffkonzentration verloren gehen würden, weil diese Spezies zweimal ausgewertet werden müssen. Nach dieser Evaluation kann das erwünschte Ergebnis in Tabelle 3 gewährleistet werden.

3 Ergebnisse und Diskussion

Für die Subtraktion und Umwandlung des Inputs in boolesche Werte, in Abbildung 1 zu sehen, und für die boolesche Auswertung, in Abbildung 2 zu sehen, benötigen wir insgesamt 25 Spezies und 29 Reaktionen. Das komplette Netzwerk wird in Abbildung 3 gezeigt. Eine genaue Auswertung der Stoffkonzentrationen des System konnte im Programm `Copasi` nach 540 Sekunden bzw. 9 Minuten für die größten Eingaben von $[Input] = 5$ und $[Input] = 6$ festgestellt werden. Nach dieser Zeit, wie man in den Diagrammen im Anhang sehen kann, gehen die Stoffkonzentrationen in den Stationären Zustand über. Die Eingaben $[Input] = 0$ bis 6 wurden nacheinander mit `Copasi` geprüft, die Ergebnisse und Zustände der Spezies nach entsprechender Eingabe können im Anhang in den Abbildungen 4 - 17 begutachtet werden.

Für das komplette Reaktionsnetzwerk wurde im Grunde nur das Prinzip der nichtnegativen Subtraktion aus der Vorlesung verwendet [1].

25 Spezies und 29 Reaktionen sind für ein einfaches Programm mit einer `switch case` Struktur, wie hier vorgegeben, recht viel. Weiterhin kommt es zu einer großen Anhäufung von Zwischen-Spezies durch das Kopieren der Spezies; das erscheint nicht „elegant“ und speichereffizient.

Der Zeitaufwand ist verglichen mit dem Java-Programm und dessen Ausführung auch sehr hoch; 9 min vs. < 1 sec. Vorher oszillieren die verschiedenen Output-Spezies, sodass man auch vorher kein Ergebnis erwarten kann.

Pro zusätzlichen Output werden 5 Spezies und 4 neue Reaktionen benötigt. Eventuell könnte man diese weiteren Spezies und Reaktionen mit einem circulären Netzwerk verringern, sodass die Auswertung für einen beliebigen Input möglich ist.

Offen bleibt die Frage, ob sich mit höherem Input auch die Laufzeit erhöht; das müsste man nochmals überprüfen.

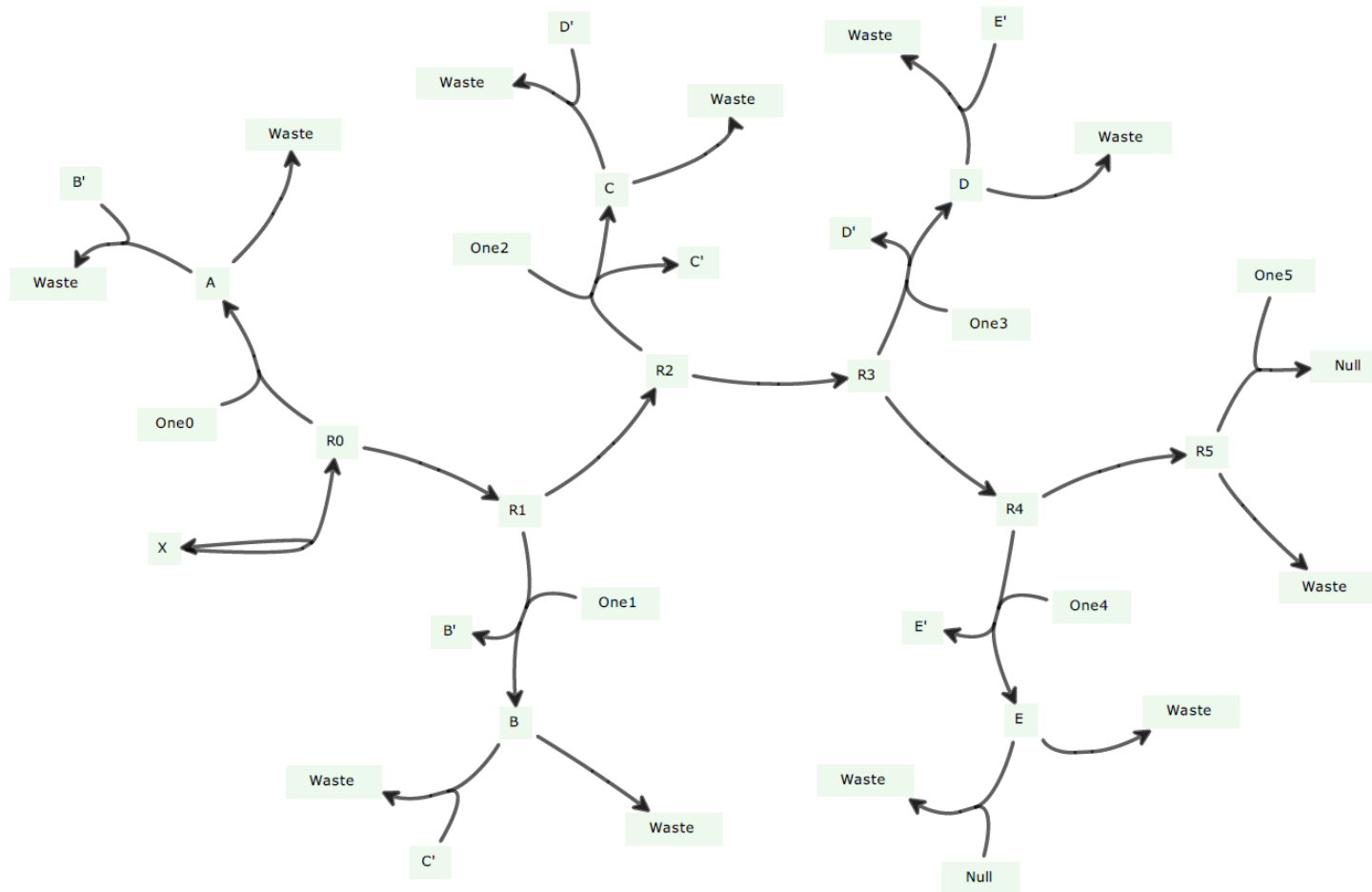


Abbildung 3. Reaktionsnetzwerk: 1 aus 5 Programm - In dem Bild sieht man das Reaktionsnetzwerk ohne die Neuerzeugung des One_x Spezies durch One des Programms 1 aus 5 mit 24 Spezies (One nicht gezeigt, insgesamt 25 Spezies) und 23 Reaktionen (6 Reaktionen für One_x Spezies nicht gezeigt, insgesamt 29 Reaktionen).

Anhang

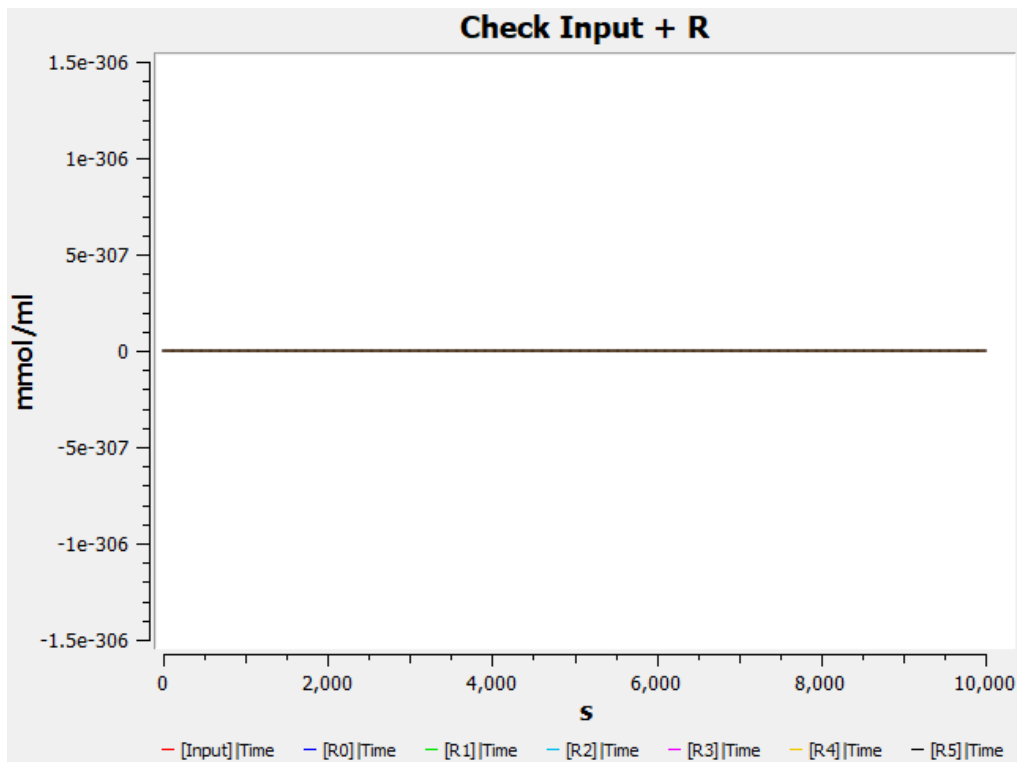


Abbildung 4. Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 0$: Das Diagramm zeigt die Konzentrationen vom Input und den Remainern R_0, \dots, R_5 .

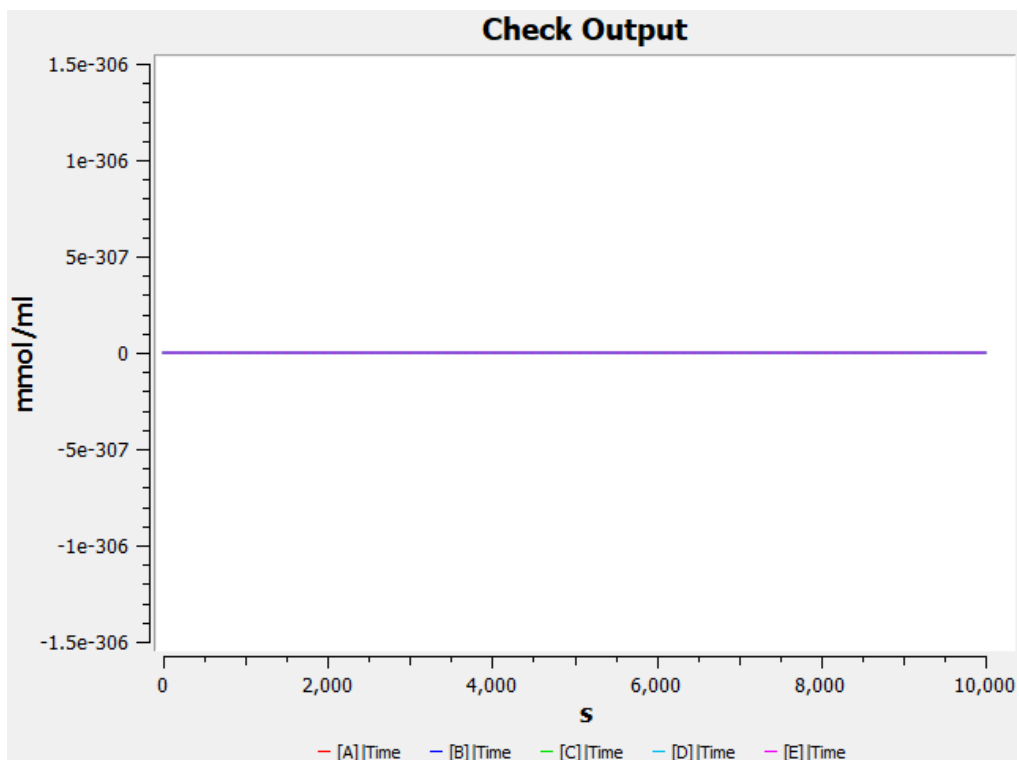


Abbildung 5. Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 0$: Das Diagramm zeigt die Konzentrationen der Output Spezies A, \dots, E und zusätzlich die *Null* Spezies, welche bei Input > 5 eine steigende Stoffkonzentration aufweist.

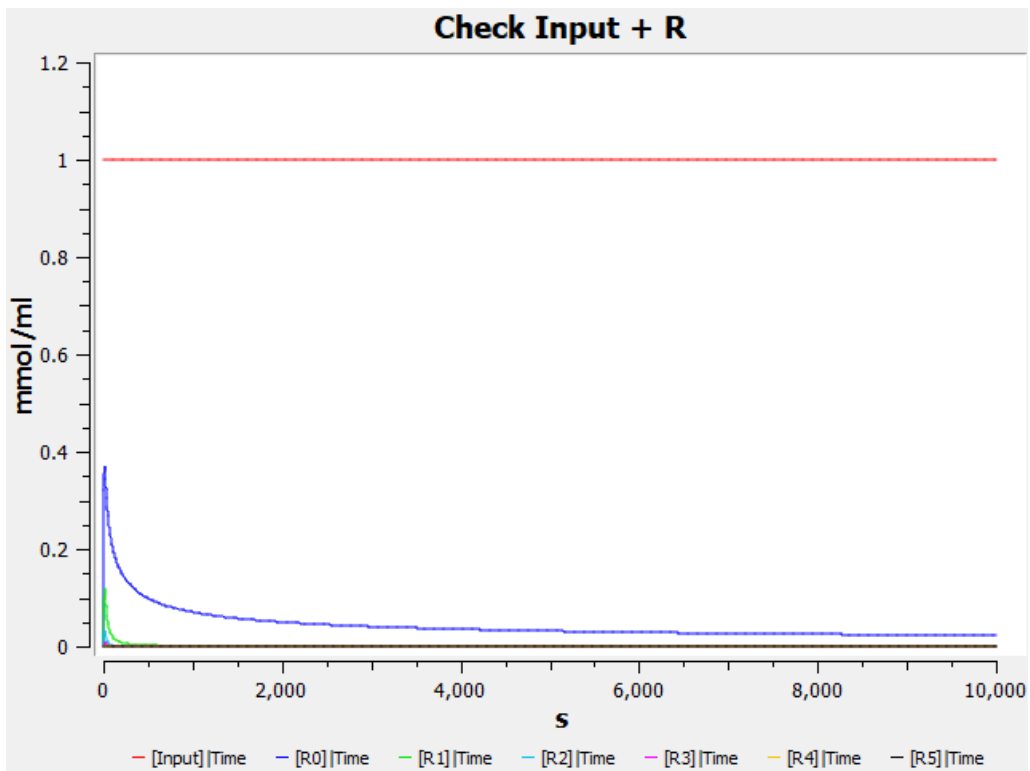


Abbildung 6. Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 1$: Das Diagramm zeigt die Konzentrationen vom Input und den Remainern R_0, \dots, R_5 .

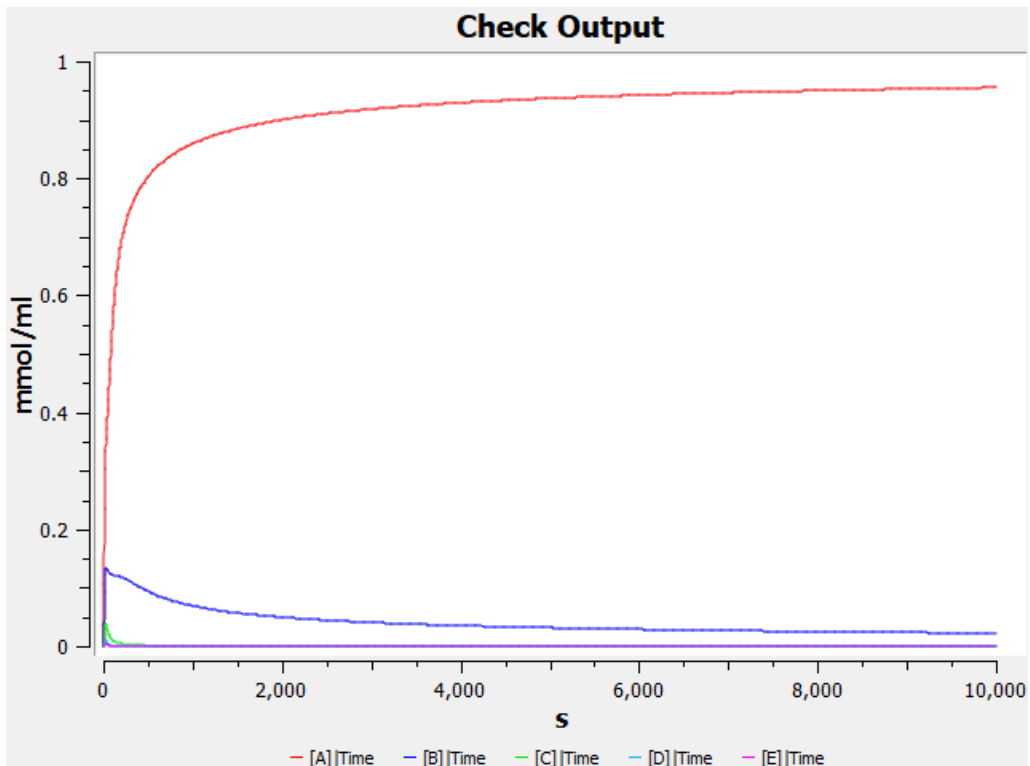


Abbildung 7. Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 1$: Das Diagramm zeigt die Konzentrationen der Output Spezies A, \dots, E und zusätzlich die *Null* Spezies, welche bei Input > 5 eine steigende Stoffkonzentration aufweist.

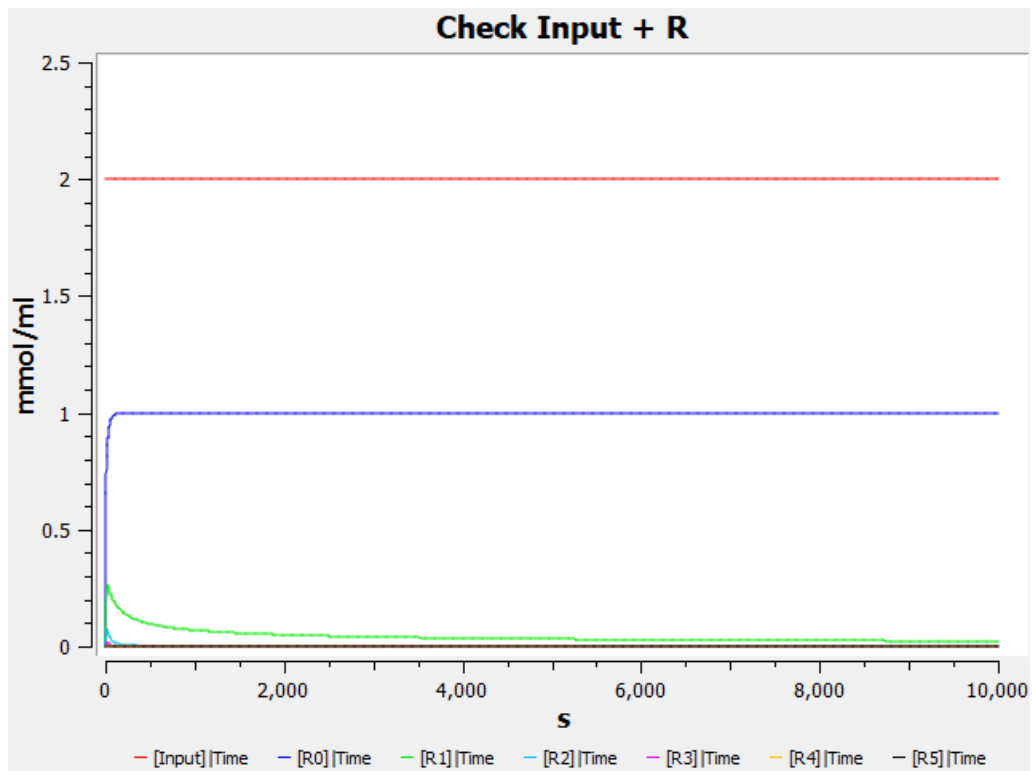


Abbildung 8. Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 2$: Das Diagramm zeigt die Konzentrationen vom Input und den Remainern R_0, \dots, R_5 .

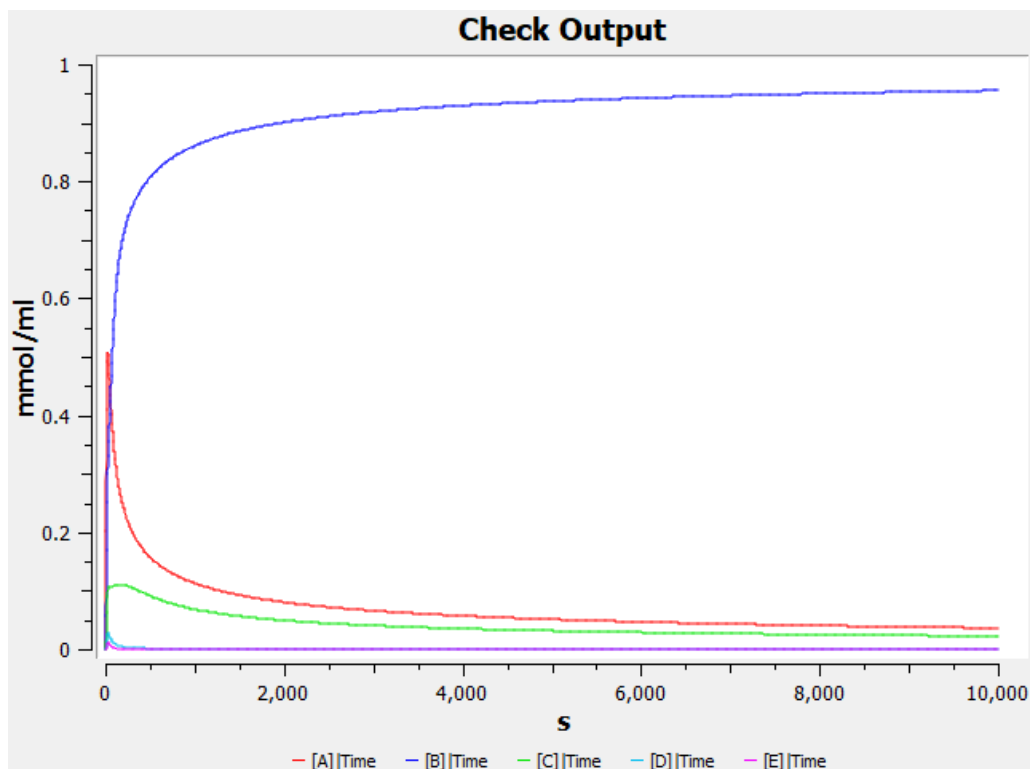


Abbildung 9. Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 2$: Das Diagramm zeigt die Konzentrationen der Output Spezies A, \dots, E und zusätzlich die *Null* Spezies, welche bei Input > 5 eine steigende Stoffkonzentration aufweist.

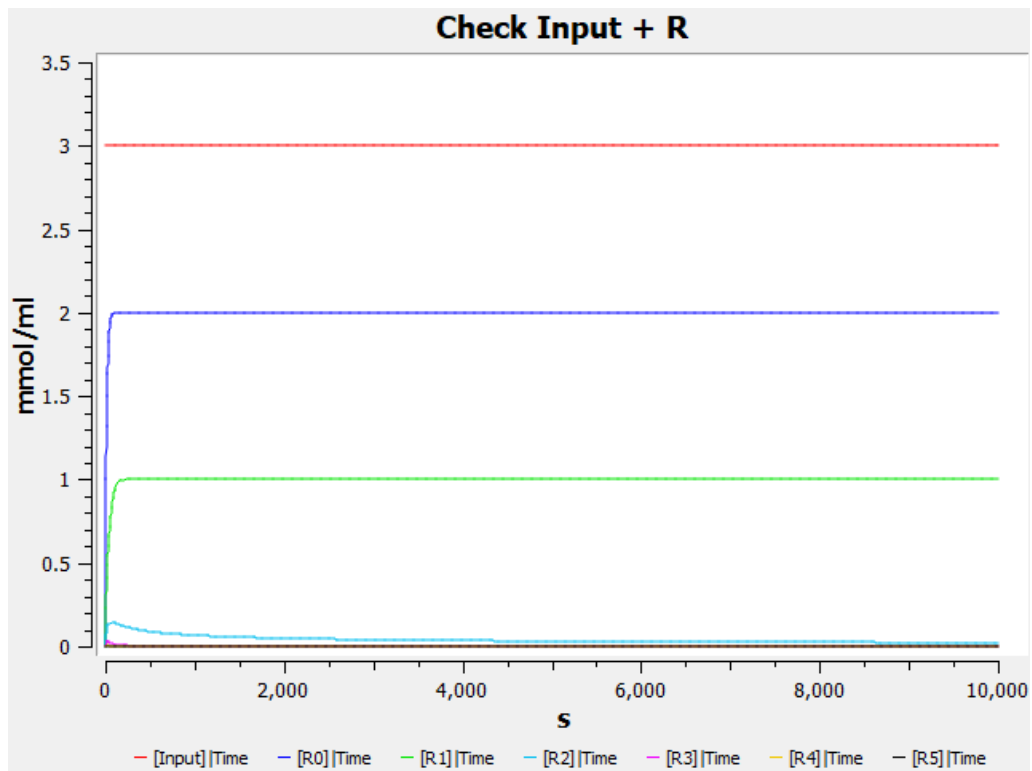


Abbildung 10. Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 3$: Das Diagramm zeigt die Konzentrationen vom Input und den Remainern R_0, \dots, R_5 .

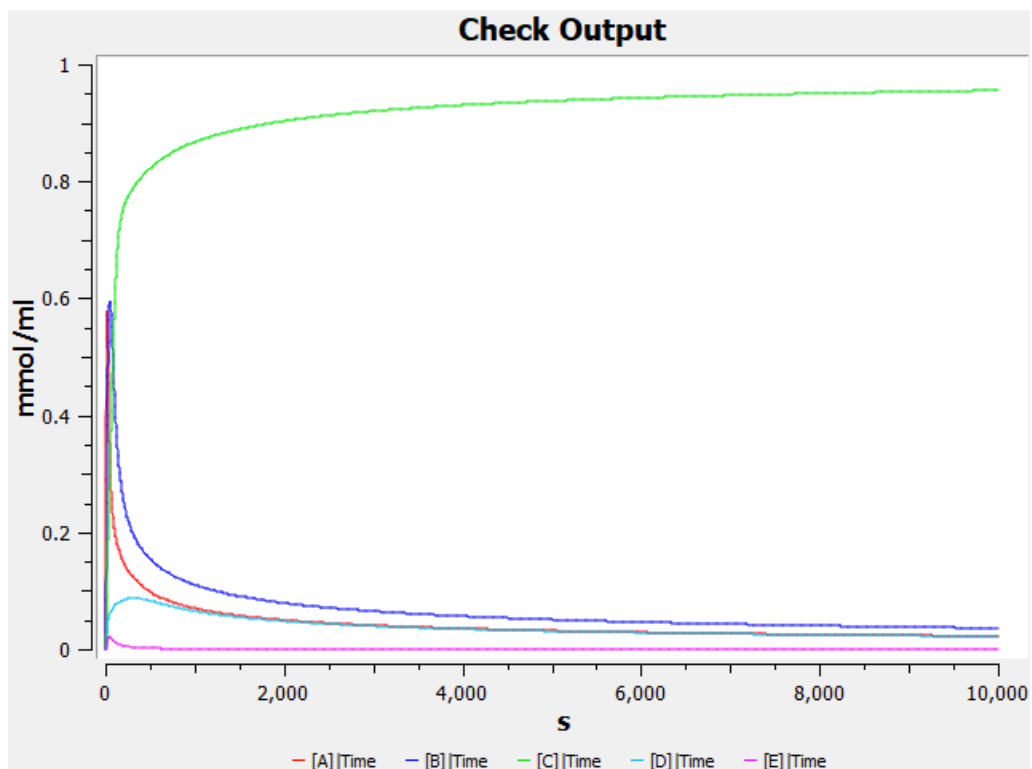


Abbildung 11. Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 3$: Das Diagramm zeigt die Konzentrationen der Output Spezies A, \dots, E und zusätzlich die *Null* Spezies, welche bei Input > 5 eine steigende Stoffkonzentration aufweist.

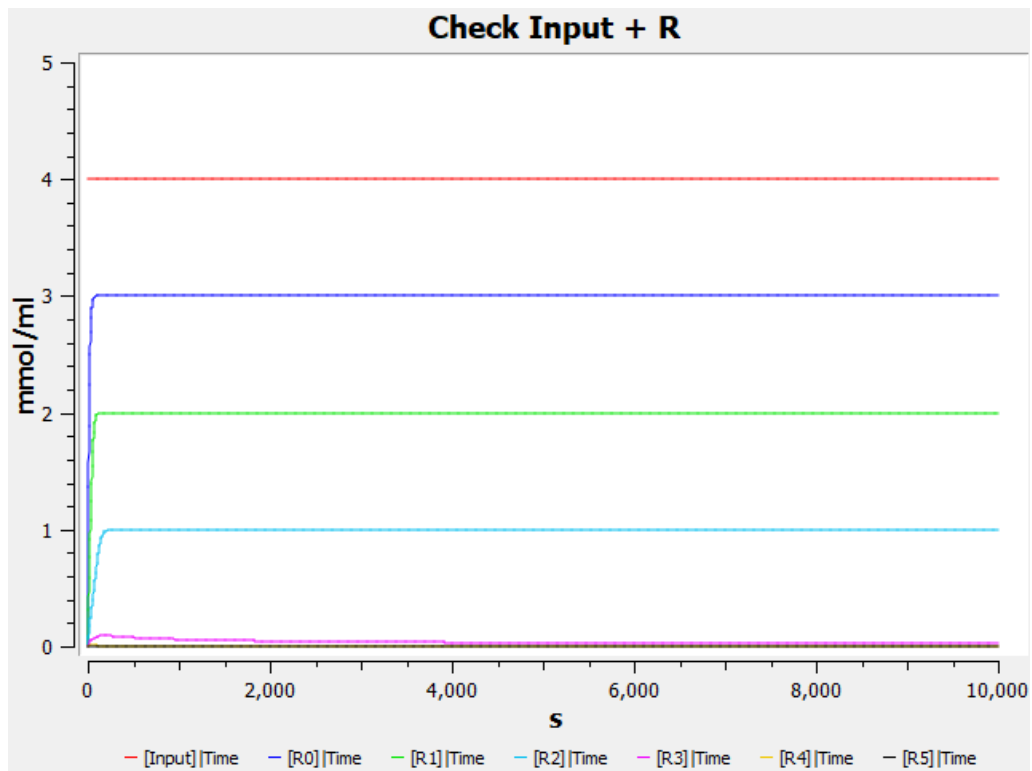


Abbildung 12. Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 4$: Das Diagramm zeigt die Konzentrationen vom Input und den Remainern R_0, \dots, R_5 .

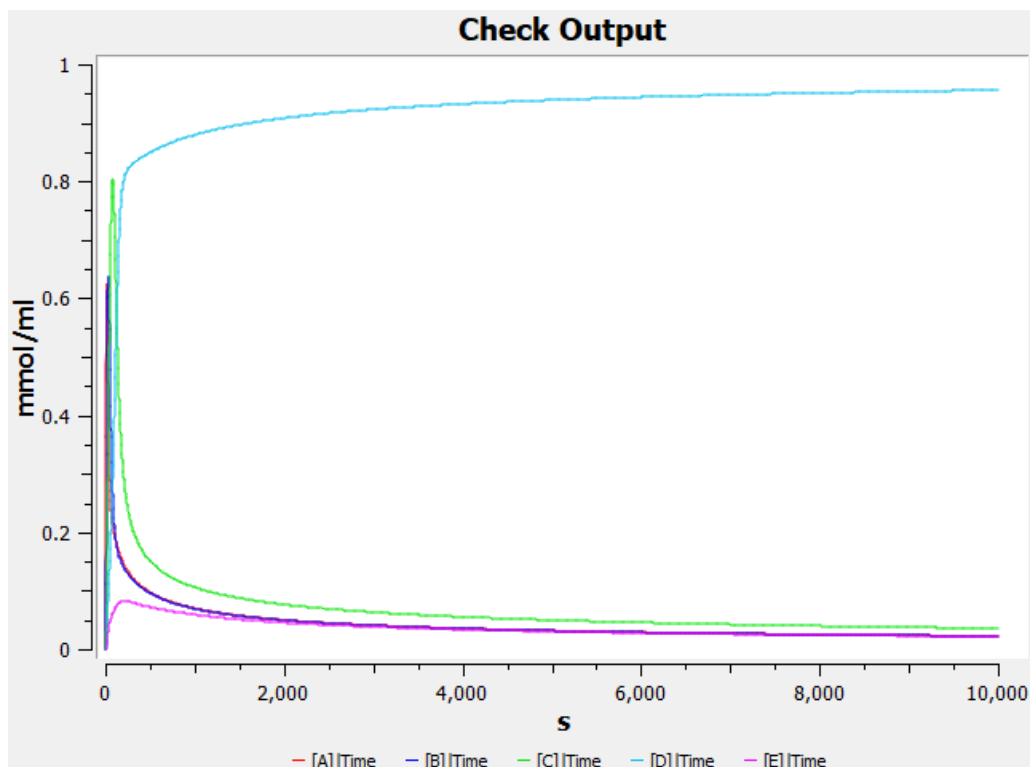


Abbildung 13. Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 4$: Das Diagramm zeigt die Konzentrationen der Output Spezies A, \dots, E und zusätzlich die *Null* Spezies, welche bei Input > 5 eine steigende Stoffkonzentration aufweist.

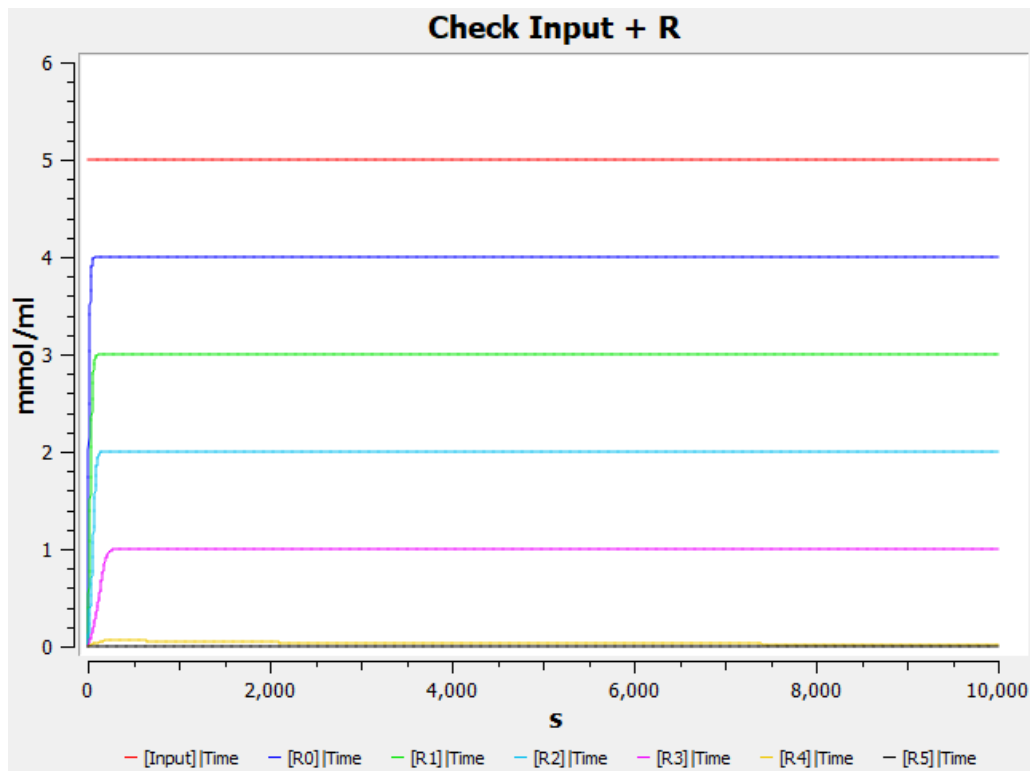


Abbildung 14. Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 5$: Das Diagramm zeigt die Konzentrationen vom Input und den Remainern R_0, \dots, R_5 .

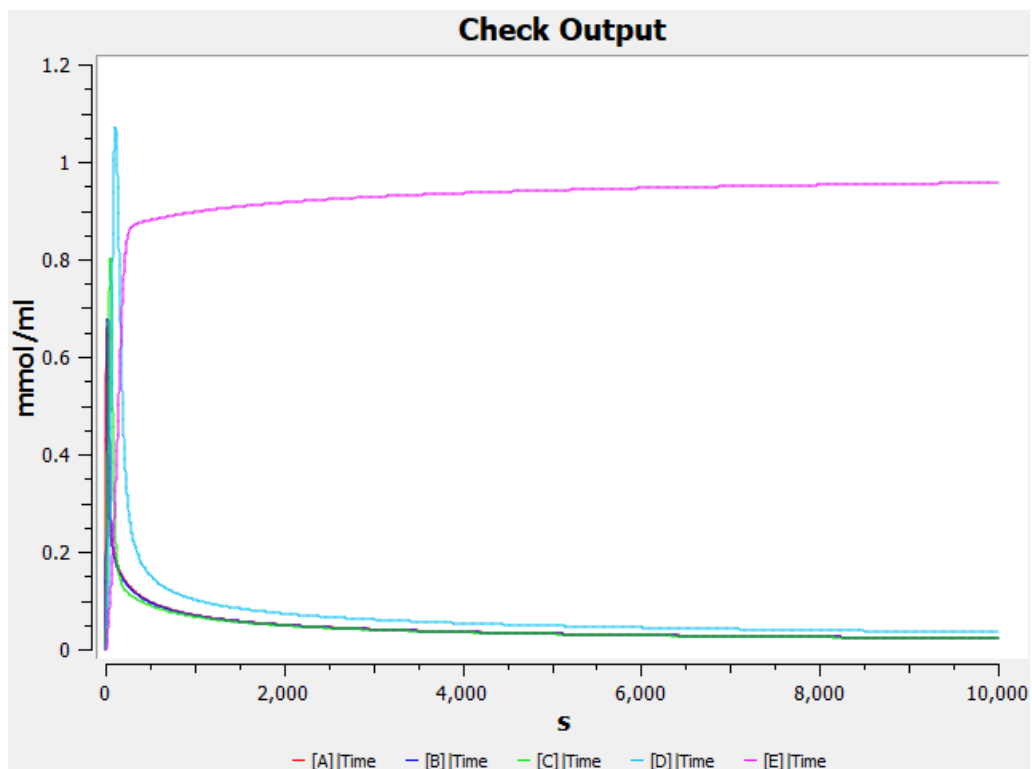


Abbildung 15. Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 5$: Das Diagramm zeigt die Konzentrationen der Output Spezies A, \dots, E und zusätzlich die *Null* Spezies, welche bei Input > 5 eine steigende Stoffkonzentration aufweist.

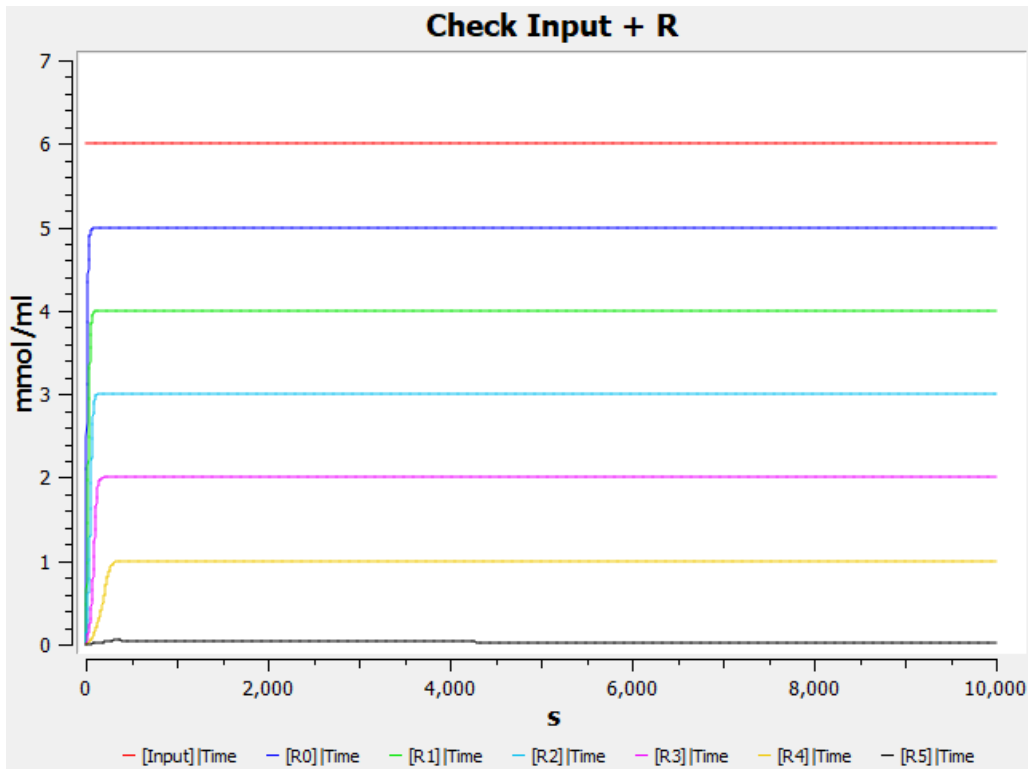


Abbildung 16. Stoffkonzentration-Zeit-Diagramm der Spezies R_x für Input $[X] = 6$: Das Diagramm zeigt die Konzentrationen vom Input und den Remainern R_0, \dots, R_5 .

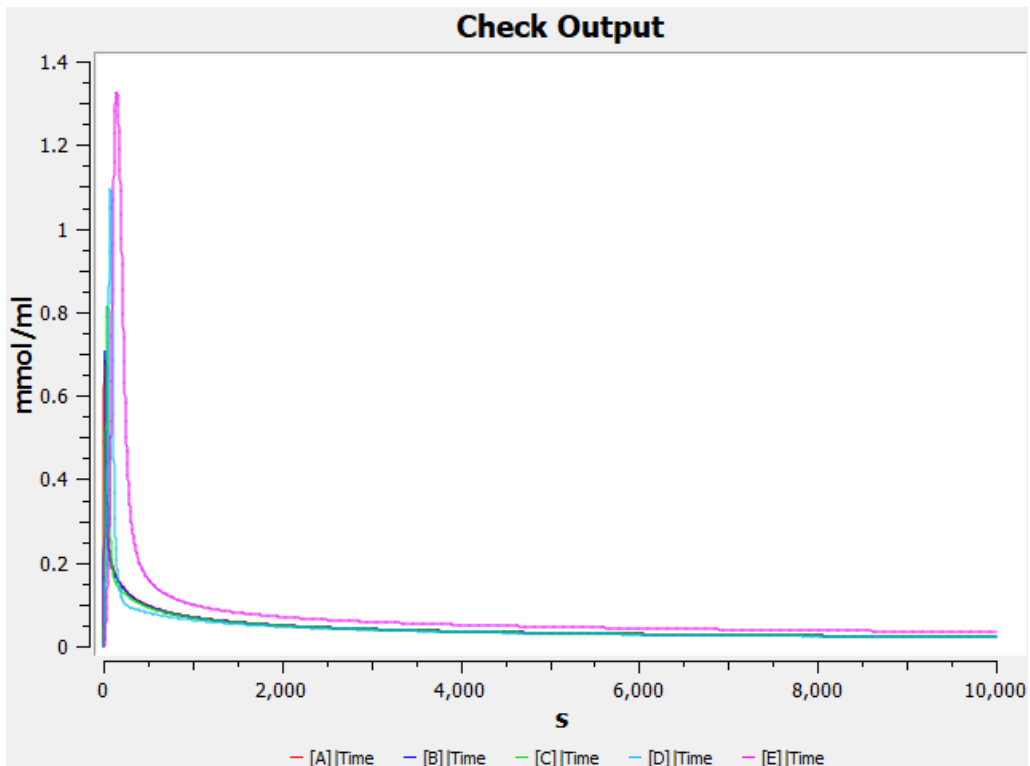


Abbildung 17. Stoffkonzentration-Zeit-Diagramm der Output-Spezies für Input $[X] = 6$: Das Diagramm zeigt die Konzentrationen der Output Spezies A, \dots, E und zusätzlich die *Null* Spezies, welche bei Input > 5 eine steigende Stoffkonzentration aufweist.

Literaturverzeichnis

- [1] Thomas Hinze. *Computer der Natur - Ausgewählte molekulare Prinzipien der biologisch inspirierten Informationsverarbeitung*. Verlag bookboon.com, London, 2013.