

Rechnen mit Molekülen

Neue Impulse und Herausforderungen einer jungen
biologisch inspirierten Informatikdisziplin

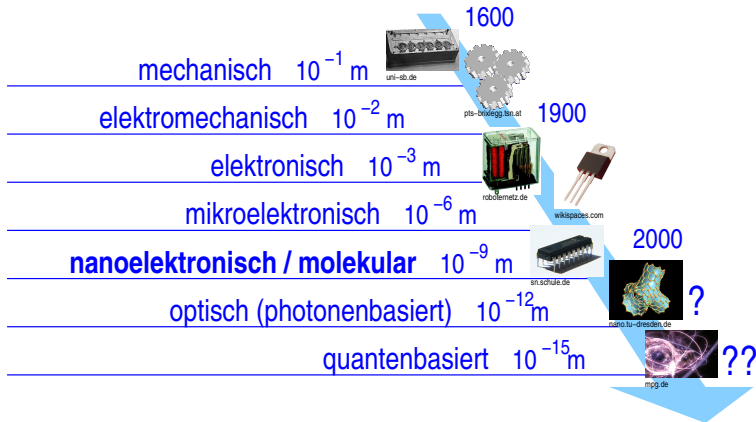
Thomas Hinze

Brandenburgische Technische Universität Cottbus
Institut für Informatik, Informations- und Medientechnik
Lehrstuhl Programmiersprachen und Compilerbau

`thomas.hinze@tu-cottbus.de`

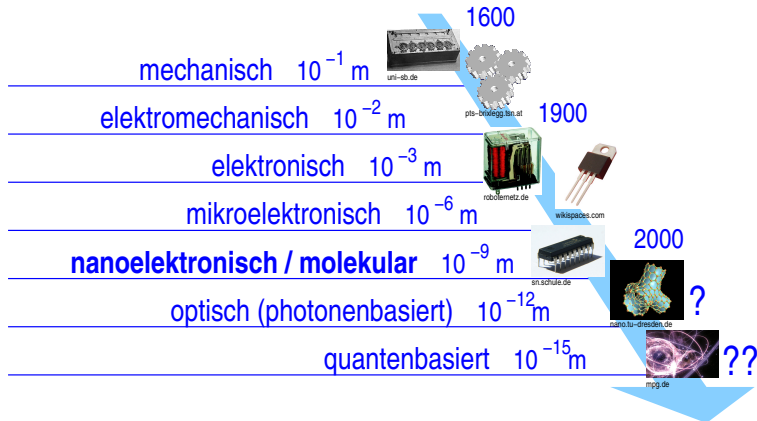
09. Juli 2012

Stoffliche Basis für Rechnerarchitekturen



⇒ ingenieurtechnische Evolution

Stoffliche Basis für Rechnerarchitekturen



⇒ **ingenieurtechnische Evolution**

Biomolekulare Speichermedien in vivo bieten

- hohe Speicherdichte bis $1 \frac{\text{Bit}}{\text{nm}^3}$
- Energieeffizienz bis $10^{18} \frac{\text{ops}}{\text{J}}$
- Dezentralität (Redundanz, Robustheit) ... bis $10^{12} \frac{\text{Zellen}}{\text{Schleimpilz}}$
- massiv parallele Verarbeitung bis $10^9 \frac{\text{Molekule}}{\text{Zelle}}$
- Selbstorganisations- und Reparaturfähigkeiten
- Biokompatibilität
- Rohstoffverfügbarkeit und Recyclingfähigkeit
- Persistenz unter geeigneten Bedingungen
- Strukturdynamik als inhärentes Beschreibungsmittel
- regelbasierte Programmierbarkeit

„In jedem Lebewesen steckt ein leistungsfähiger Computer.“

Biomolekulare Speichermedien in vivo bieten

- hohe Speicherdichte bis $1 \frac{\text{Bit}}{\text{nm}^3}$
- Energieeffizienz bis $10^{18} \frac{\text{ops}}{\text{J}}$
- Dezentralität (Redundanz, Robustheit) ... bis $10^{12} \frac{\text{Zellen}}{\text{Schleimpilz}}$
- massiv parallele Verarbeitung bis $10^9 \frac{\text{Molekule}}{\text{Zelle}}$
- Selbstorganisations- und Reparaturfähigkeiten
- Biokompatibilität
- Rohstoffverfügbarkeit und Recyclingfähigkeit
- Persistenz unter geeigneten Bedingungen
- Strukturdynamik als inhärentes Beschreibungsmittel
- regelbasierte Programmierbarkeit

„In jedem Lebewesen steckt ein leistungsfähiger Computer.“

Biomolekulare Speichermedien in vivo bieten

- hohe Speicherdichte bis $1 \frac{\text{Bit}}{\text{nm}^3}$
- Energieeffizienz bis $10^{18} \frac{\text{ops}}{\text{J}}$
- Dezentralität (Redundanz, Robustheit) ... bis $10^{12} \frac{\text{Zellen}}{\text{Schleimpilz}}$
- massiv parallele Verarbeitung bis $10^9 \frac{\text{Molekule}}{\text{Zelle}}$
- Selbstorganisations- und Reparaturfähigkeiten
- Biokompatibilität
- Rohstoffverfügbarkeit und Recyclingfähigkeit
- Persistenz unter geeigneten Bedingungen
- Strukturdynamik als inhärentes Beschreibungsmittel
- regelbasierte Programmierbarkeit

„In jedem Lebewesen steckt ein leistungsfähiger Computer.“

Biomolekulare Speichermedien in vivo bieten

- hohe Speicherdichte bis $1 \frac{\text{Bit}}{\text{nm}^3}$
- Energieeffizienz bis $10^{18} \frac{\text{ops}}{\text{J}}$
- Dezentralität (Redundanz, Robustheit) ... bis $10^{12} \frac{\text{Zellen}}{\text{Schleimpilz}}$
- massiv parallele Verarbeitung bis $10^9 \frac{\text{Moleküle}}{\text{Zelle}}$
- Selbstorganisations- und Reparaturfähigkeiten
- Biokompatibilität
- Rohstoffverfügbarkeit und Recyclingfähigkeit
- Persistenz unter geeigneten Bedingungen
- Strukturdynamik als inhärentes Beschreibungsmittel
- regelbasierte Programmierbarkeit

„In jedem Lebewesen steckt ein leistungsfähiger Computer.“

Biomolekulare Speichermedien in vivo bieten

- hohe Speicherdichte bis $1 \frac{\text{Bit}}{\text{nm}^3}$
- Energieeffizienz bis $10^{18} \frac{\text{ops}}{\text{J}}$
- Dezentralität (Redundanz, Robustheit) ... bis $10^{12} \frac{\text{Zellen}}{\text{Schleimpilz}}$
- massiv parallele Verarbeitung bis $10^9 \frac{\text{Moleküle}}{\text{Zelle}}$
- Selbstorganisations- und Reparaturfähigkeiten
- Biokompatibilität
- Rohstoffverfügbarkeit und Recyclingfähigkeit
- Persistenz unter geeigneten Bedingungen
- Strukturdynamik als inhärentes Beschreibungsmittel
- regelbasierte Programmierbarkeit

„In jedem Lebewesen steckt ein leistungsfähiger Computer.“

Biomolekulare Speichermedien in vivo bieten

- hohe Speicherdichte bis $1 \frac{\text{Bit}}{\text{nm}^3}$
- Energieeffizienz bis $10^{18} \frac{\text{ops}}{\text{J}}$
- Dezentralität (Redundanz, Robustheit) ... bis $10^{12} \frac{\text{Zellen}}{\text{Schleimpilz}}$
- massiv parallele Verarbeitung bis $10^9 \frac{\text{Moleküle}}{\text{Zelle}}$
- Selbstorganisations- und Reparaturfähigkeiten
- Biokompatibilität
- Rohstoffverfügbarkeit und Recyclingfähigkeit
- Persistenz unter geeigneten Bedingungen
- Strukturdynamik als inhärentes Beschreibungsmittel
- regelbasierte Programmierbarkeit

„In jedem Lebewesen steckt ein leistungsfähiger Computer.“

Biomolekulare Speichermedien in vivo bieten

- hohe Speicherdichte bis $1 \frac{\text{Bit}}{\text{nm}^3}$
- Energieeffizienz bis $10^{18} \frac{\text{ops}}{\text{J}}$
- Dezentralität (Redundanz, Robustheit) ... bis $10^{12} \frac{\text{Zellen}}{\text{Schleimpilz}}$
- massiv parallele Verarbeitung bis $10^9 \frac{\text{Moleküle}}{\text{Zelle}}$
- Selbstorganisations- und Reparaturfähigkeiten
- Biokompatibilität
- Rohstoffverfügbarkeit und Recyclingfähigkeit
- Persistenz unter geeigneten Bedingungen
- Strukturdynamik als inhärentes Beschreibungsmittel
- regelbasierte Programmierbarkeit

„In jedem Lebewesen steckt ein leistungsfähiger Computer.“

Biomolekulare Speichermedien in vivo bieten

- hohe Speicherdichte bis $1 \frac{\text{Bit}}{\text{nm}^3}$
- Energieeffizienz bis $10^{18} \frac{\text{ops}}{\text{J}}$
- Dezentralität (Redundanz, Robustheit) ... bis $10^{12} \frac{\text{Zellen}}{\text{Schleimpilz}}$
- massiv parallele Verarbeitung bis $10^9 \frac{\text{Moleküle}}{\text{Zelle}}$
- Selbstorganisations- und Reparaturfähigkeiten
- Biokompatibilität
- Rohstoffverfügbarkeit und Recyclingfähigkeit
- Persistenz unter geeigneten Bedingungen
- Strukturdynamik als inhärentes Beschreibungsmittel
- regelbasierte Programmierbarkeit

„In jedem Lebewesen steckt ein leistungsfähiger Computer.“

Biomolekulare Speichermedien in vivo bieten

- hohe Speicherdichte bis $1 \frac{\text{Bit}}{\text{nm}^3}$
- Energieeffizienz bis $10^{18} \frac{\text{ops}}{\text{J}}$
- Dezentralität (Redundanz, Robustheit) ... bis $10^{12} \frac{\text{Zellen}}{\text{Schleimpilz}}$
- massiv parallele Verarbeitung bis $10^9 \frac{\text{Moleküle}}{\text{Zelle}}$
- Selbstorganisations- und Reparaturfähigkeiten
- Biokompatibilität
- Rohstoffverfügbarkeit und Recyclingfähigkeit
- Persistenz unter geeigneten Bedingungen
- Strukturdynamik als inhärentes Beschreibungsmittel
- regelbasierte Programmierbarkeit

„In jedem Lebewesen steckt ein leistungsfähiger Computer.“

Biomolekulare Speichermedien in vivo bieten

- hohe Speicherdichte bis $1 \frac{\text{Bit}}{\text{nm}^3}$
- Energieeffizienz bis $10^{18} \frac{\text{ops}}{\text{J}}$
- Dezentralität (Redundanz, Robustheit) ... bis $10^{12} \frac{\text{Zellen}}{\text{Schleimpilz}}$
- massiv parallele Verarbeitung bis $10^9 \frac{\text{Moleküle}}{\text{Zelle}}$
- Selbstorganisations- und Reparaturfähigkeiten
- Biokompatibilität
- Rohstoffverfügbarkeit und Recyclingfähigkeit
- Persistenz unter geeigneten Bedingungen
- Strukturdynamik als inhärentes Beschreibungsmittel
- regelbasierte Programmierbarkeit

„In jedem Lebewesen steckt ein leistungsfähiger Computer.“

Ausprägungen im biomolekularen Computing

- **Chemisches Rechnen**
 - Reaktionssysteme als Analog- und Digitalcomputer
- **DNA- und RNA-Computing**
- **Protein-Computing und genetische Schaltkreise**
 - Zellsignal- und Genregulationsnetzwerke
- **Molecular Assembly** – Selbstorganisation nach Regeln
- **Membran- und zellbasiertes Computing**
 - dynamische Raumstrukturen
- **Neuronale Netze** – Aufbau, Training, Anwendungen
- **Evolutionäres Computing** – eine universelle Heuristik
- **Amorphes/Organisches Computing**
 - komplexe Teamlösungen

Ausprägungen im biomolekularen Computing

- **Chemisches Rechnen**
 - Reaktionssysteme als Analog- und Digitalcomputer
- **DNA- und RNA-Computing**
- **Protein-Computing und genetische Schaltkreise**
 - Zellsignal- und Genregulationsnetzwerke
- **Molecular Assembly** – Selbstorganisation nach Regeln
- **Membran- und zellbasiertes Computing**
 - dynamische Raumstrukturen
- **Neuronale Netze** – Aufbau, Training, Anwendungen
- **Evolutionäres Computing** – eine universelle Heuristik
- **Amorphes/Organisches Computing**
 - komplexe Teamlösungen

Ausprägungen im biomolekularen Computing

- **Chemisches Rechnen**
 - Reaktionssysteme als Analog- und Digitalcomputer
- **DNA- und RNA-Computing**
- **Protein-Computing und genetische Schaltkreise**
 - Zellsignal- und Genregulationsnetzwerke
- **Molecular Assembly** – Selbstorganisation nach Regeln
- **Membran- und zellbasiertes Computing**
 - dynamische Raumstrukturen
- **Neuronale Netze** – Aufbau, Training, Anwendungen
- **Evolutionäres Computing** – eine universelle Heuristik
- **Amorphes/Organisches Computing**
 - komplexe Teamlösungen

Ausprägungen im biomolekularen Computing

- **Chemisches Rechnen**
 - Reaktionssysteme als Analog- und Digitalcomputer
- **DNA- und RNA-Computing**
- **Protein-Computing und genetische Schaltkreise**
 - Zellsignal- und Genregulationsnetzwerke
- **Molecular Assembly** – Selbstorganisation nach Regeln
- **Membran- und zellbasiertes Computing**
 - dynamische Raumstrukturen
- **Neuronale Netze** – Aufbau, Training, Anwendungen
- **Evolutionäres Computing** – eine universelle Heuristik
- **Amorphes/Organisches Computing**
 - komplexe Teamlösungen

Ausprägungen im biomolekularen Computing

- **Chemisches Rechnen**
 - Reaktionssysteme als Analog- und Digitalcomputer
- **DNA- und RNA-Computing**
- **Protein-Computing und genetische Schaltkreise**
 - Zellsignal- und Genregulationsnetzwerke
- **Molecular Assembly** – Selbstorganisation nach Regeln
- **Membran- und zellbasiertes Computing**
 - dynamische Raumstrukturen
- **Neuronale Netze** – Aufbau, Training, Anwendungen
- **Evolutionäres Computing** – eine universelle Heuristik
- **Amorphes/Organisches Computing**
 - komplexe Teamlösungen

Ausprägungen im biomolekularen Computing

- **Chemisches Rechnen**
 - Reaktionssysteme als Analog- und Digitalcomputer
- **DNA- und RNA-Computing**
- **Protein-Computing und genetische Schaltkreise**
 - Zellsignal- und Genregulationsnetzwerke
- **Molecular Assembly** – Selbstorganisation nach Regeln
- **Membran- und zellbasiertes Computing**
 - dynamische Raumstrukturen
- **Neuronale Netze** – Aufbau, Training, Anwendungen
- **Evolutionäres Computing** – eine universelle Heuristik
- **Amorphes/Organisches Computing**
 - komplexe Teamlösungen

Ausprägungen im biomolekularen Computing

- **Chemisches Rechnen**
 - Reaktionssysteme als Analog- und Digitalcomputer
- **DNA- und RNA-Computing**
- **Protein-Computing und genetische Schaltkreise**
 - Zellsignal- und Genregulationsnetzwerke
- **Molecular Assembly** – Selbstorganisation nach Regeln
- **Membran- und zellbasiertes Computing**
 - dynamische Raumstrukturen
- **Neuronale Netze** – Aufbau, Training, Anwendungen
- **Evolutionäres Computing** – eine universelle Heuristik
- **Amorphes/Organisches Computing**
 - komplexe Teamlösungen

Ausprägungen im biomolekularen Computing

- **Chemisches Rechnen**
 - Reaktionssysteme als Analog- und Digitalcomputer
- **DNA- und RNA-Computing**
- **Protein-Computing und genetische Schaltkreise**
 - Zellsignal- und Genregulationsnetzwerke
- **Molecular Assembly** – Selbstorganisation nach Regeln
- **Membran- und zellbasiertes Computing**
 - dynamische Raumstrukturen
- **Neuronale Netze** – Aufbau, Training, Anwendungen
- **Evolutionäres Computing** – eine universelle Heuristik
- **Amorphes/Organisches Computing**
 - komplexe Teamlösungen

Reaktionskinetik – Grundbegriffe

System

abgegrenzter Reaktionsraum mit Volumen V

Stoffmenge n

Anzahl Moleküle, Atome oder Ionen gleicher Art
(gleichen Stoffes, gleicher Struktur, gleicher Spezies)
im System

Stoffkonzentration c

Quotient $c = \frac{n}{V}$

Reaktionsgeschwindigkeit v_R

Quotient $v_R = \frac{\Delta c}{\Delta t}$,

für infinitesimal kleine Zeitänderungen: $v_R = \frac{dc}{dt}$

Annahme:

Gleichverteilung der Teilchen im System (Homogenität)

Reaktionskinetik – Grundbegriffe

System

abgegrenzter Reaktionsraum mit Volumen V

Stoffmenge n

Anzahl Moleküle, Atome oder Ionen gleicher Art
(gleichen Stoffes, gleicher Struktur, gleicher Spezies)
im System

Stoffkonzentration c

Quotient $c = \frac{n}{V}$

Reaktionsgeschwindigkeit v_R

Quotient $v_R = \frac{\Delta c}{\Delta t}$,

für infinitesimal kleine Zeitänderungen: $v_R = \frac{dc}{dt}$

Annahme:

Gleichverteilung der Teilchen im System (Homogenität)

Reaktionskinetik – Grundbegriffe

System

abgegrenzter Reaktionsraum mit Volumen V

Stoffmenge n

Anzahl Moleküle, Atome oder Ionen gleicher Art
(gleichen Stoffes, gleicher Struktur, gleicher Spezies)
im System

Stoffkonzentration c

Quotient $c = \frac{n}{V}$

Reaktionsgeschwindigkeit v_R

Quotient $v_R = \frac{\Delta c}{\Delta t}$,

für infinitesimal kleine Zeitänderungen: $v_R = \frac{dc}{dt}$

Annahme:

Gleichverteilung der Teilchen im System (Homogenität)

Reaktionskinetik – Grundbegriffe

System

abgegrenzter Reaktionsraum mit Volumen V

Stoffmenge n

Anzahl Moleküle, Atome oder Ionen gleicher Art
(gleichen Stoffes, gleicher Struktur, gleicher Spezies)
im System

Stoffkonzentration c

Quotient $c = \frac{n}{V}$

Reaktionsgeschwindigkeit v_R

Quotient $v_R = \frac{\Delta c}{\Delta t}$,

für infinitesimal kleine Zeitänderungen: $v_R = \frac{dc}{dt}$

Annahme:

Gleichverteilung der Teilchen im System (Homogenität)

Reaktionskinetik – Grundbegriffe

System

abgegrenzter Reaktionsraum mit Volumen V

Stoffmenge n

Anzahl Moleküle, Atome oder Ionen gleicher Art
(gleichen Stoffes, gleicher Struktur, gleicher Spezies)
im System

Stoffkonzentration c

Quotient $c = \frac{n}{V}$

Reaktionsgeschwindigkeit v_R

Quotient $v_R = \frac{\Delta c}{\Delta t}$,

für infinitesimal kleine Zeitänderungen: $v_R = \frac{dc}{dt}$

Annahme:

Gleichverteilung der Teilchen im System (Homogenität)

Reaktionskinetik – Konzentrations-Zeit-Verläufe



Reaktionsverlauf im homogenen Medium (Guldberg 1867)

Nichtelastische Kollisionen Z zwischen A und B produzieren C:

$$Z \sim c_A \text{ und } Z \sim c_B$$

$$Z \sim c_A \cdot c_B$$

$$v_C = k \cdot c_A \cdot c_B \text{ mit Ratenkonstante } k$$

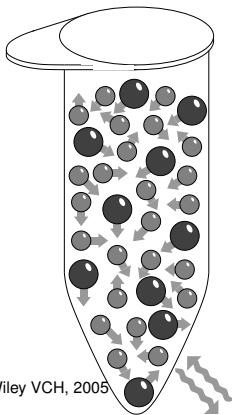
$$\frac{d c_C}{d t} = k \cdot c_A \cdot c_B \text{ Differentialgleichung}$$

Anfangsbedingung vorgegeben:

$$c_A(0), c_B(0), c_C(0)$$

Lösung der Differentialgleichung:

Stoffkonzentrations-Zeitverlauf $c_C(t)$
für Reaktionsprodukt C



E. Klipp, R. Herwig, A. Kowald, C. Wierling, H. Lehrach. Systems Biology in Practice. Wiley VCH, 2005

Reaktionskinetik – Konzentrations-Zeit-Verläufe



Reaktionsverlauf im homogenen Medium (Guldberg 1867)

Nichtelastische Kollisionen Z zwischen A und B produzieren C:

$$Z \sim c_A \text{ und } Z \sim c_B$$

$$Z \sim c_A \cdot c_B$$

$$v_C = k \cdot c_A \cdot c_B \text{ mit Ratenkonstante } k$$

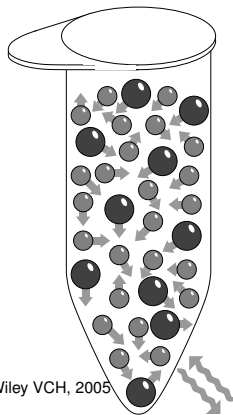
$$\frac{dc_C}{dt} = k \cdot c_A \cdot c_B \text{ Differentialgleichung}$$

Anfangsbedingung vorgegeben:

$$c_A(0), c_B(0), c_C(0)$$

Lösung der Differentialgleichung:

Stoffkonzentrations-Zeitverlauf $c_C(t)$
für Reaktionsprodukt C



E. Klipp, R. Herwig, A. Kowald, C. Wierling, H. Lehrach. Systems Biology in Practice. Wiley VCH, 2005

Reaktionskinetik – Konzentrations-Zeit-Verläufe



Reaktionsverlauf im homogenen Medium (Guldberg 1867)

Nichtelastische Kollisionen Z zwischen A und B produzieren C :

$$Z \sim c_A \text{ und } Z \sim c_B$$

$$Z \sim c_A \cdot c_B$$

$$v_C = k \cdot c_A \cdot c_B \text{ mit Ratenkonstante } k$$

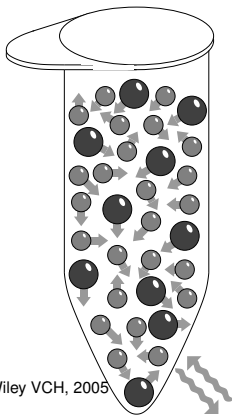
$$\frac{d c_C}{d t} = k \cdot c_A \cdot c_B \text{ Differentialgleichung}$$

Anfangsbedingung vorgegeben:

$$c_A(0), c_B(0), c_C(0)$$

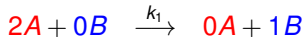
Lösung der Differentialgleichung:

Stoffkonzentrations-Zeitverlauf $c_C(t)$
für Reaktionsprodukt C



E. Klipp, R. Herwig, A. Kowald, C. Wierling, H. Lehrach. Systems Biology in Practice. Wiley VCH, 2005

Dynamik einer Einzelreaktion am Beispiel



Differentialgleichungssystem

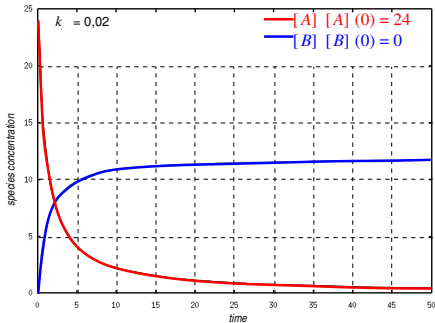
$$\frac{d c_A}{d t} = -2 \cdot k_1 \cdot c_A^2$$

$$\frac{d c_B}{d t} = k_1 \cdot c_A^2$$

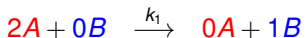
Analytische Lösung

$$c_A(t) = \left(2k_1 t + \frac{1}{c_A(0)} \right)^{-1} \quad \text{gdw. } c_A(0) > 0 \quad \text{sonst } c_A(t) = 0$$

$$c_B(t) = \left(-2 \left(2k_1 t + \frac{1}{c_A(0)} \right) \right)^{-1} + \frac{c_A(0)}{2} + c_B(0)$$



Dynamik einer Einzelreaktion am Beispiel



Differentialgleichungssystem

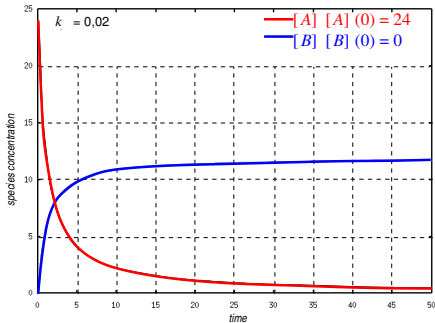
$$\frac{d c_A}{d t} = -2 \cdot k_1 \cdot c_A^2$$

$$\frac{d c_B}{d t} = k_1 \cdot c_A^2$$

Analytische Lösung

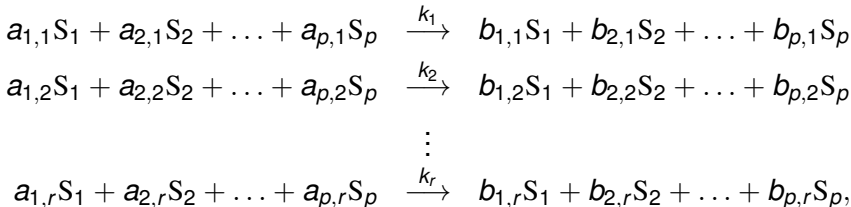
$$c_A(t) = \left(2k_1 t + \frac{1}{c_A(0)} \right)^{-1} \quad \text{gdw. } c_A(0) > 0 \quad \text{sonst } c_A(t) = 0$$

$$c_B(t) = \left(-2 \left(2k_1 t + \frac{1}{c_A(0)} \right) \right)^{-1} + \frac{c_A(0)}{2} + c_B(0)$$



Massenwirkungskinetik allgemein

Erweiterung auf r Reaktionsgleichungen durch Aufsummieren



Beschreibung des dyn. Verhaltens durch DGL-System:

$$\frac{dc_{S_i}}{dt} = \sum_{h=1}^r \left(k_h \cdot (b_{i,h} - a_{i,h}) \cdot \prod_{j=1}^p c_{S_j}^{a_{j,h}} \right) \quad \text{mit } i = 1, \dots, p$$

Künstliche Evolution von Reaktionsnetzwerken: Idee

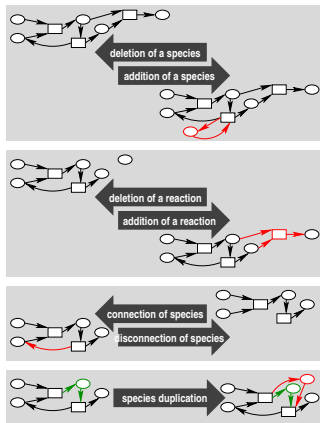
Operatoren für Strukturevolution (Topologie)

- Hinzufügen/Löschen einer Spezies
- Hinzufügen/Löschen einer Reaktion
- Vorhandene Spezies in Reaktion einbinden/daraus entfernen
- Duplizieren einer Spezies

Operatoren für Parameteranpassung

- Mutation zufällig ausgewählter Ratenkonstanten durch Addition einer Gauss-Variablen

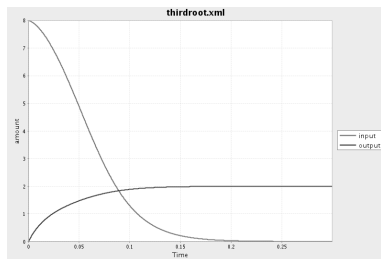
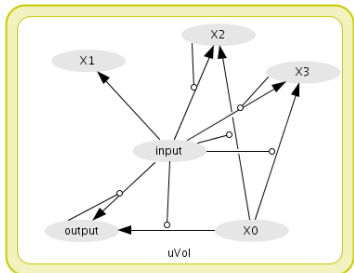
maximale Netzwerkgröße vorgebar



T. Lenser, T. Hinze, B. Ibrahim, P. Dittrich. Towards Evolutionary Network Reconstruction Tools for Systems Biology. LNCS 4447:132-142, Springer, 2007

Beispiel: Evolution eines Kubikwurzel-Netzwerks

Bestes Ergebnisnetzwerk berechnet Kubikwurzel exakt

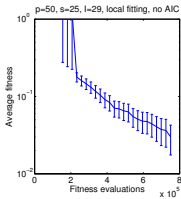


T. Lenser, N. Matsumaru, T. Hinze, P. Dittrich. Tracking the Evolution of Chemical Computing Networks. In S. Bullock, J. Noble, R. Watson, M.A. Bedau (Eds.), Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems (ALife XI), ISBN 978-0-262-75017-2, pp. 343-350, MIT Press, 2008

Evolutionärer Algorithmus (EA) für $\sqrt[3]{\square}$ -Netzwerk

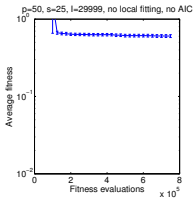
Fitnessverläufe unter verschiedenen Experimentierbedingungen:

zweistufiger EA



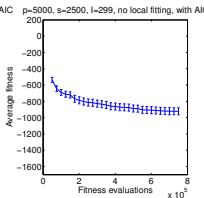
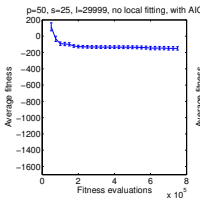
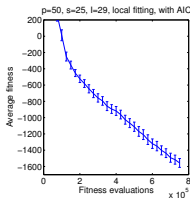
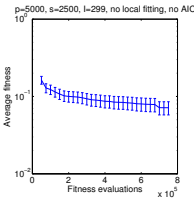
einstufiger EA

tausendfache Laufzeit



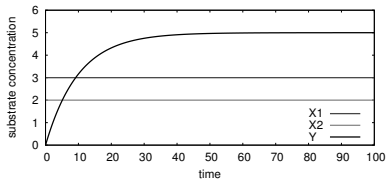
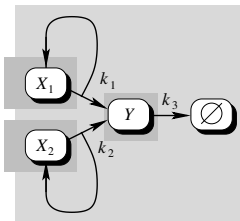
einstufiger EA

hundertfach größere Population

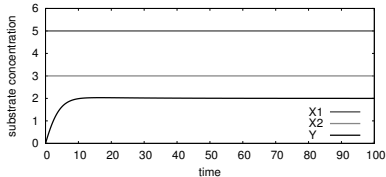
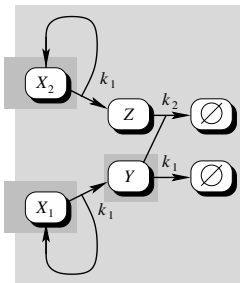


T. Lenser, N. Matsumaru, T. Hinze, P. Dittrich. Tracking the Evolution of Chemical Computing Networks. In S. Bullock, J. Noble, R. Watson, M.A. Bedau (Eds.), Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems (ALife XI), ISBN 978-0-262-75017-2, pp. 343-350, MIT Press, 2008

Addition und nichtnegative Subtraktion



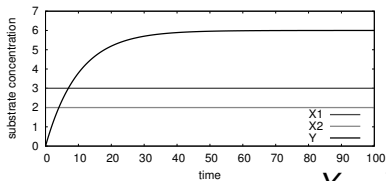
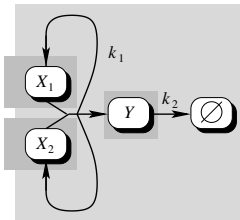
$$Y = X_1 + X_2$$



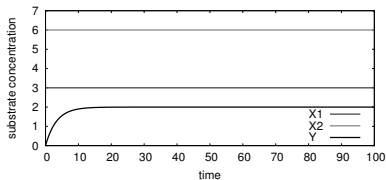
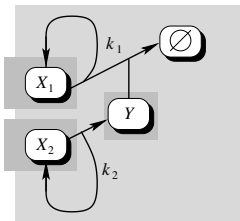
$$Y = X_1 - X_2$$

T. Hinze, C. Bodenstern, B. Schau, I. Heiland, S. Schuster. Chemical Analog Computers for Clock Frequency Control Based on P Modules. Lecture Notes in Computer Science **7184**:182-202, 2012

Multiplikation und Division



$$Y = X_1 \cdot X_2$$

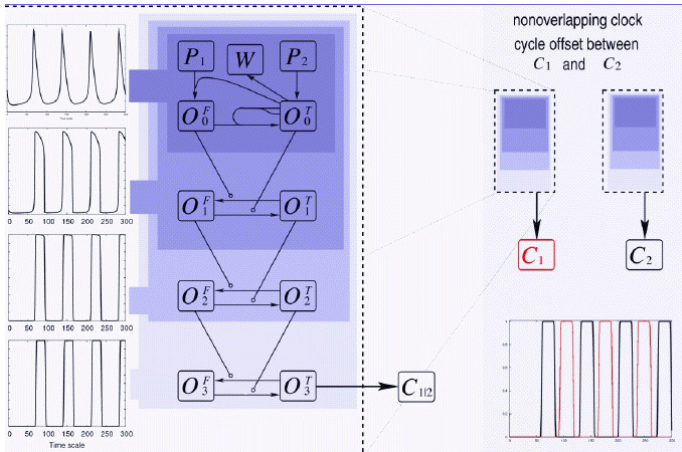


$$Y = X_2 / X_1$$

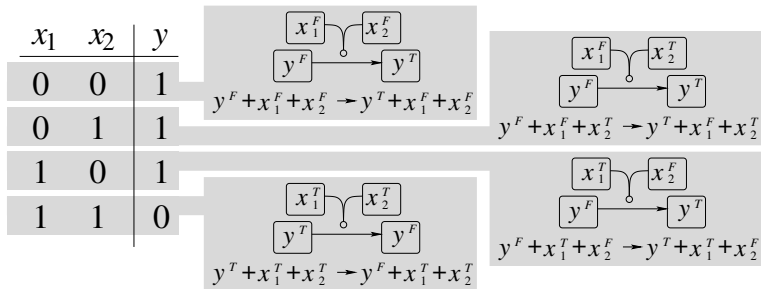
T. Hinze, C. Bodenstern, B. Schau, I. Heiland, S. Schuster. Chemical Analog Computers for Clock Frequency Control Based on P Modules. Lecture Notes in Computer Science **7184**:182-202, 2012

Ein chemischer Taktgeber

- basierend auf Belousov-Zhabotinsky-Reaktion (Brusselator)
- Kaskade von Hilfsreaktionen für schnelles Umschaltverhalten
- zwei versetzte Oszillationen liefern Taktsignale C_1 und C_2



Chemisches NAND-Gatter (mit Katalysatoren)



- Schalterenzyme nach dem Vorbild von Zellsignalnetzwerken
- zulässige Initialisierungen gemäß Belegungstabelle
- schnelles Schaltverhalten über hohe Konstanten k
- Eingabe x_1, x_2 bleibt erhalten (nicht chemisch abgebaut)
- alle binären Funktionen nach diesem Prinzip realisierbar

Registermaschine (RAM)

Ein befehlsorientiertes Berechnungsmodell, Turing-vollständig

- Definition der Komponenten

$$RAM = (R, L, P, l_1)$$



- verfügbare Befehle

- l_j : INC R_k l_j Register R_k inkrementieren, gehe zu l_j
- l_j : DEC R_k l_j Register R_k dekrementieren, gehe zu l_j
- l_j : IFZ R_k l_j l_p Falls $R_k = 0$ gehe zu l_j sonst zu l_p
- l_j : HALT Programmende

- Festlegungen

- jedes Register speichert eine natürliche Zahl
- bei Start alle Register mit Eingabewerten initialisiert
- dediziertes Ausgaberegister festgelegt
- deterministische Arbeitsweise

Registermaschine (RAM)

Ein befehlsorientiertes Berechnungsmodell, Turing-vollständig

- Definition der Komponenten

$$RAM = (R, L, P, l_1)$$



- verfügbare Befehle

- l_j : INC R_k l_j Register R_k inkrementieren, gehe zu l_j
- l_j : DEC R_k l_j Register R_k dekrementieren, gehe zu l_j
- l_j : IFZ R_k l_j l_p Falls $R_k = 0$ gehe zu l_j sonst zu l_p
- l_j : HALT Programmende

- Festlegungen

- jedes Register speichert eine natürliche Zahl
- bei Start alle Register mit Eingabewerten initialisiert
- dediziertes Ausgaberegister festgelegt
- deterministische Arbeitsweise

Registermaschine (RAM)

Ein befehlsorientiertes Berechnungsmodell, Turing-vollständig

- Definition der Komponenten

$$RAM = (R, L, P, l_1)$$



- verfügbare Befehle

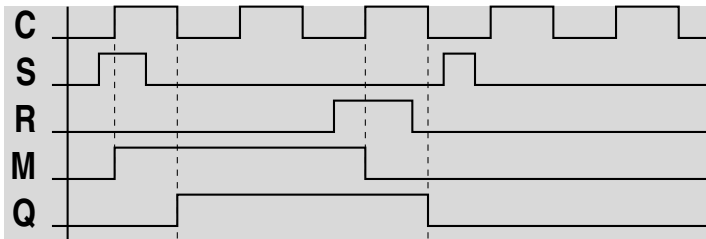
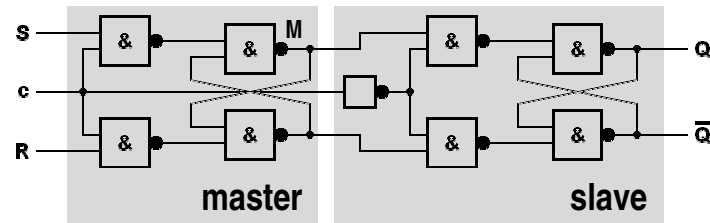
- l_j : INC R_k l_j Register R_k inkrementieren, gehe zu l_j
- l_j : DEC R_k l_j Register R_k dekrementieren, gehe zu l_j
- l_j : IFZ R_k l_j l_p Falls $R_k = 0$ gehe zu l_j sonst zu l_p
- l_j : HALT Programmende

- Festlegungen

- jedes Register speichert eine natürliche Zahl
- bei Start alle Register mit Eingabewerten initialisiert
- dediziertes Ausgaberegister festgelegt
- deterministische Arbeitsweise

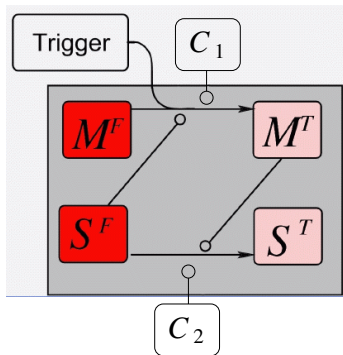
Master-Slave Flip-Flop (MSFF)

zuverlässiger 1-bit-Speicher, gut untersucht



Chemische MSFF Implementierung

zweistufiges Schalten von **FALSE** auf **TRUE** mittels
Trigger-Spezien und versetzten Taktsignalen C_1 und C_2

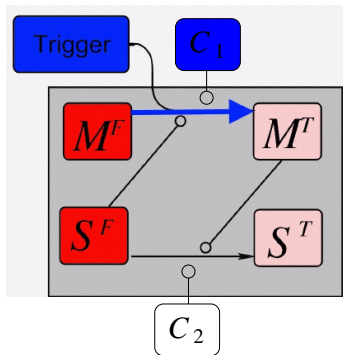


Spezies M^F , M^T : Bitwert im Master-Teil

Spezies S^F , S^T : Bitwert im Slave-Teil

Chemische MSFF Implementierung

zweistufiges Schalten von **FALSE** auf **TRUE** mittels
Trigger-Spezien und versetzten Taktsignalen C_1 und C_2

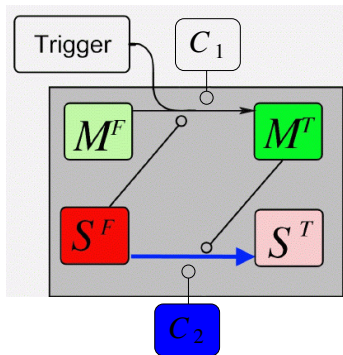


Spezies M^F , M^T : Bitwert im Master-Teil

Spezies S^F , S^T : Bitwert im Slave-Teil

Chemische MSFF Implementierung

zweistufiges Schalten von **FALSE** auf **TRUE** mittels
Trigger-Spezien und versetzten Taktsignalen C_1 und C_2

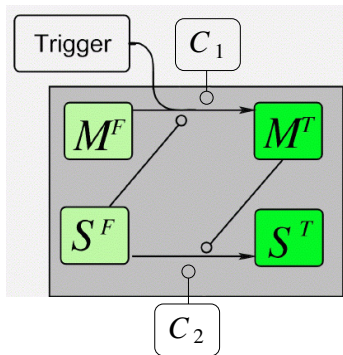


Spezies M^F , M^T : Bitwert im Master-Teil

Spezies S^F , S^T : Bitwert im Slave-Teil

Chemische MSFF Implementierung

zweistufiges Schalten von **FALSE** auf **TRUE** mittels
Trigger-Spezien und versetzten Taktsignalen C_1 und C_2

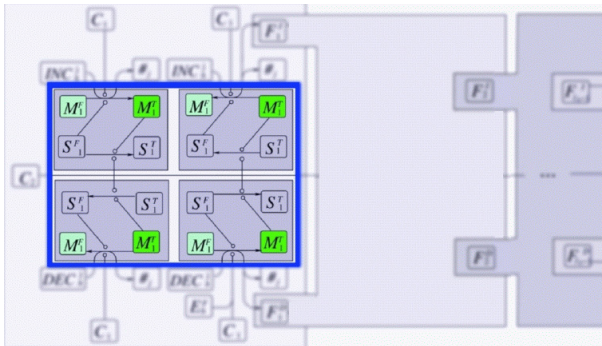


Spezies M^F , M^T : Bitwert im Master-Teil

Spezies S^F , S^T : Bitwert im Slave-Teil

Vom MSFF zum Register

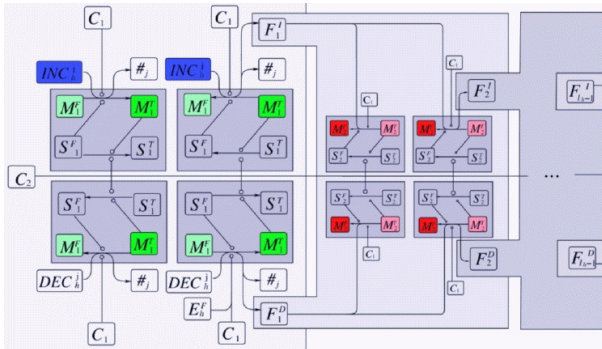
- vier Netzwerkmodule (alle Schaltszenarien) bilden MSFF
- Verkettung von MSFFs zu Registern beliebiger Bitlänge
- Annahme für Universalität: MSFF als selbstreplizierbare modulare Einheit



T. Hinze, R. Fassler, T. Lenser, P. Dittrich. Register Machine Computations on Binary Numbers by Oscillating and Catalytic Chemical Reactions Modelled using Mass-Action Kinetics. International Journal of Foundations of Computer Science 20(3):411-426, 2009

Vom MSFF zum Register

- vier Netzwerkmodule (alle Schaltszenarien) bilden MSFF
- Verkettung von MSFFs zu Registern beliebiger Bitlänge
- Annahme für Universalität: MSFF als selbstreplizierbare modulare Einheit



T. Hinze, R. Fassler, T. Lenser, P. Dittrich. Register Machine Computations on Binary Numbers by Oscillating and Catalytic Chemical Reactions Modelled using Mass-Action Kinetics.
International Journal of Foundations of Computer Science 20(3):411-426, 2009

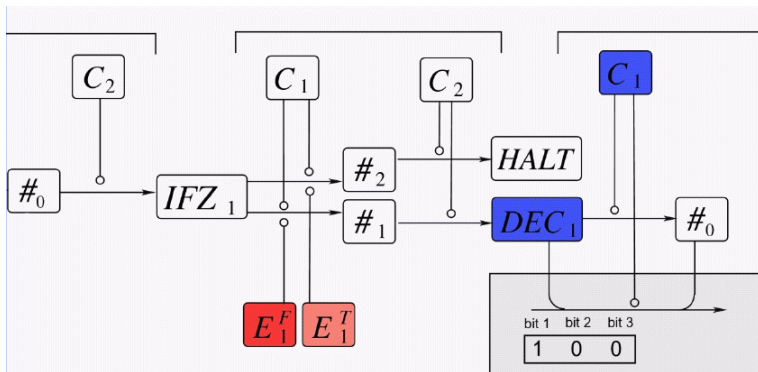
Chemische Programmsteuerung

einfaches Beispiel für sequentiellen Befehlsfluss:

#₀ : IFZ R₁ #₂ #₁

#₁ : DEC R₁ #₀

#₂ : HALT



T. Hinze, R. Fessler, T. Lenser, P. Dittrich. Register Machine Computations on Binary Numbers by Oscillating and Catalytic Chemical Reactions Modelled using Mass-Action Kinetics.

International Journal of Foundations of Computer Science 20(3):411-426, 2009

Chemische Programmsteuerung

Transformationschema

Befehl	Chemische Reaktionen
$\#_i : \text{INC } R_h \#_j$	$\#_i + C_2 \xrightarrow{k_p} \text{INC}_h^j + C_2$ $\text{INC}_h^j + C_1 \xrightarrow{k_b} \#_j + C_1$
$\#_i : \text{DEC } R_h \#_j$	$\#_i + C_2 \xrightarrow{k_p} \text{DEC}_h^j + C_2$ $\text{DEC}_h^j + C_1 \xrightarrow{k_b} \#_j + C_1$
$\#_i : \text{IFZ } R_h \#_j \#_q$	$\#_i + C_2 \xrightarrow{k_p} \text{IFZ}_h^{j,q} + C_2$ $\text{IFZ}_h^{j,q} + E_h^T + C_1 \xrightarrow{k_s} \#_j + E_h^T + C_1$ $\text{IFZ}_h^{j,q} + E_h^F + C_1 \xrightarrow{k_s} \#_q + E_h^F + C_1$
$\#_i : \text{HALT}$	$\#_i + C_2 \xrightarrow{k_p} \text{HALT} + C_2$

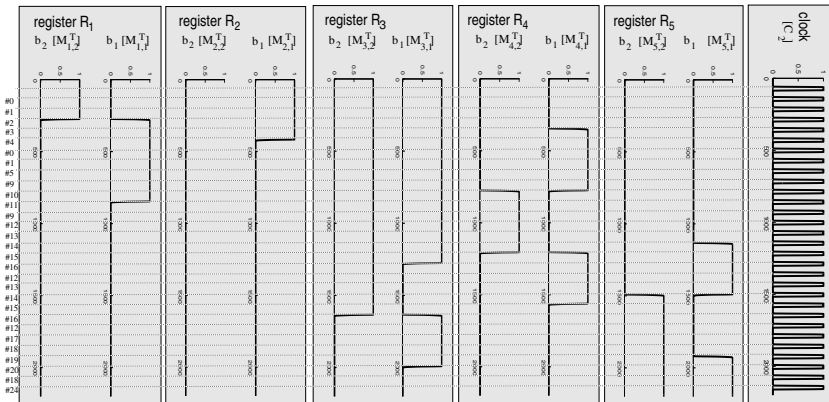
C_1, C_2 : Spezien, die Taktsignale bereitstellen

T. Hinze, R. Fassler, T. Lenser, P. Dittrich. Register Machine Computations on Binary Numbers by Oscillating and Catalytic Chemical Reactions Modelled using Mass-Action Kinetics.

International Journal of Foundations of Computer Science 20(3):411-426, 2009

Beispiel: Maximum von drei Zahlen „max(2, 1, 3)“

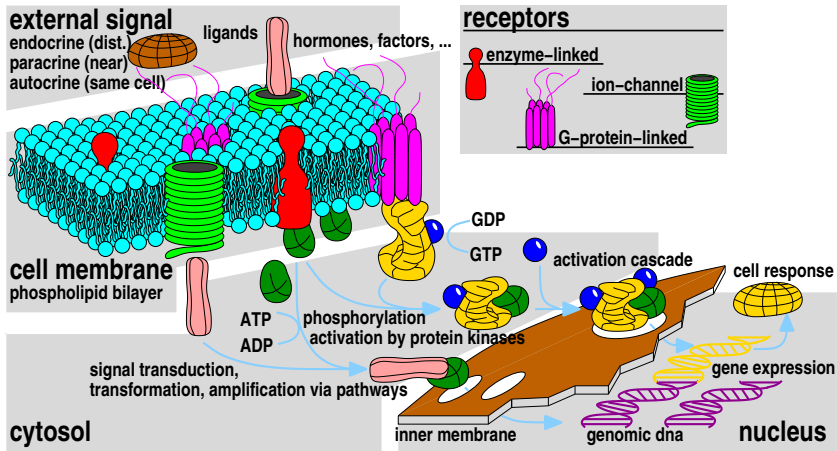
- $R_5 := \max(R_1, R_2, R_3)$
- Idee: $R_4 := \max(R_1, R_2)$; $R_5 := \max(R_4, R_3)$
- gesamtes Netzwerk: 142 Spezies und 223 Reaktionen



T. Hinze, R. Fassler, T. Lenser, N. Matsumaru, P. Dittrich. Event-Driven Metamorphoses of P Systems.
Lecture Notes in Computer Science **5391**:231-245, 2009

1. Chemisches Rechnen und Netzwerkevolution
2. **Genetische Schaltkreise**
 - Zellsignalverarbeitung
 - RS-Flip-Flop durch Genexpression
 - Implementierung in vivo
3. Molecular Assembly

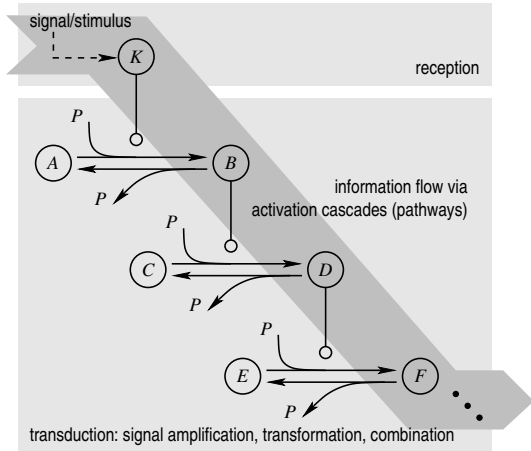
Signalverarbeitung in eukaryotischen Zellen



T. Hinze, T. Lenser, P. Dittrich. A Protein Substructure Based P System for Description and Analysis of Cell Signalling Networks. LNCS 4361:409-423, Springer, 2006

Informationsfluss in Zellsignalnetzwerken

- Vorhandensein/
Nichtvorhandensein
komplex konstruierter
Moleküle kodiert
Information
- i.A. niedrige
Molekülanzahl
im System
- Enzyme als Trigger
- Aktivierungskaskaden
durch schrittweise
Phosphorylierung

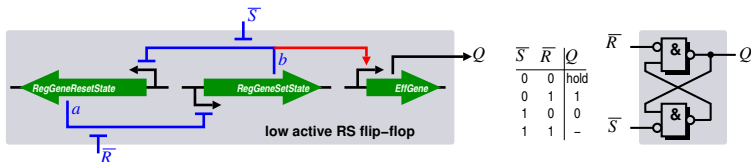


T. Hinze, J. Decraene, G.G. Mitchell, J. Behre, S. Schuster. Towards a Unified Approach for the Modelling, Analysis, and Simulation of Cell Signalling Networks. In Z. Zhao (Ed.), Sequence and Genome Analysis: Methods and Applications. ISBN 978-0-9807330-4-4, Chapter 13, pp. 227-257, iConcept Press, 2010

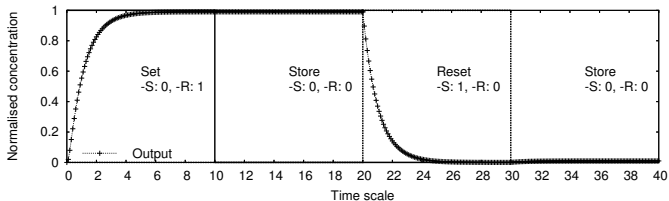
RS-Flip-Flop mittels Genexpression

Eingabe: Konzentration der Transkriptionsfaktoren \bar{S}, \bar{R}

Ausgabe: Konzentration des Genprodukts Q

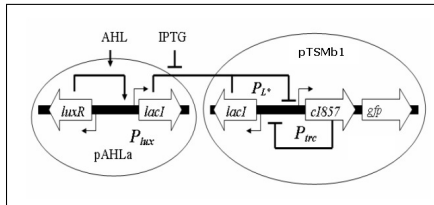


Dynamisches Verhalten (Simulation auf Basis der Reaktionskinetik)



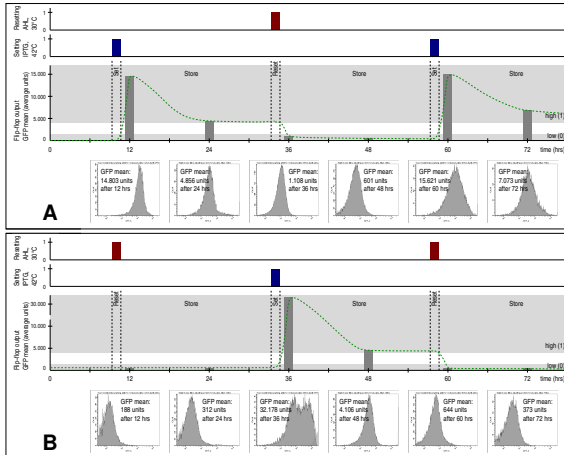
Einbettung RS-Flip-Flop in *Vibrio fischeri*

- Kodierung aller benötigten Gene auf konstruiertem Plasmid
- Eingabesignale: **AHL** und **IPTG**,
Ausgabesignal: green fluorescent protein (*gfp*)
- Messung der *gfp*-Konzentration mittels Durchflusszytometrie



T. Hinze, S. Hayat, T. Lenser, N. Matsumaru, P. Dittrich. Biosignal-Based Computing by AHL Induced Synthetic Gene Regulatory Networks. IEEE Engineering in Medicine and Biology 1:162-169, INSTICC, 2008

Experimentelle Ergebnisse in vivo



Wiederholtes Setzen und Rücksetzen des RS-Flip-Flops. Zellen wurden inkubiert mit den Eingabesignalen für jeweils 12 Stunden, gefolgt von 12 weiteren Stunden ohne Eingabesignale (Store). Abbildung zeigt dreimaliges Umschalten des RS-Flip-Flops über 72 Stunden aus zwei verschiedenen Initialkonfigurationen heraus (A und B)

S. Hayat, T. Hinze. Toward integration of in vivo molecular computing devices: successes and challenges. HFSP Journal 2(5):239-243, 2008

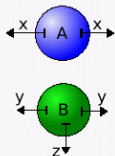
1. Chemisches Rechnen und Netzwerkevolution
2. Genetische Schaltkreise

3. **Molecular Assembly**

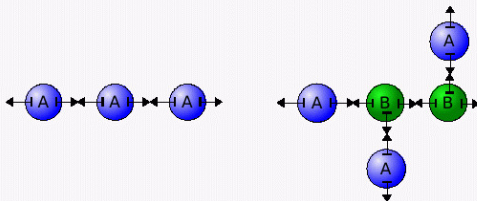
- Strukturdynamik von Molekülen mit SRSim
- Regelbasierte Erzeugung makromolekularer Strukturen
- Universalität

SRSim: Simulator für mol. Programmieren

a) monomers with their domains



b) examples for complex molecule graphs



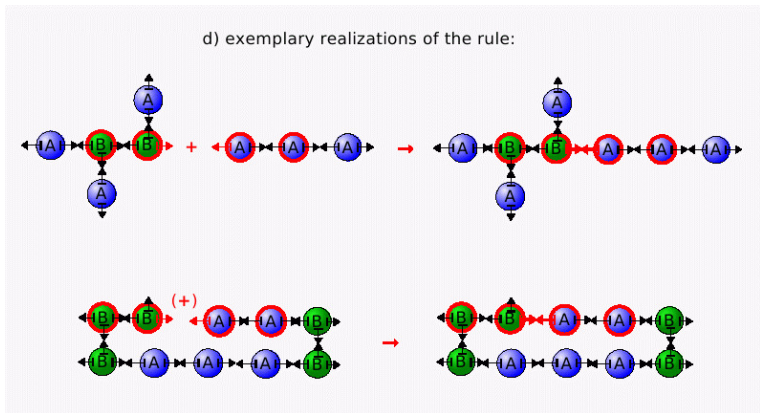
c) exemplary reaction rule:



d) exemplary realizations of the rule:

G. Grünert, B. Ibrahim, T. Lenser, T. Hinze, P. Dittrich. Rule-based Spatial Modeling with Diffusing, Geometrically Constrained Molecules. BMC Bioinformatics 11:307, 2010

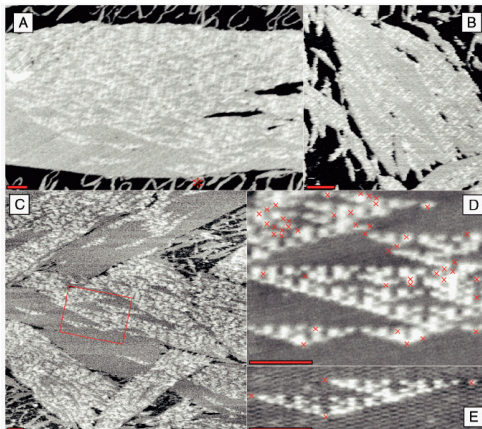
SRSim: Simulator für mol. Programmieren



G. Grünert, B. Ibrahim, T. Lenser, T. Hinze, P. Dittrich. Rule-based Spatial Modeling with Diffusing, Geometrically Constrained Molecules. BMC Bioinformatics 11:307, 2010

Sierpinski XOR-Dreiecke: Verifikation in vitro

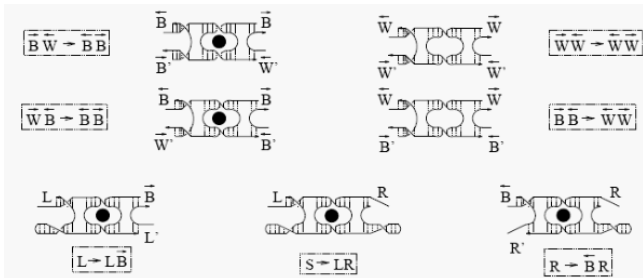
elektronenmikroskopische Aufnahmen, **x**: Anordnungsfehler



P.W.K. Rothmund, N. Papadakis, E. Winfree.
Algorithmic Self-Assembly of DNA Sierpinski Triangles.
PLoS Biology 2(12):2041-2053, 2011

Regelbasiertes mol. Programmieren: Universalität (I)

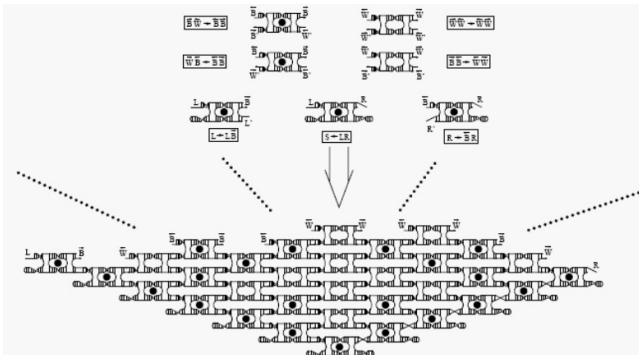
- Emulation von Chomsky-Grammatiken (Typ0), mithin Turing-Universalität
- Chomsky-Grammatik hierzu in Kuroda-Normalform überführen ($AB \rightarrow CD, A \rightarrow BC, A \rightarrow a, A \rightarrow \varepsilon$)
- viele Startsymbol-Andockstellen vorgeben, Sprache „wachsen lassen“



P.W.K. Rothmund, E. Winfree. The Program-Size Complexity of Self-Assembled Squares. ACM Symposium on Theory of Computing, 2010

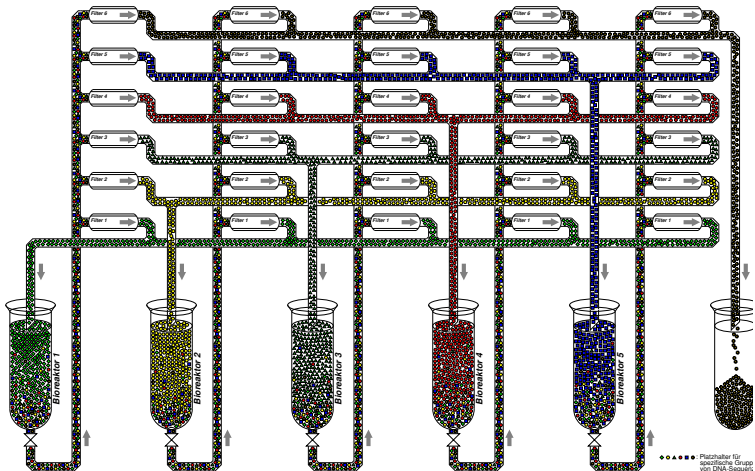
Regelbasiertes mol. Programmieren: Universalität (II)

Aus jeder vorgegebenen Startsymbol-Andockstelle kann ein Wort der Sprache wachsen,
Auslesen z.B. durch Fluoreszenzstoffe, die in all jene Tiles eingelagert werden, die Terminalsymbole hervorbringen



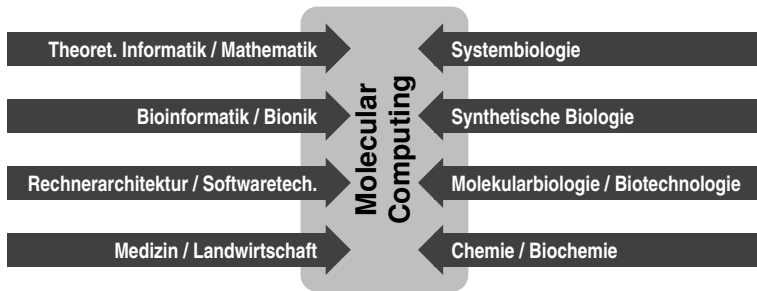
P.W.K. Rothmund, E. Winfree. The Program-Size Complexity of Self-Assembled Squares. ACM Symposium on Theory of Computing, 2010

Mikroflusssysteme nach dem Vorbild kompartimentierter Zellen



T. Hinze, M. Sturm. Rechnen mit DNA. Oldenbourg-Wissenschaftsverlag, 2004

Wohin entwickelt sich das Fachgebiet?



- Nanomaschinen und personalisierte Medizin
- Smart Materials und intelligente Werkstoffe
- Beiträge zum Cloud Computing und zur Künstlichen Intelligenz
- ...

Dankeschön an unser Team in Jena



Jena Centre for
Bioinformatics



Research Initiative in
Systems Biology



European Molecular
Computing Consortium

Fachbereich Bioinformatik an der Biologisch-Pharmazeutischen Fakultät
Friedrich-Schiller-Universität Jena