
Autonomic Computer Systems (CS321)

**Chemical Networking I:
A Formal Method for the Design and Analysis of Stochastic Protocols**

Thomas Meyer

Computer Science Department, University of Basel, Switzerland

November 24th, 2011

CS321 - Autonomic Computer Systems - 2011-11-24

2

Contents (two lectures)

Stochastic protocols

from Ethernet to reactive systems: an overview

Chemical networking

a formal approach to describe stochastic reactive systems

The three levels of structuring (distributed) computations

- **symbolic**

we know it all: $1 + 2 \Rightarrow 3$; HTTP GET + File \Rightarrow 200 OK

- **dynamic**

more challenging: global behavior emerging from the *dynamic* behavior of network nodes

- **organizational (next lecture)**

use the long-term trend of dynamic behavior to carry out computation

From robust to self-healing protocols (next lecture)

combining all the three levels to achieve robust protocol behavior and protocol code that heals itself

Contents (two lectures)

Stochastic protocols

from Ethernet to reactive systems: an overview

Chemical networking

a formal approach to describe stochastic reactive systems

The three levels of structuring (distributed) computations

- **symbolic**

we know it all: $1 + 2 \Rightarrow 3$; HTTP GET + File \Rightarrow 200 OK

- **dynamic**

more challenging: global behavior emerging from the *dynamic* behavior of network nodes

- **organizational**

use the long-term trend of dynamic behavior to carry out computation

From robust to self-healing protocols

combining all the three levels to achieve robust protocol behavior and protocol code that heals itself

Stochastic Protocols

from Ethernet to reactive systems: an overview

- What are stochastic protocols?

- Examples:

- Medium Access Control (MAC) protocols (e.g. Ethernet)

- Gossip protocols

- Rule-based protocols (Fraglets)

- What is the benefit of relaxing determinism?

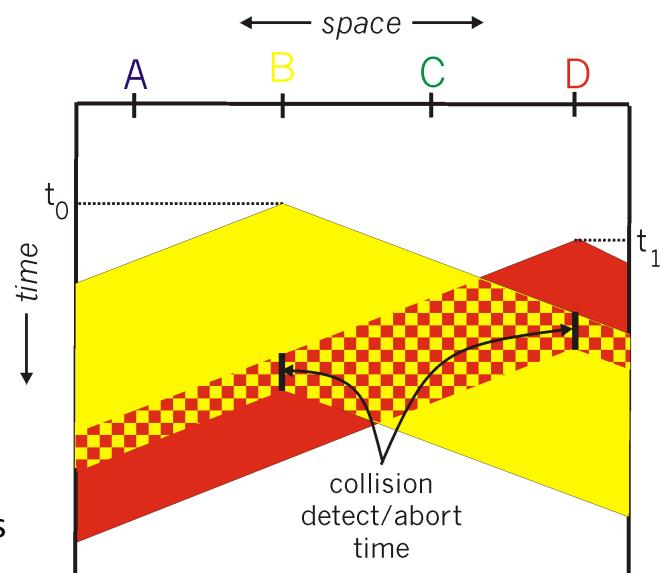
Stochastic Protocols

- Most of the algorithms discussed so far are deterministic:
same input \rightarrow same output
- Exception: Gossip-style protocols
 - select random peers, exchange state information with them
 - despite the randomness: the network reaches consensus and computes a distributed result
- Benefit of relaxing determinism:
 - increased robustness: if it works with random peers, it also works when the link to one peer is disconnected
 - symmetry breaking: avoiding biased result in regular networks
 - resolving access conflicts: e.g. shared medium access protocols

Medium Access Control

e.g. Ethernet

- CSMA (Carrier Sense Multiple Access):
listen before transmitting
- CD (Collision Detection):
listen while transmitting, abort transmission when collision detected
- Ethernet: is a CSMA/CD variant:
after aborting, station enters **exponential backoff**:
 - after m -th collision, station chooses K at random from $\{0, 1, 2, \dots, 2m-1\}$.
 - station waits $512 \cdot K$ bit times,
and starts sensing again
- Exponential random backoff solves problem that both colliding stations start to send at the same later time: **resolves access conflicts**

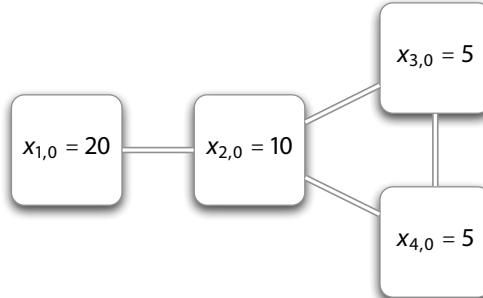


Gossip Protocols

e.g. Push-Sum: distributed computation of an aggregated value

Problem Statement:

- Each node contains an initial value (e.g. measurement of temperature)



- Goal: Compute the **average value** in a distributed way
- At the end, each node should know this **aggregated information**
- How could this be achieved with the gossip approach?

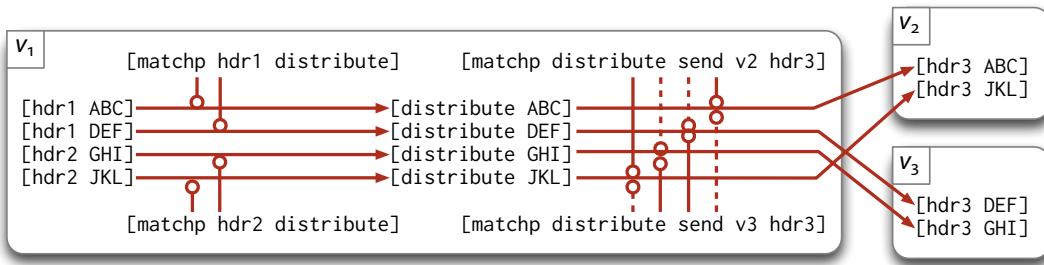
Gossip Protocols

Push-Sum (Kempe 2003)

- Each node v_i maintains a **sum** x_i initialized to the measured value, and a **weight** w_i initialized to $w_{i,0} = 1$.
- In each discrete time step t , each node follows the **algorithm**:
 1. Let $\{(\hat{x}_r, \hat{w}_r)\}$ be all pairs sent to v_i in round $t-1$
 2. Let $x_{t,i} = \sum_r \hat{x}_r$ and $w_{t,i} = \sum_r \hat{w}_r$
 3. Choose a target $v_j \in \text{neighbors}(v_i)$ at random
 4. Send the pair $(\frac{1}{2}x_{i,t}, \frac{1}{2}w_{i,t})$ to v_j and to v_i (yourself)
 5. $\frac{x_{i,t}}{w_{i,t}}$ is the estimate of the average in step 5

Rule-Based Protocols: e.g. Fraglets

- Packet processing can often be described by rules; without the need of a sequential program
- Rules operate on packets based on the packets' features (e.g. header fields)
- Multiple rules may be ready to operate on the same packet



- The rule execution algorithm randomly chooses a rule that matches a given packet.
- The outcome (packet content in nodes v_2 and v_3) is random, but proportional to the number of matching rule; good for load-balancing, for example

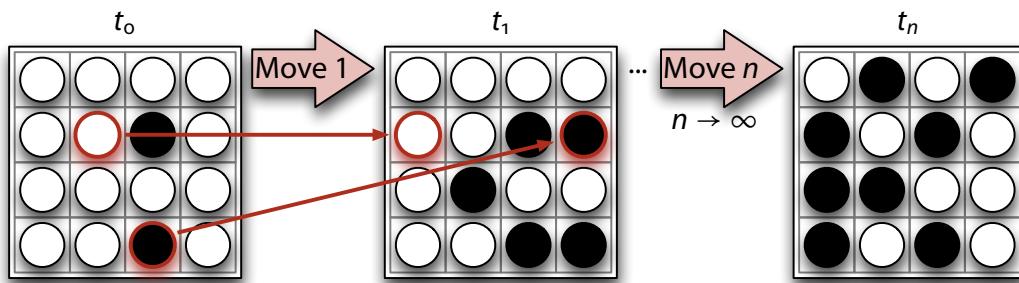
Why relaxing determinism?

- Losing determinism seems evil:
 - Why forcing random outcome if the computer is very good in exact computation?
 - Can we 'trust' stochastic results?
- Note: stochastic \neq arbitrarily random
 - results are distributed according to a well-known pattern (e.g. uniform, exponential, normal)
- We have to **distinguish two levels of description:**
 - **microscopic** level: **stochastic** algorithm
 - **macroscopic** level: **deterministic** behavior

The Mate-replicate-spread Game

A simple board game to illustrate the deterministic macroscopic outcome of stochastic microscopic interaction

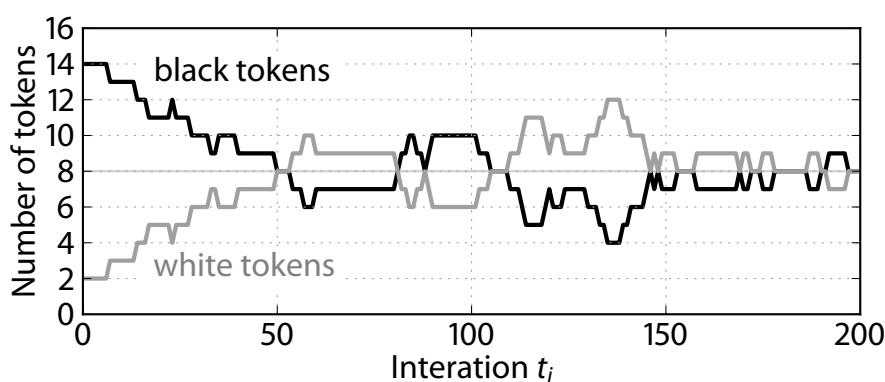
1. **Randomly** (uniform distribution) pick two squares (e.g. red circles)
2. **Mate:** If the colors of the tokens there is different: continue; otherwise: back to 1
3. **Replicate:** 'Copy' the two tokens (i.e., take a white and black token from outside)
4. **Spread: Randomly** (uniform distribution) pick two other squares and replace the tokens there with the copied tokens



other board games : Eigen, Winkler: *Das Spiel: Naturgesetze steuern den Zufall*, Piper, 1996

The Mate-replicate-spread Game

The population of black and white tokens strives towards the same size



- **Microscopic level:** individual tokens
 - random selection and replacement of the individuals
 - cannot predict which individual will survive
- **Macroscopic level:** quantity of tokens of a color
 - deterministic trend of this observed macroscopic measure

Problem of Stochastic Protocols

- Huge variety of stochastic protocols, different approaches
 - Stochastic protocols are often difficult to analyze:
 - We can model the stochastic process of a single Ethernet stations,
 - but how about a network of N stations?
Has been done, but much later than the protocol was in use.
 - In general:
The dynamic behavior of protocols is often neglected or analyzed after its deployment
 - Our research agenda:
Come up with a framework to engineer stochastic protocols, having their dynamic behavior under control from the beginning
- ==> Chemical Networking Protocols**

Contents (two lectures)

Stochastic protocols

from Ethernet to reactive systems: an overview

Chemical networking

a formal approach to describe stochastic reactive systems

The three levels of structuring (distributed) computations

- **symbolic**
we know it all: $1 + 2 \Rightarrow 3$; HTTP GET + File \Rightarrow 200 OK

- **dynamic**
more challenging: global behavior emerging from the *dynamic* behavior of network nodes

- **organizational**
use the long-term trend of dynamic behavior to carry out computation

From robust to self-healing protocols

combining all the three levels to achieve robust protocol behavior and protocol code that heals itself

Contents (two lectures)

Stochastic protocols

from Ethernet to reactive systems: an overview

Chemical networking

a formal approach to describe stochastic reactive systems

The three levels of structuring (distributed) computations

- **symbolic**

we know it all: $1 + 2 \Rightarrow 3$; HTTP GET + File \Rightarrow 200 OK

- **dynamic**

more challenging: global behavior emerging from the *dynamic* behavior of network nodes

- **organizational**

use the long-term trend of dynamic behavior to carry out computation

From robust to self-healing protocols

combining all the three levels to achieve robust protocol behavior and protocol code that heals itself

Chemical Networking

a formal approach to describe stochastic reactive systems

- **Reactive systems**

from rules to reactions

- **Reactive system examples**

examples in the field of computation, politics

- **Artificial Chemistry (AC)**

a formal description of reactive systems

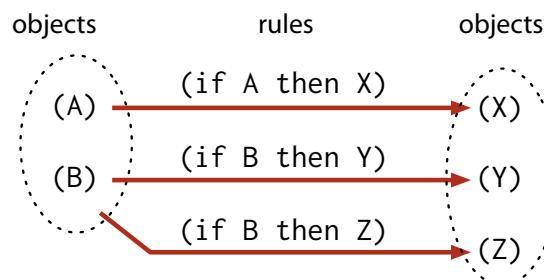
- **Chemical Networking Protocols**

Artificial Chemistry in the distributed context of a computer network

From Rules to Reactions

Rule-based systems can be described by...

- a set of (data) **objects** (e.g. packets)
- a set of (active) **rules** operating on objects; each rule consists of
 - a condition: defines when a rule matches an object, and
 - an action: defines what to do with the object.
- an **algorithm** that decides in which order the rules are applied to the objects

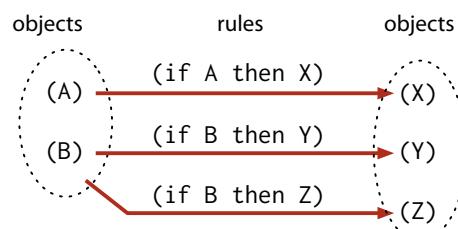
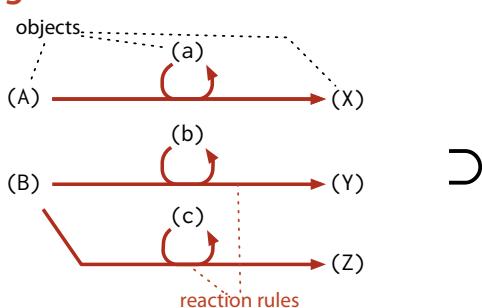


→ rules are persistent, data objects are being transformed by the rules

From Rules to Reactions

Reaction systems can be described by...

- a set of **objects**
- a set of **reaction rules** showing
 - which objects (types) are consumed,
 - and which objects (types) are produced by the reaction
- an **algorithm** that decides in which order the reaction rules are applied

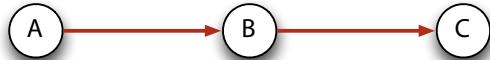


→ rule-based systems can be described as reactive systems
(rules: special persistent objects)

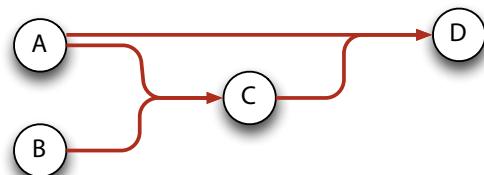
From Rules to Reactions

Reaction systems: no distinction between active and passive objects

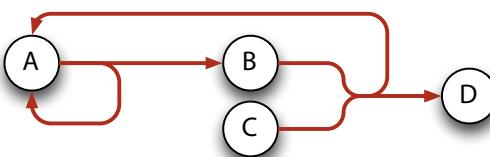
- an object may **transform spontaneously** into another object without interaction



- an object can **react with other objects** and produce one or more objects
(note: A and B are consumed when producing C)



- objects can **regenerate themselves** (spontaneously or by reaction with others)



→ reactive systems can be used as a model of many real-world systems...

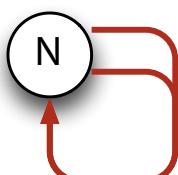
Reactive System Examples

Unconventional Computation

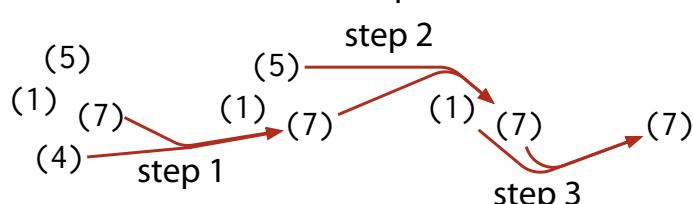
Goal: compute the maximum of a set of numbers

- Objects:** numbers
- One **reaction rule:** pick two numbers randomly, replace them with the bigger one
- Algorithm:** apply the reaction rule until there is only one number left: the result

reaction network



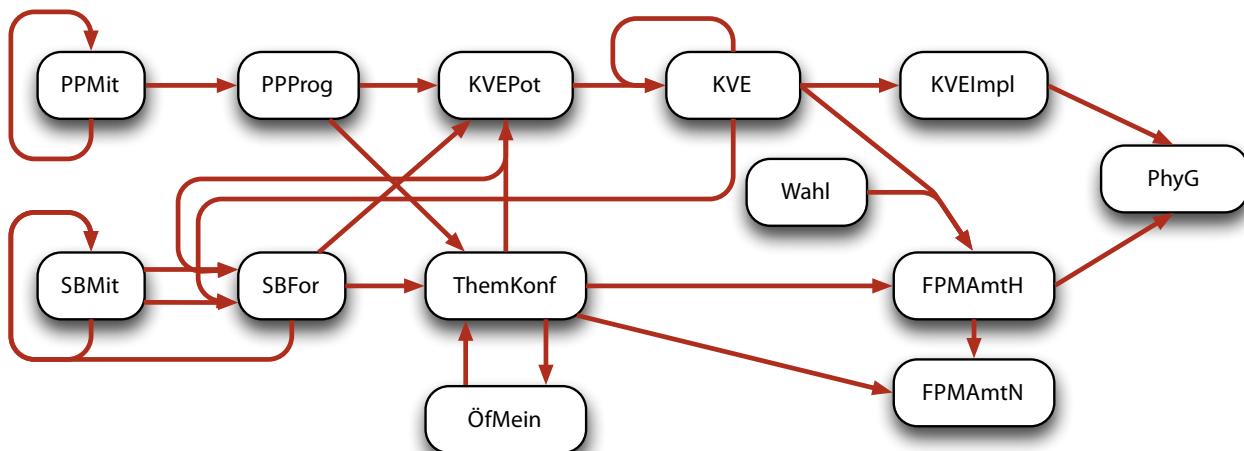
example



Reactive System Examples

A reactive model of the political system according to Luhmann

- Objects:** PPMit (Politische Partei, Mitglied), PPProg (PP, Programm), SBMit (Soziale Bewegung, Mitglied), SBFor (SB, Forderung), KVE (kollektiv verbindliche Entscheidungen), KVEpot (potentielle KVE), FPMAmtH (formale politische Macht, hohe Ämter), FPMAmtN (formale politische Macht, niedrige Ämter), KVEimpl (KVE, implementierung), PhyG (physische Gewalt), ThemKonf (thematischer Konflikt), ÖfMein (öffentliche Meinung), Wahl

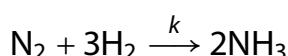


Reactive System Examples

Chemistry - science of matter

Chemistry describes the reaction of molecules

- Objects = molecules:** ensemble of chemically identical molecular entities i.e., with identical atomic structure, e.g. H₂O (water), NH₃ (ammonia)
- Reaction rules = chemical reactions:** mass-conserving transformation from a multiset of “input” molecules (**educts**, or **reactants**) to a multiset of “output” molecules (**products**)



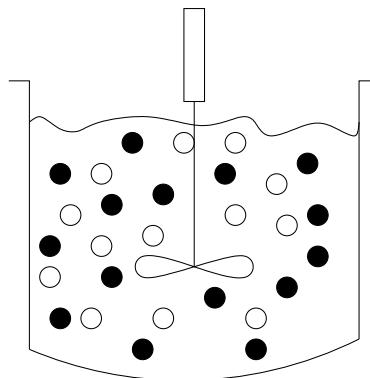
k is the **kinetic coefficient** or **rate coefficient** of the reaction. It determines the reaction's **speed**.

Reactive System Examples

Chemistry - science of matter

Other chemical terms that we will reuse later:

- **Multiset:** a set in which elements may occur more than once
e.g. the multiset $\mathcal{M} = \{A, A, B, B, B, C\}$ contains 6 molecular instances of 3 molecular species A, B, and C
- **Reactor or vessel:** the container (e.g. test tube) where the reactions take place
- **Well-stirred vessel:** all molecules have the same probability to collide with each other (no spatial considerations)



Artificial Chemistry (AC)

A formal description of reactive systems

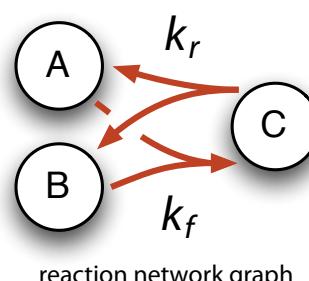
An **Artificial Chemistry** ([Dittrich, 2001](#)) is defined by the **tuple** $(\mathcal{S}, \mathcal{R}, \mathcal{A})$

- \mathcal{S} = set of possible **molecular species**

e.g. $\mathcal{S} = \{A, B, C\}$

- \mathcal{R} = set of **reaction rules** that determine how to operate on molecules on \mathcal{S}

e.g. $\mathcal{R} = \{A + B \xrightarrow{k_f} C, C \xrightarrow{k_r} A + B\}$



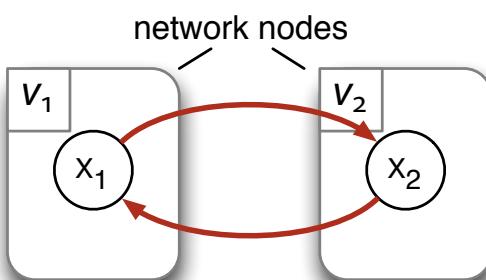
- \mathcal{A} = **reactor algorithm** that describes how to apply the rules \mathcal{R} to a **multiset** \mathcal{M} of molecule instances from \mathcal{S}

- **deterministic** vs. **stochastic** algorithms

Chemical Networking Protocols (CNP)

Artificial Chemistry in the distributed context of a computer network

A **Chemical Networking Protocol**
is an **Artificial Chemistry**
where each **species**
is assigned to a node
in the computer network



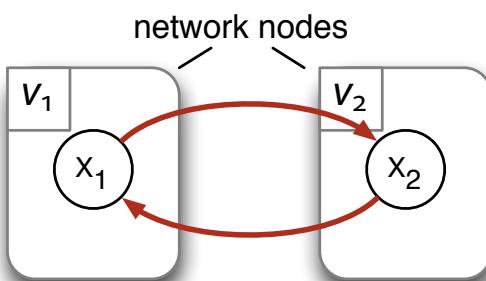
$$\mathcal{S} = \{X_1, X_2\}$$

$$\mathcal{R} = \{X_1 \rightarrow X_2, X_2 \rightarrow X_1\}$$

Communication: modeled by a reaction that
consumes molecule(s) in one node (the source node)
and produces molecule(s) in other node(s) (destination nodes)

Chemical Networking Protocols (CNP)

Artificial Chemistry in the distributed context of a computer network



$$\mathcal{S} = \{X_1, X_2\}$$

$$\mathcal{R} = \{X_1 \rightarrow X_2, X_2 \rightarrow X_1\}$$

- Still open, to be discussed later:
 - How can distributed chemical reactions perform some useful service?
 - What do you think the above reaction network does?
 - Even if we understand the principle of chemical reactions, can we **design CNPs** to fulfill given protocol requirements?
 - Promise: The **dynamic behavior of CNPs is easier to analyze** than the dynamics of traditional protocols?

Contents (two lectures)

Stochastic protocols

from Ethernet to reactive systems: an overview

Chemical networking

a formal approach to describe stochastic reactive systems

The three levels of structuring (distributed) computations

- **symbolic**

we know it all: $1 + 2 \Rightarrow 3$; HTTP GET + File \Rightarrow 200 OK

- **dynamic**

more challenging: global behavior emerging from the *dynamic* behavior of network nodes

- **organizational**

use the long-term trend of dynamic behavior to carry out computation

From robust to self-healing protocols

combining all the three levels to achieve robust protocol behavior and protocol code that heals itself

Contents (two lectures)

Stochastic protocols

from Ethernet to reactive systems: an overview

Chemical networking

a formal approach to describe stochastic reactive systems

The three levels of structuring (distributed) computations

- **symbolic**

we know it all: $1 + 2 \Rightarrow 3$; HTTP GET + File \Rightarrow 200 OK

- **dynamic**

more challenging: global behavior emerging from the *dynamic* behavior of network nodes

- **organizational**

use the long-term trend of dynamic behavior to carry out computation

From robust to self-healing protocols

combining all the three levels to achieve robust protocol behavior and protocol code that heals itself

The Three Levels of Structuring Chemical Computation

Applicable to local and distributed computation

Symbolic Chemical Computation

The result of the computation is found in (the structure) of a single molecule instance

- Examples

Dynamic Chemical Computation

The result of the computation is found in the number of instances of a species

- Dynamic behavior of (artificial) chemical reactions - the Law of Mass Action
- Equilibrium properties
- Simulating the dynamic behavior on a computer - the Gillespie algorithm
- Chemical Networking Protocols

Organizational Chemical Computation

The result of the computation is found in the presence/absence of instances of molecular species

- Examples

The Three Levels of Structuring Chemical Computation

Applicable to local and distributed computation

Symbolic Chemical Computation

The result of the computation is found in (the structure) of a single molecule instance

- Examples

Dynamic Chemical Computation

The result of the computation is found in the number of instances of a species

- Dynamic behavior of (artificial) chemical reactions - the Law of Mass Action
- Equilibrium properties
- Simulating the dynamic behavior on a computer - the Gillespie algorithm
- Chemical Networking Protocols

Organizational Chemical Computation

The result of the computation is found in the presence/absence of instances of molecular species

- Examples

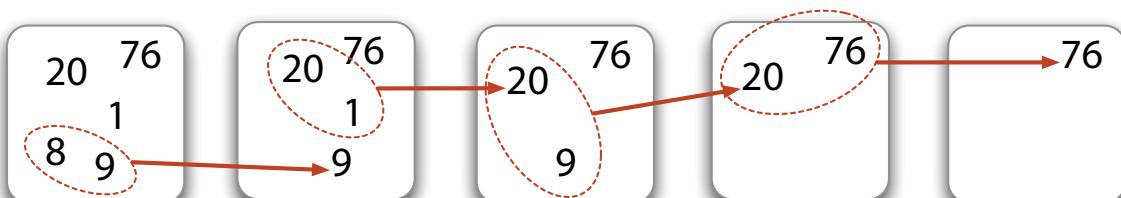
Symbolic Chemical Computation

- Program: set of “**molecules**” (data and/or code) floating around in a well-stirred vessel
- Execution:
 - “Molecules” are **consumed** and **produced** by “**chemical reactions**”
 - Computation stops when no more reactions can take place (inert solution)
- Information: The molecules are usually enriched with symbolic information (values, strings). This **structure** of an individual molecule instance **contains the result**.
- Examples: Gamma, CHAM, Membrane Computing (P-Systems), Fraglets

Symbolic Chemical Computation

Gamma - Example #1

- Gamma: Programming by multiset transformations
Banâtre et al, *A Generalized Higher-Order Chemical Computation Model with Infinite and Hybrid Multisets*, 1st International Workshop on New Developments in Computational Models, 5-14, 1996
- Goal: Design of parallel programs without superfluous sequential steps
- Example: Compute the **maximum** number of a set of numbers
 - Species: $\mathcal{S} = \mathbb{N}$ (natural numbers)
 - Reactions: $\mathcal{R} = \{\max\}$ $\max(x,y)$: **replace** x,y **by** x **if** $x \geq y$
 - Algorithm: \mathcal{A} : pick two molecules randomly and apply the reaction

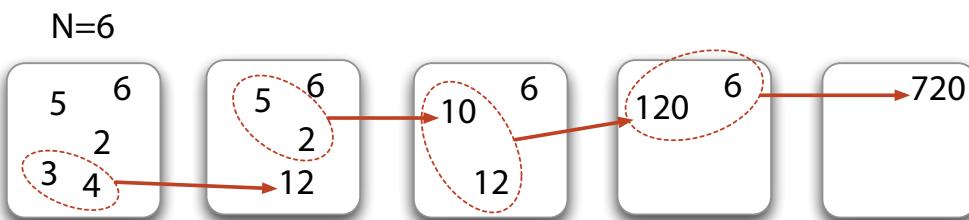


Symbolic Chemical Computation

Gamma - Example #2

Another example: What does the following Artificial Chemistry compute?

- Species: $\mathcal{S} = \{2, 3, \dots, N\}$
- Reactions: $\mathcal{R} = \{\text{mult}\}$ $\text{mult}(x,y)$: **replace** x,y by $x \cdot y$
- Algorithm: \mathcal{A} : initialize vessel with one instance of each species, pick two molecules randomly and apply the reaction



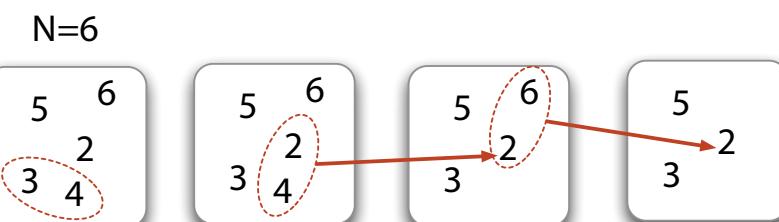
This chemistry computes the **factorial of N** .

Symbolic Chemical Computation

Gamma - Example #3

And yet another example: What is the result of the following chemistry?

- Species: $\mathcal{S} = \{2, 3, \dots, N\}$
- Reactions: $\mathcal{R} = \{\text{rem}\}$ $\text{rem}(x,y)$: **replace** x,y by y if $\text{multiple}(x,y)$
- Algorithm: \mathcal{A} : initialize vessel with one instance of each species, pick two molecules randomly and apply the reaction



This chemistry computes the **prime numbers up to N** .

Symbolic Chemical Computation

Fraglets

- Remember from earlier lecture:

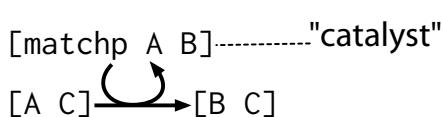
Active Network Architectures: A Programmable Network

- Fraglets is an active network approach
- Nodes send executable code (fraglets)
- New functionality (node behavior) can be added/defined at runtime
- The sender can run a protocol without needing the receiver to know anything about it
- The network becomes programmable

Symbolic Chemical Computation

Fraglets

- Quick repetition of Fraglets — some rewriting rules:



- Can we define Fraglets as an Artificial Chemistry ($\mathcal{S}, \mathcal{R}, \mathcal{A}$)?

Symbolic Chemical Computation

Fraglets

Fraglets is an Artificial Chemistry $(\mathcal{S}, \mathcal{R}, \mathcal{A})$ where

- the **species** are strings over the symbol alphabet Σ : $\mathcal{S} = \{s \in \Sigma^*\}$;
- the alphabet contains all Fraglets instructions and tags
 $\Sigma = \{\text{match}, \text{nop}, A, B, \dots\}$;
- the set of species is defined **implicitly** by the instruction set; and where
- the set of **reaction rules** \mathcal{R} is also defined **implicitly** by the interaction possibilities of all strings.
- We are able to “**program**” the **reaction network** at run-time by injecting molecules (strings) into the vessel
- The **algorithm** \mathcal{A} follows the Law of Mass Action (e.g. Gillespie algorithm, discussed later)

The Three Levels of Structuring Chemical Computation

Applicable to local and distributed computation

Symbolic Chemical Computation

The result of the computation is found in (the structure) of a single molecule instance

- Examples

Dynamic Chemical Computation

The result of the computation is found in the number of instances of a species

- Dynamic behavior of (artificial) chemical reactions - the Law of Mass Action
- Equilibrium properties
- Simulating the dynamic behavior on a computer - the Gillespie algorithm
- Chemical Networking Protocols

Organizational Chemical Computation

The result of the computation is found in the presence/absence of instances of molecular species

- Examples

The Three Levels of Structuring Chemical Computation

Applicable to local and distributed computation

Symbolic Chemical Computation

The result of the computation is found in (the structure) of a single molecule instance

- Examples

Dynamic Chemical Computation

The result of the computation is found in the number of instances of a species

- Dynamic behavior of (artificial) chemical reactions - the Law of Mass Action
- Equilibrium properties
- Simulating the dynamic behavior on a computer - the Gillespie algorithm
- Chemical Networking Protocols

Organizational Chemical Computation

The result of the computation is found in the presence/absence of instances of molecular species

- Examples

Dynamical Chemical Computation

Compute with concentrations (=molecule quantities)

- Program: set of “**molecules**” that float in a solution
- Execution:
 - “molecules” are **consumed** and **produced** in “**chemical reactions**”
 - Computation (usually) never stops, **continuous execution**.
- Information:
 - Encoded by the **multiplicity of molecules** (individual instance not important)
 - Macroscopic system state: vector of concentration of all species
 - Computation: change in concentration (state transition)
 - **Result** of computation **ready** when system reaches **equilibrium**
- Examples: Biochemistry, Fraglets

Dynamical Chemical Computation

Compute with concentrations (=molecule quantities)

- Dynamic behavior of (artificial) chemical reactions - the Law of Mass Action
- Equilibrium properties
- Simulating the dynamic behavior on a computer - the Gillespie algorithm
- Computing with concentrations - some example reaction networks
- Chemical Networking Protocols

The Law of Mass Action

The reaction **rate** is proportional to the concentration of its reactants

In chemistry, the **Law of Mass Action** defines the rate of reactions:

- On the level of individual molecules:

*The more molecules are in a fixed volume,
the more frequently they collide.*

- On the macroscopic level:

*The higher the concentration of the reactants of a reaction,
the more often this reaction occurs.*

- For Chemical Networking Protocols:

*The higher the multiplicity of molecules in a node,
the faster that node sends molecules to its neighbors.*

The Law of Mass Action

The reaction **rate** is proportional to the concentration of its reactants

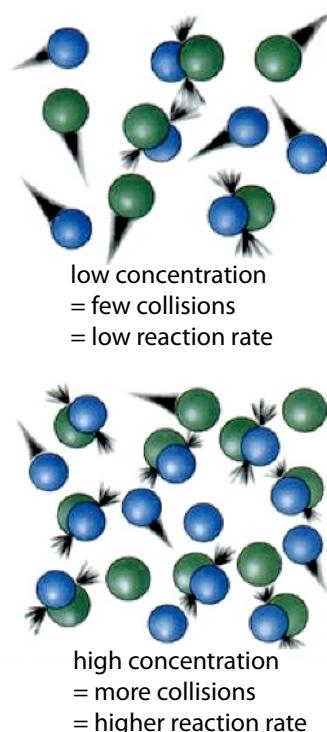
In a well stirred reactor, the speed (rate) of a reaction such as



is given by

$$v = k[A]^2[B]^3$$

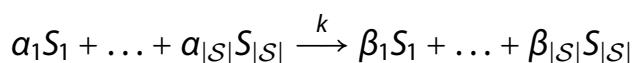
- ❖ [A] and [B] are the concentrations of reactants A and B, respectively
- ❖ k is the **kinetic coefficient** or **rate coefficient** of the reaction
- ❖ v is the number of reaction events per second



The Law of Mass Action

The reaction **rate** is proportional to the concentration of its reactants

Generalized formula for a chemical reaction: ($|S|$, the number of species, is the max. number of reactants)



with speed

$$v = k \prod_{i=1}^{|S|} [S_i]^{\alpha_i}$$

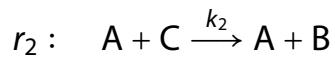
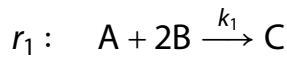
- ❖ $[S_i]$ = concentration of the i -th species
- ❖ α_i, β_i = **stoichiometric coefficients**: number of molecules of S_i required (resp. produced) in the chemical reaction

The Law of Mass Action

Concentration dynamics

The concentration dynamics of all species can be described by a system of **Ordinary Differential Equations (ODEs)**.

Example, for these reactions,



using the law of mass action, the ODE system is

$$\frac{d[A]}{dt} = -v_1$$

$$\frac{d[A]}{dt} = -k_1[A][B]^2$$

$$\frac{d[B]}{dt} = -2v_1 + v_2$$

$$\frac{d[B]}{dt} = -2k_1[A][B]^2 + k_2[A][C]$$

$$\frac{d[C]}{dt} = v_1 - v_2$$

$$\frac{d[C]}{dt} = k_1[A][B]^2 - k_2[A][C]$$

There is one equation for each species.

Each equation sums the ingress and egress reaction rates according to the LoMA

The Law of Mass Action

Concentration dynamics

$$\frac{d[A]}{dt} = -v_1$$

$$\frac{d[A]}{dt} = -k_1[A][B]^2$$

$$\frac{d[B]}{dt} = -2v_1 + v_2$$

$$\frac{d[B]}{dt} = -2k_1[A][B]^2 + k_2[A][C]$$

$$\frac{d[C]}{dt} = v_1 - v_2$$

$$\frac{d[C]}{dt} = k_1[A][B]^2 - k_2[A][C]$$

or, in vector notation

$$\frac{d\vec{x}}{dt} = \mathbf{M}\vec{v}$$

$$\mathbf{M} = \begin{pmatrix} -1 & 0 \\ -2 & 1 \\ 1 & -1 \end{pmatrix} \quad \vec{v} = \begin{pmatrix} k_1[A][B]^2 \\ k_2[A][C] \end{pmatrix}$$

- $\mathbf{M} = [\beta_{ij} - \alpha_{ij}]_{ij}$: stoichiometric matrix
- $v = k \prod_{i=1}^{|S|} [S_i]^{\alpha_i}$: reaction speed/rate according to the law of mass action

Dynamical Chemical Computation

Compute with concentrations (=molecule quantities)

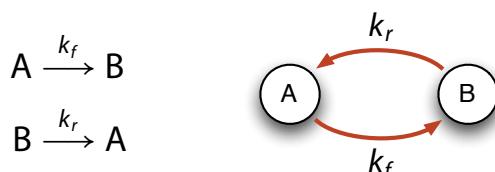
- Dynamic behavior of (artificial) chemical reactions - the Law of Mass Action
- Equilibrium properties
- Simulating the dynamic behavior on a computer - the Gillespie algorithm
- Computing with concentrations - some example reaction networks
- Chemical Networking Protocols

Equilibrium

At equilibrium, the concentrations of all species do not change anymore:

$$\frac{d\vec{x}}{dt} \equiv \vec{0}$$

Example:



At equilibrium:

$$\begin{aligned} \frac{d[A]}{dt} &= -k_f[A] + k_r[B] = 0 \\ \frac{d[B]}{dt} &= +k_f[A] - k_r[B] = 0 \end{aligned} \Rightarrow k_f[A] = k_r[B] \Rightarrow K = \frac{k_f}{k_r} = \frac{[B]}{[A]}$$

K: equilibrium constant

Central goal of Chemical Networking Protocols: design reaction networks that have an equilibrium (stable fixed point) and that present their solution there

Dynamical Chemical Computation

Compute with concentrations (=molecule quantities)

- Dynamic behavior of (artificial) chemical reactions - the Law of Mass Action
- Equilibrium properties
- Simulating the dynamic behavior on a computer - the Gillespie algorithm
- Computing with concentrations - some example reaction networks
- Chemical Networking Protocols

Simulation of the LoMA on a Computer

Deterministic approach: numeric ODE integration

- ODE system fully describes the dynamics of the concentration vector
- The computer could integrate the ODEs numerically to compute the reaction system's deterministic trajectory
- Could be implemented with existing math packages, e.g. Matlab, Scilab, etc.
- Advantages:
 - Accurate/fast for a small set of species, each with a large number of instances
- Shortcomings:
 - Cannot capture some qualitative phenomena arising for low concentrations e.g. the complete loss of a species due to stochastic fluctuations
 - Cannot easily deal with the dynamic creation of new species (novelty)

Simulation of the LoMA on a Computer

Stochastic approach: exact stochastic simulation algorithms

- Take stochastic effects into account:

In real chemistry: Brownian motion results in random molecule collision

- Some well-known stochastic reaction algorithms:

- **Gillespie's Algorithm**

Gillespie, *Exact Stochastic Simulation of Coupled Chemical Reactions*, Journal of Physical Chemistry 81(25): 2340--2361, 1977.

- **StochSim**

Morton-Firth et al, *Predicting temporal fluctuations in an intracellular signaling pathway*, Journal of Theoretical Biology, 192: 117-128, 1998.

- **Next Reaction Method**

Gibson et al, *Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels*. Journal of Physical Chemistry A, 104(9): 1876–1889, 2000.

Simulation of the LoMA on a Computer

Stochastic approach: exact stochastic simulation algorithms

- Advantages:

- Realistic even for small numbers of molecules
- Can easily deal with the dynamic creation of new species

- Shortcomings:

- Not scalable to large number of molecules
- Not scalable to large number of possible reactions

A Simplistic Stochastic Reaction Algorithm

- In each iteration:

- Pick molecules at random:

If there is a reaction rule among them:

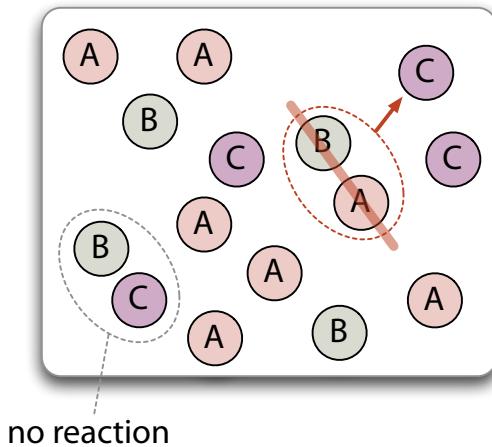
- remove the reactants
- add the products

- Advantage: very easy

- Shortcoming: inefficient if there are few reactions but many species

$$\mathcal{S} = \{A, B, C\}$$

$$\mathcal{R} = \{A + B \rightarrow C, C \rightarrow A + B\}$$



Exact Stochastic Simulation - Gillespie's Algorithm

Instead of “trying out” reactions in each iteration, calculate **which** reaction occurs **when**

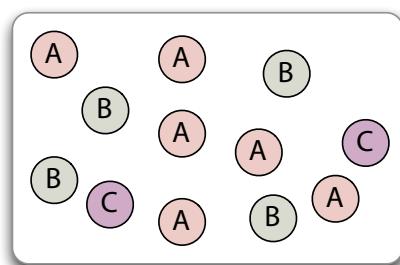
1. Which reaction is likely to occur next?

- Intuitively: Reactions with a higher number of reactants are more frequently executed
- Algorithm:

- Calculate the **propensity** (weight) of all reactions: a_i
= microscopic reaction rate
= probability that reaction occurs in the next infinitesimal time dt

- Randomly select a reaction based on the distribution of the reaction propensities

$$\begin{aligned}\mathcal{S} &= \{A, B, C\} \\ \mathcal{R} &= \{A + B \xrightarrow{k_1} C, C \xrightarrow{k_2} A + B\} \\ k_1 &= 1 \quad k_2 = 10\end{aligned}$$



$$n_A = 6 \quad n_B = 4 \quad n_C = 2$$

$$\begin{aligned}a_1 &= k_1 n_A n_B = 1 \cdot 6 \cdot 4 = 24 \\ a_2 &= k_2 n_C = 10 \cdot 2 = 20\end{aligned}$$

$$\begin{aligned}b) \quad P(r_1) &= \frac{a_1}{a_1 + a_2} = 0.545 \\ P(r_2) &= \frac{a_2}{a_1 + a_2} = 0.454\end{aligned}$$

Exact Stochastic Simulation - Gillespie's Algorithm

Instead of "trying out" reactions in each iteration, calculate **which** reaction occurs **when**

2. **When** does the next reaction occur?

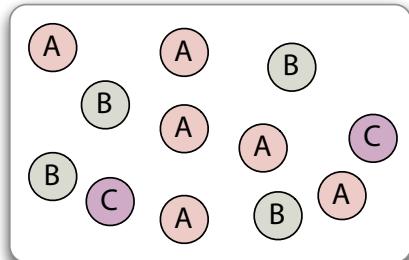
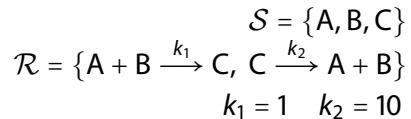
- Intuitively: Reactions occur in parallel:
high concentration \Rightarrow short time interval
- Algorithm:

- Calculate the sum of all reaction propensities:

$$a_0 = \sum_{r \in \mathcal{R}} a_r$$

- Next reaction interval: Draw an exponentially distributed random variable:

$$\tau \sim \text{Exp}\left(\frac{1}{a_0}\right)$$



$$n_A = 6 \quad n_B = 4 \quad n_C = 2$$

$$a_1 = k_1 n_A n_B = 1 \cdot 6 \cdot 4 = 24$$

$$a_2 = k_2 n_C = 10 \cdot 2 = 20$$

$$a) \quad a_0 = a_1 + a_2 = 24 + 20 = 44$$

$$b) \quad E[\tau] = \frac{1}{a_0} = 22.7 \times 10^{-3}$$

Exact Stochastic Simulation - Gillespie's Algorithm

Formal algorithm to determine **which** reaction occurs next and **when**:

- **Which** reaction occurs next?
 - Draw a uniform random variable $u \sim U(0, \sum_{j=1}^{|\mathcal{R}|} w_j)$
 - Pick reaction r_i , such that $\sum_{j=1}^{i-1} a_j < u \leq \sum_{j=1}^i a_j$
- **When** does the next reaction occur?
 - Draw an exponentially distributed random variable as reaction interval $\tau \sim \text{Exp}(\sum_{j=1}^{|\mathcal{R}|} w_j)$
 - Schedule the next reaction for time $t = t_{\text{now}} + \tau$
- Gillespie's algorithm is widely used to simulate artificial chemistries
- Inefficient if many reaction and a small number of species

Dynamical Chemical Computation

Compute with concentrations (=molecule quantities)

- Dynamic behavior of (artificial) chemical reactions - the Law of Mass Action
- Equilibrium properties
- Simulating the dynamic behavior on a computer - the Gillespie algorithm
- Computing with concentrations - some example reaction networks
- Chemical Networking Protocols

Arithmetic Computation with Concentrations

Example #1: Proportional Scaling

Goal: Map concentration of one species proportionally to another one

- Chemical reaction equations:



- Differential equations in equilibrium:

$$\vec{0} = \mathbf{M}\vec{v} = \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} k_1 n_X \\ k_2 n_Y \end{pmatrix} = \begin{pmatrix} 0 \\ k_1 n_X - k_2 n_Y \end{pmatrix}$$

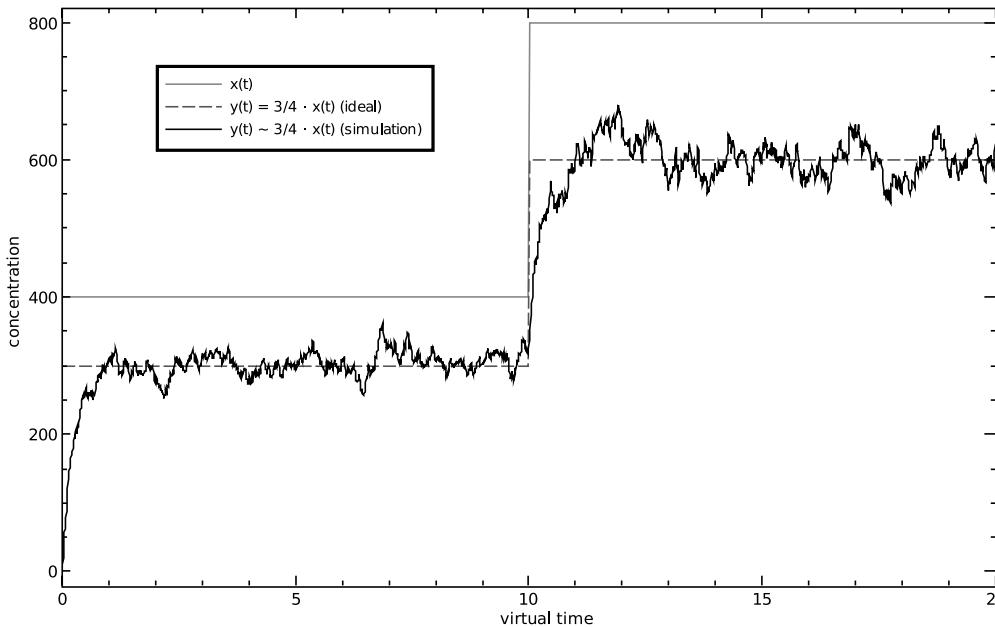
- Steady-state solution:

$$n_Y^* = \frac{k_1}{k_2} n_X^*$$

Arithmetic Computation with Concentrations

Example #1: Proportional Scaling

Response of n_Y to changes of n_X for $k_1=3$, $k_2=4$

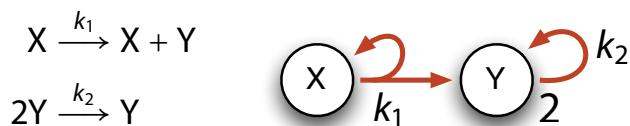


Arithmetic Computation with Concentrations

Example #2: Square Root

Goal: Map the concentration of species X to Y, following

- Chemical reaction equations:



- Differential equations in equilibrium:

$$\vec{0} = \mathbf{M}\vec{v} = \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} k_1 n_X \\ k_2 n_Y^2 \end{pmatrix} = \begin{pmatrix} 0 \\ k_1 n_X - k_2 n_Y^2 \end{pmatrix}$$

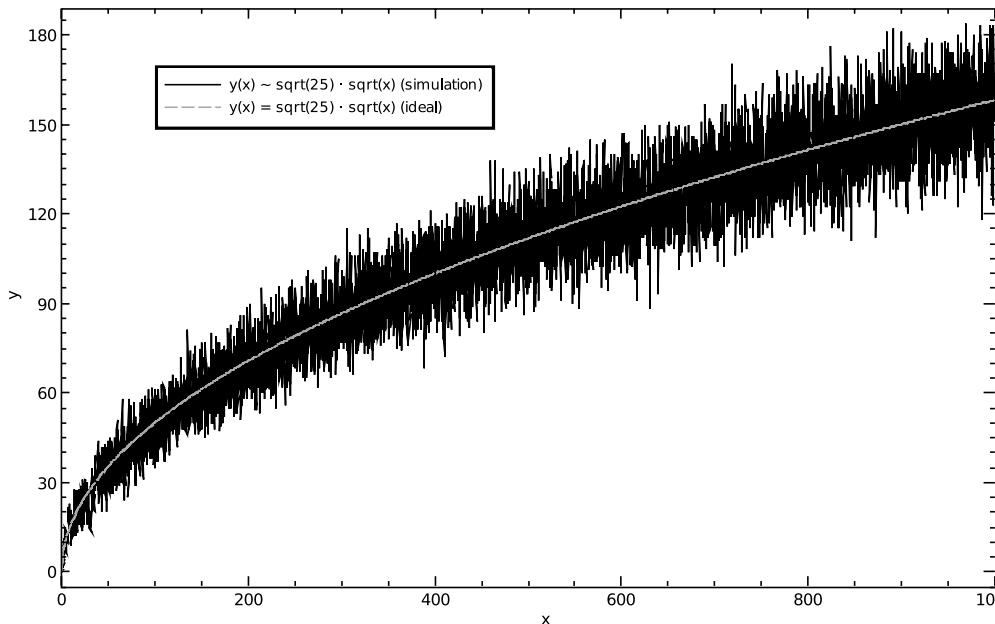
- Steady-state solution:

$$n_Y^* = \sqrt{\frac{k_1}{k_2} n_X^*}$$

Arithmetic Computation with Concentrations

Example #2: Square Root

Response of n_Y to changes of n_X for $k_1=25, k_2=1$



Dynamical Chemical Computation

Compute with concentrations (=molecule quantities)

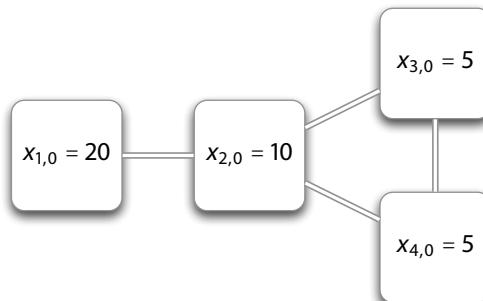
- Dynamic behavior of (artificial) chemical reactions - the Law of Mass Action
- Equilibrium properties
- Simulating the dynamic behavior on a computer - the Gillespie algorithm
- Computing with concentrations - some example reaction networks
- Chemical Networking Protocols
 - An introductory example: The chemical *Disperser* protocol
 - Properties of Chemical Networking Protocols
 - Dynamic Analysis of Chemical Networking Protocols
 - Design of Chemical Networking Protocols

The Chemical Disperser Protocol

Distributed Computation of an Aggregated Value

Problem Statement:

- Each node contains an initial value (e.g. measurement of temperature)



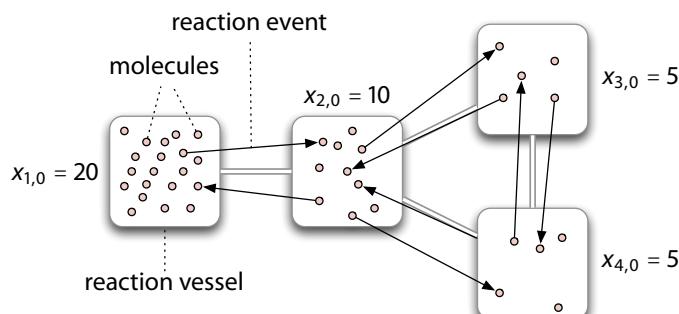
- **Compute the average value in a distributed way** (same as for Push-Sum)
- At the end, each node should know this **aggregated information**
- How could this be achieved with distributed chemical reactions, and by calculating with concentrations?

The Chemical Disperser Protocol

Distributed Computation of an Aggregated Value

Chemical mind set: think “molecules” and “reactions”

- One reaction per node and link
- Reaction: transform molecule in one node to molecule in corresp. neighbor

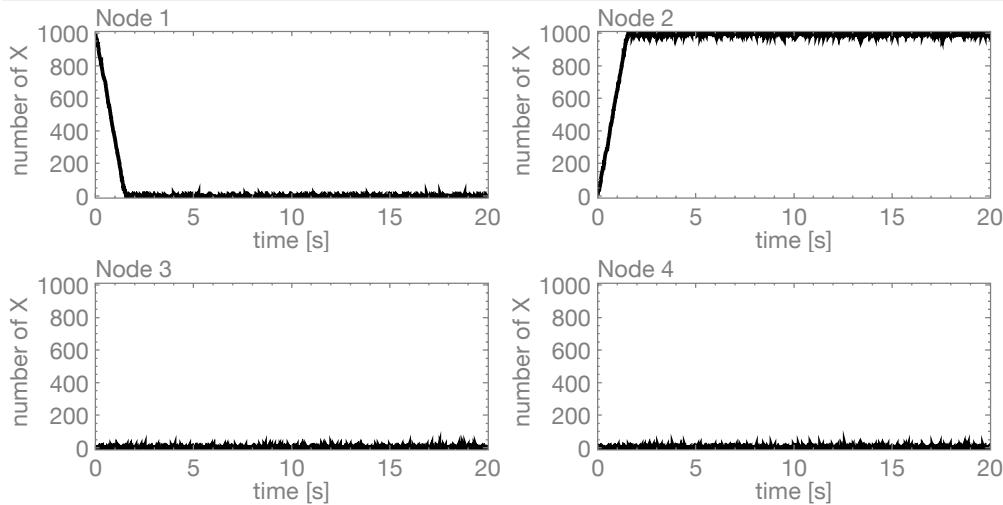


- What happens if we execute the reactions as fast as possible?
- What happens if we execute the reactions with Gillespie (according to the LoMA)?

The Chemical Disperser Protocol

AFAP Algorithm: Execute reactions as fast as possible

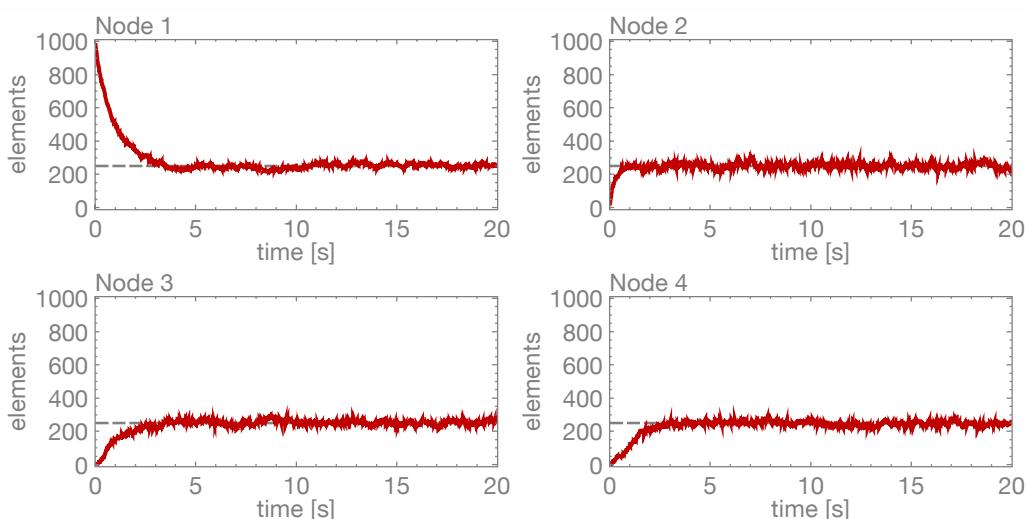
- Nodes with many neighbors accumulate molecules
- Systems does *not* compute the right result (i.e. the average value)



The Chemical Disperser Protocol

Gillespie Algorithm: Execute reactions according to the LoMA

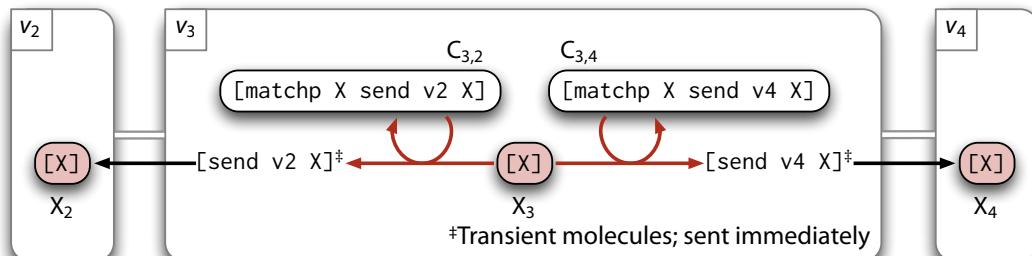
⇒ Equilibrium: all nodes contain the same (the average) number of molecules



The Chemical Disperser Protocol

Implementation in Fraglets

- The multiplicity of the “passive” data fraglet $[X]$ represents the value
- For each link, there is one instance of an “active” catalyst fraglet $[\text{matchp } X \text{ vj } X]$:
 - The catalyst reacts with a random data fraglet $[X]$,
 - regenerates itself,
 - and produces a $[\text{send vj } X]$ fraglet that sends the data fraglet to the corresponding neighbor.



Dynamical Chemical Computation

Compute with concentrations (=molecule quantities)

- Dynamic behavior of (artificial) chemical reactions - the Law of Mass Action
- Equilibrium properties
- Simulating the dynamic behavior on a computer - the Gillespie algorithm
- Computing with concentrations - some example reaction networks
- Chemical Networking Protocols
 - An introductory example: The chemical *Disperser* protocol
 - Properties of Chemical Networking Protocols
 - Dynamic Analysis of Chemical Networking Protocols
 - Design of Chemical Networking Protocols

Properties of Chemical Networking Protocols

- We build a distributed reaction network that computes with concentrations:
 - Values are not represented by symbolic (binary) numbers but by the multiplicity/**concentration** of molecules
 - Information is not conveyed symbolically *inside* packets but by the **packet rate**
 - Advantage: Higher robustness:
Packet loss only marginally disturbs the system
 - Disadvantage: Higher message complexity
- Able to plan/study/proof the dynamics of an algorithm in the first place.
 - Simple protocol analysis? We need a formal mathematical framework
 - Simple protocol design? We need chemical design patterns

Dynamical Chemical Computation

Compute with concentrations (=molecule quantities)

- Dynamic behavior of (artificial) chemical reactions - the Law of Mass Action
- Equilibrium properties
- Simulating the dynamic behavior on a computer - the Gillespie algorithm
- Computing with concentrations - some example reaction networks
- Chemical Networking Protocols
 - An introductory example: The chemical *Disperser* protocol
 - Properties of Chemical Networking Protocols
 - Dynamic Analysis of Chemical Networking Protocols
 - Design of Chemical Networking Protocols

Dynamic Analysis of Chemical Networking Protocols

The general receipt

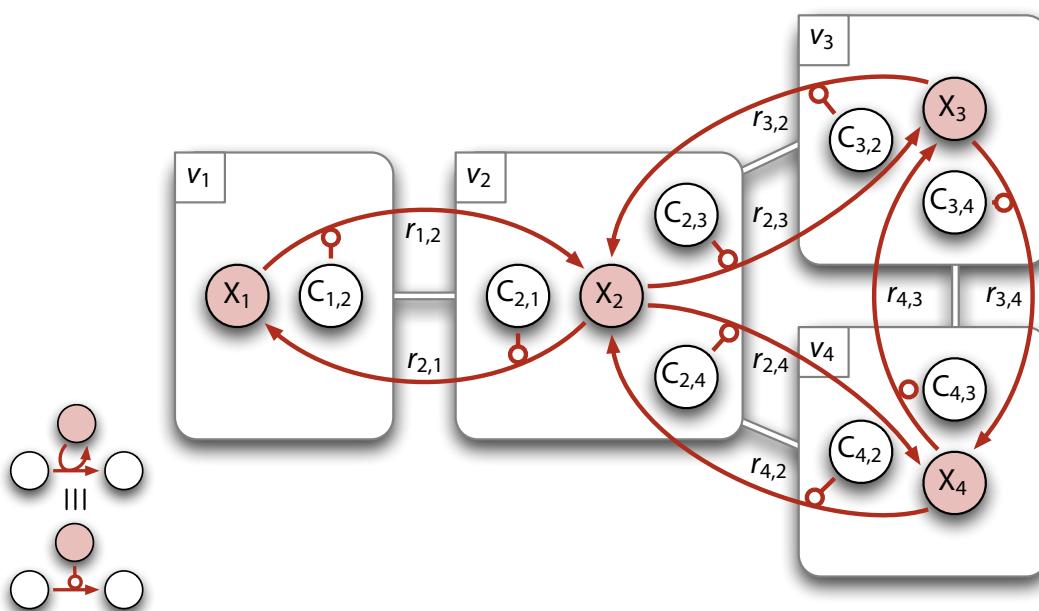
Analysis of the dynamic behavior of the distributed algorithm in Fraglets:

1. Draw the abstract reaction network (ignore the structure of the strings)
2. Write down the differential equations for the distributed reaction network
3. Find the equilibrium of the differential equation system
4. Show that the equilibrium is stable

Dynamic Analysis of Chemical Networking Protocols

Formal analysis of the *Disperser* Protocol

1. Draw the abstract reaction network (ignore the structure of the strings)



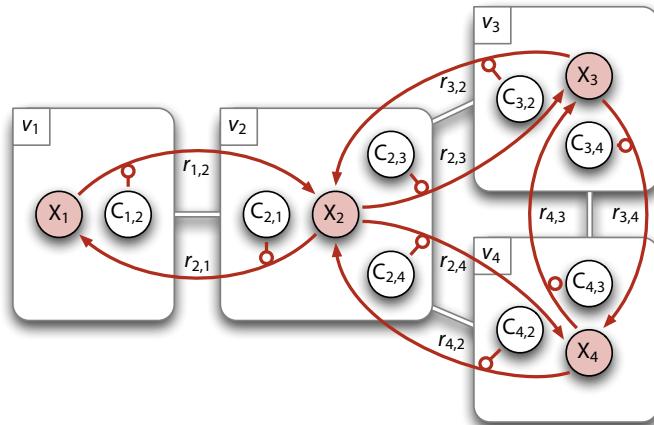
Dynamic Analysis of Chemical Networking Protocols

Formal analysis of the *Disperser* Protocol

2. Write down the differential equations for the distributed reaction network

Node 2:

$$\dot{x}_2 = \underbrace{x_1 + x_3 + x_4}_{\text{inflow}} - \underbrace{3x_2}_{\text{outflow}}$$



General:

$$\dot{x}_i = \underbrace{\sum_{j \in \mathcal{N}_i} x_j}_{\text{inflow}} - \underbrace{\deg(i)x_i}_{\text{outflow}}$$

$\deg(i)$ = degree of node i , i.e. number of neighbors

\mathcal{N}_i = set of neighbors of node i

Dynamic Analysis of Chemical Networking Protocols

Formal analysis of the *Disperser* Protocol

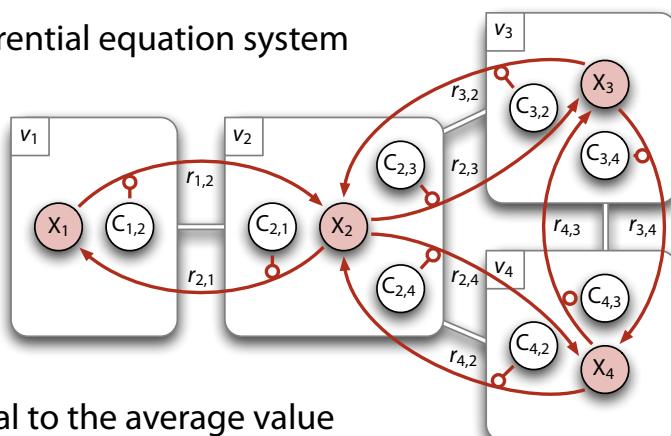
3. Find the equilibrium of the differential equation system

Node 2:

$$x_2 = \frac{x_1 + x_3 + x_4}{3}$$

General:

$$x_i^* = \frac{\sum_{j \in \mathcal{N}_i} x_j^*}{\deg(i)} = \langle x_j \rangle_{\mathcal{N}_i}$$



- ❖ The value of every node is equal to the average value of its neighbors.

- ❖ This only holds if the value of all nodes are equal: $x_i^* = x_j^* \quad \forall x_i, x_j \in \mathcal{V}$

- ❖ The total number of molecules is conserved $\sum_{i \in \mathcal{V}} x_i = x_{\text{tot}}$, hence: $x_i^* = \frac{x_{\text{tot}}}{|\mathcal{V}|}$

\mathcal{V} = set of all network nodes

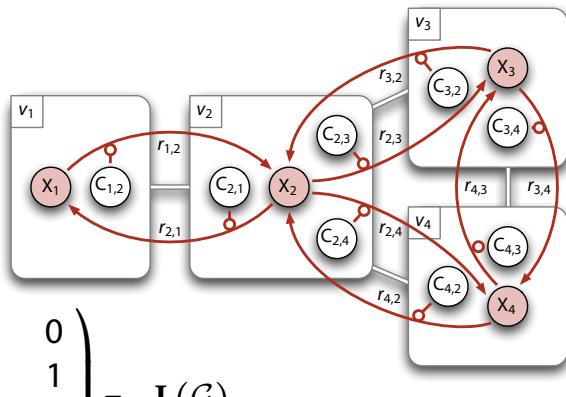
Dynamic Analysis of Chemical Networking Protocols

Formal analysis of the *Disperser Protocol*

4. Show that the equilibrium is stable:

- a) Linearize the ODE system around the fixed point = Compute the Jacobian matrix of the ODE system

$$\mathbf{J} = \begin{pmatrix} \frac{\partial \dot{x}_1}{\partial x_1} & \dots & \frac{\partial \dot{x}_4}{\partial x_4} \\ \vdots & \ddots & \vdots \\ \frac{\partial \dot{x}_1}{\partial x_4} & \dots & \frac{\partial \dot{x}_4}{\partial x_4} \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 1 & -3 & 1 & 1 \\ 0 & 1 & -2 & 1 \\ 0 & 1 & 1 & -2 \end{pmatrix} = -\mathbf{L}(\mathcal{G})$$



Laplacian of the network graph:
Diagonal elements: node degree
Off-diagonal elements: links

- b) Show that the eigenvalues of the Jacobian are negative:

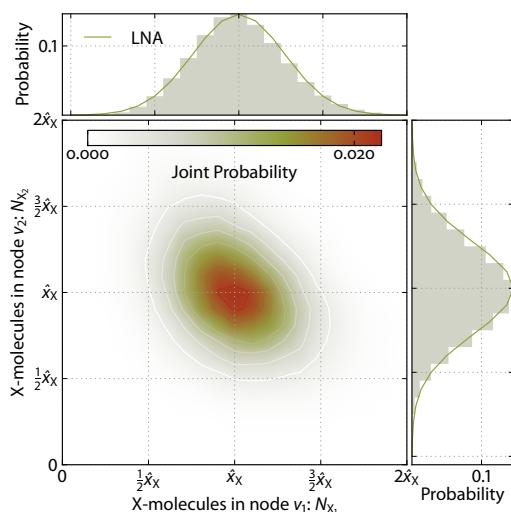
It is known that the eigenvalues of the graph Laplacian are all positive; the eigenvalues of \mathbf{J} are negative and the equilibrium is stable.

Inherent Noise in Stochastically Executed CNPs

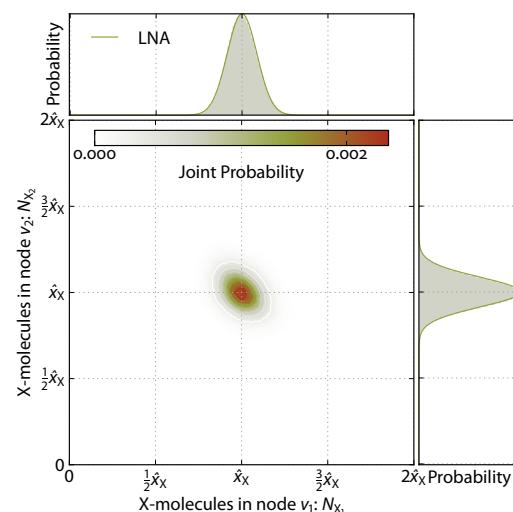
Noise in the result (=concentration) of the *Disperser Protocol*

The ODE system does not model the inherent stochastic noise of the algorithm. The amount of noise depends on the steady-state concentration of the "value".

lower concentration = more noise



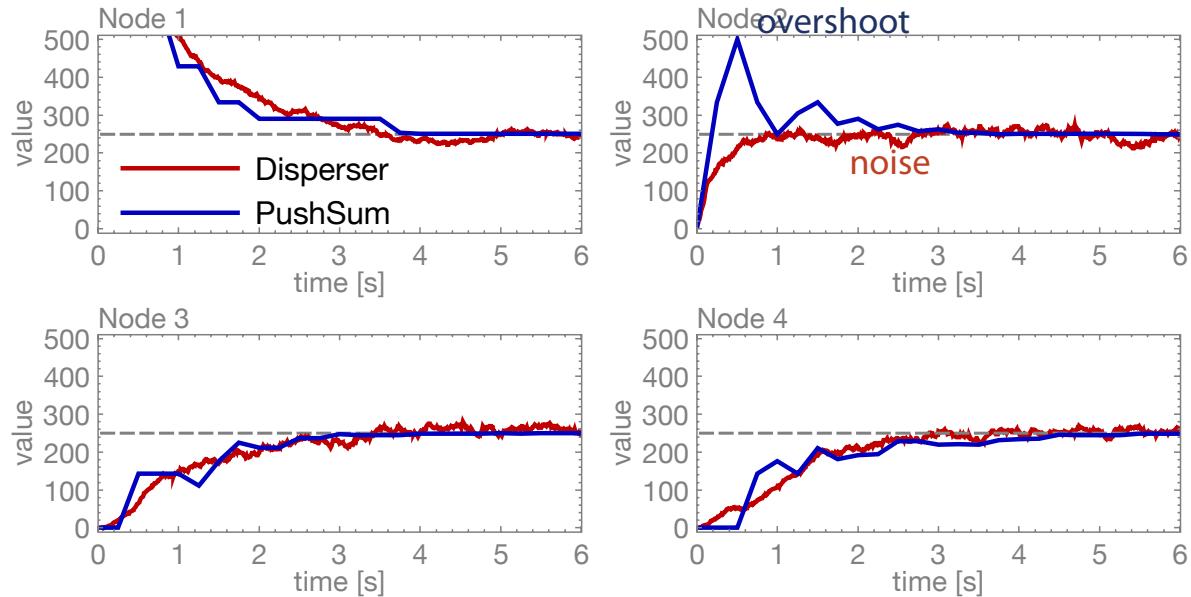
higher concentration = less noise



Comparison of Distributed Aggregation Algorithms

Gossip-style *Push-Sum* vs. Chemical *Disperser* Algorithm

- **Push-Sum:** Isochronous message exchange, few messages, overshoot
- **Disperser:** Dynamic message timing (LoMA), more messages, noise



Analysis of Chemical Networking Protocols in Fraglets

Summary

- Fraglets reaction networks emerge from the structure of the reacting strings
- Fraglets reactions are **scheduled stochastically** (Gillespie's Algorithm)
- However, the macroscopic dynamic behavior can be **approximated** by deterministic differential equations (ODE system)
- ODEs allow
 - predictions of the system's behavior,
 - **convergence proofs by finding the equilibrium!!!**
Such stability proofs are usually hard to find for traditional protocols, because the dynamics of symbolically computing networking algorithms is not considered at all, or there is not straight-forward mathematical model to do so.
- ODEs do not predict
 - the inherent noise of the stochastic scheduling algorithm
other mathematical tools are required: e.g. the Linear Noise Approximation

Dynamical Chemical Computation

Compute with concentrations (=molecule quantities)

- Dynamic behavior of (artificial) chemical reactions - the Law of Mass Action
- Equilibrium properties
- Simulating the dynamic behavior on a computer - the Gillespie algorithm
- Computing with concentrations - some example reaction networks
- Chemical Networking Protocols
 - An introductory example: The chemical *Disperser* protocol
 - Properties of Chemical Networking Protocols
 - Dynamic Analysis of Chemical Networking Protocols
 - Design of Chemical Networking Protocols

Design of Chemical Protocols

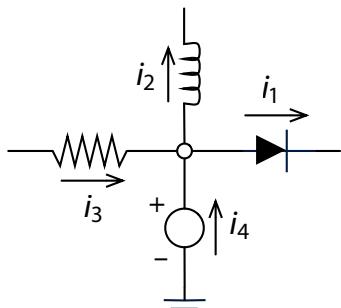
- How to come up with distributed algorithms?
- Algorithm design is never straight forward!
 - Needs intuition, experience, knowledge of the execution environment.
 - Do not expect general recipes.
- For **chemical algorithm design**, there are supporting principles:
 - Basic principle: Design **dynamics first, functional aspects second**
This is opposite to most traditional approaches
 - Use conservation laws:
e.g. Kirchhoff's current law
 - Combine well-understood design patterns:
There is a dozen of simple chemical reaction networks, each solving a specific (distributed) computation task.

Design of Chemical Protocols

Conservation Laws: Kirchhoff's Current Law

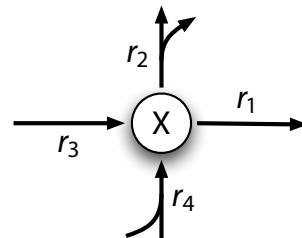
Kirchhoff's Current Law can be transposed to chemical reaction networks in equilibrium:

Electronic Circuit:



$$i_1 + i_2 = i_3 + i_4$$

Chemical Reaction Network:



$$\hat{r}_1 + \hat{r}_2 = \hat{r}_3 + \hat{r}_4$$

At equilibrium:

sum of current flowing into a node =
sum of current flowing out of that node

total influx of a species =
total efflux of that species

Design of Chemical Protocols

Conservation Laws: Molecule Conservation Loops

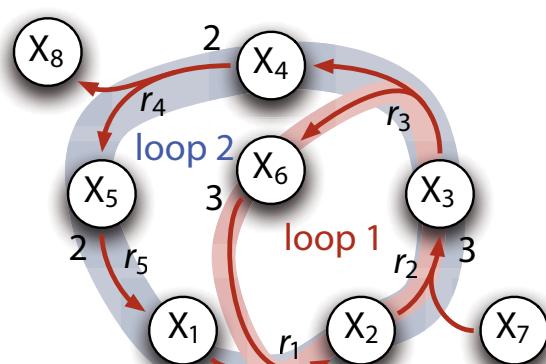
The number of molecules is conserved in reaction loops with these properties:

- Total number of molecules consumed along the loop = total number of molecules produced along the loop
- Every species of the loop must only be altered (consumed or produced) by reactions in this or any other conservation loop.

We implicitly used this rule
in the *Disperser* algorithm:

loop = reaction along a network link

Result: total number of molecules
remains constant



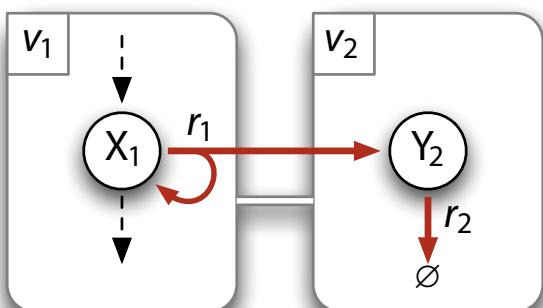
$$X_{X_1} + X_{X_2} + X_{X_3} + X_{X_4} + X_{X_5} = \text{const.}$$

$$X_{X_2} + X_{X_4} + X_{X_6} = \text{const.}$$

Design Patterns for Chemical Protocols

Remote Concentration Mirroring

- Goal: "Copy" the concentration of a local species to the concentration of a remote species.
- Use Kirchhoff's Current Law to show that at equilibrium, the concentration of Y follows the concentration of X



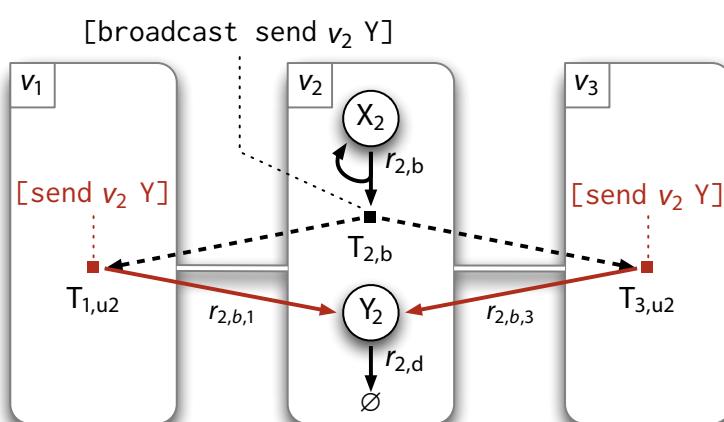
Current Law:

$$\begin{aligned}v_1 &= v_2 \\x_1^* &= y_2^*\end{aligned}$$

Design Patterns for Chemical Protocols

Neighbor Quantification

- Goal: Determine the number of neighbor nodes
- Approach: Broadcast active molecules to all neighbors. Once the molecules arrive, they send another molecule back.



T_i ■—> transient molecule, triggers broadcast
 $T_{j,ui}$ ■—> transient molecule, triggers unicast to origin

Current Law:

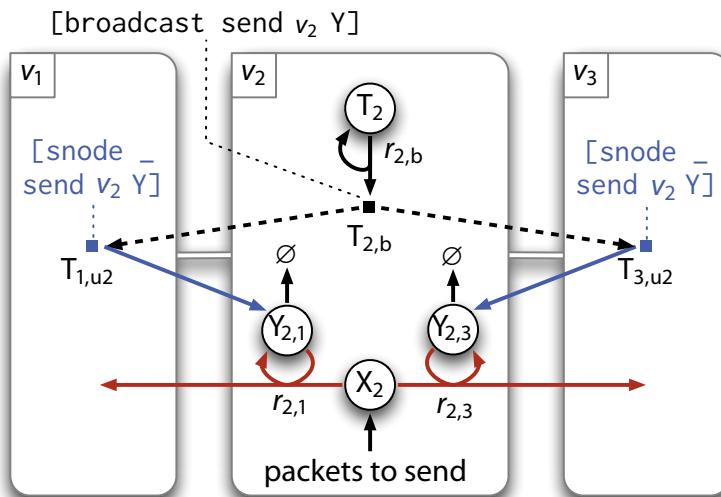
$$\sum_{j \in N_i} v_{2,b,j} = V_{2,d}$$

$$\deg(i) x_2^* = y_2^*$$

Design Patterns for Chemical Protocols

Anycast

- Goal: Send a data packet to any (one) neighbor node
- Approach: Use the molecular echo from before to build a steady concentration of transmission catalysts for every link



Artificial Chemical Computing

Summary of this lecture

- Artificial chemical computing is an unconventional way of organizing computation.
- Artificial Chemistries compute in an inherently parallel way
- There are three fundamentally different levels at which AChems **represent and process information**:
 1. Symbolically within the structure of the molecules
efficient, halts when result present
 2. As the concentrations of molecular species (multiplicity of molecules)
robust, perpetual reaction execution, result at equilibrium
 3. With the presence/absence of molecules of different species
even more robust, perpetual reaction execution, result at equilibrium

Chemical Networking Protocols

Summary of this lecture

- Chemical Networking Protocols are Artificial Chemistries in a distributed context
- CNPs aim at providing a framework to design, execute, and analyze stochastic protocols.
- The focus of CNPs are:
 - Predictable dynamic behavior
 - Solid design patterns, similar more to electrical circuit engineering than programming
 - Robust algorithms by computing the result in a dynamic equilibrium
 - Fraglets as execution environment that allows both, symbolic and dynamic computation.

Next Lecture

Stochastic protocols

from Ethernet to reactive systems: an overview

Chemical networking

a formal approach to describe stochastic reactive systems

The three levels of structuring (distributed) computations

- **symbolic**
we know it all: $1 + 2 \Rightarrow 3$; HTTP GET + File \Rightarrow 200 OK
- **dynamic**
more challenging: global behavior emerging from the *dynamic* behavior of network nodes
- **organizational**
use the long-term trend of dynamic behavior to carry out computation

From robust to self-healing protocols

combining all the three levels to achieve robust protocol behavior and protocol code that heals itself

Thank you