Simple-Minded Agents & Collective Intelligence

Rachel Huls, Dr. Xiang Huang
University of Illinois Springfield

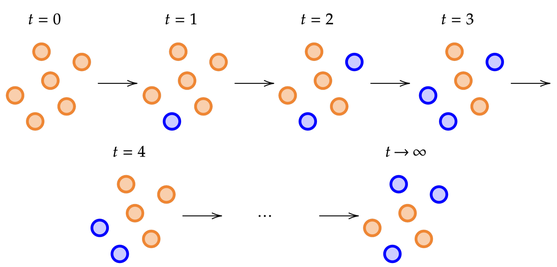
Overview

A **Population Protocol** (PP) is a distributed computing model for a population of finite-state agents. Random pairwise interactions allow agents to update their states per a set of transition rules.

A **Large Population Protocol** (LPP) considers the long-term behavior of an PP as its population approaches infinity. Thus, an LPP can be defined by the tuple (Q, I, S) where Q is the state space, I is the initial configuration, and S is the transitions or their associated differential system.

What does it mean to compute? A number is said to be computable by an LPP if a sum of state proportions converges to said number as the LPP's converges. Therefore, the model is restricted to computing numbers between zero and one.

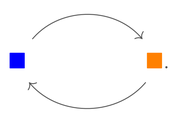
A Unimolecular Protocol



As a simplified introduction to LPPs, we provide this example in which an agent's state is either orange or blue. Although this protocol consists of solely unimolecular reactions, a PP may have any combination of unimolecular and bimolecular reactions. Let the protocol be described by:

- State Space: {■, ■},
- Initial Configuration: ■ = 100% and ■ = 0%,
- Transition Rules: {■ → ■, ■ → ■}.

Once selected for a reaction an agent will update its state to the complementary color. Thus, we have a very simple Standard Reaction Diagram:



We begin at $t = 0$ by initializing the agents' states. At each time step, one agent is randomly selected to update its state. With large populations, the proportion of blue agents will converge to 1/2 as time approaches infinity. Hence, this protocol computes 1/2.

Results

We have two important results today, each demonstrable through simulation with the PPsim python package.

Today's Results:

1. We extend the class of computable numbers to include transcendetals.
2. We define a multiplicative operator for LPPs based on the cross product.

In addition, we continue to progress in our goal of designing an algorithm that transforms any chemical reaction network into an LPP.

Computing Euler's e

Previous research has proposed individual protocols capable of computing whole classes of numbers, namely, a rational protocol and an algebraic protocol. Our research extends the computational power of LPPs to transcendental numbers.

We can compute a reciprocal of Euler's e using only 5 states and 2 transitions. The protocol design comes from encoding the equation,

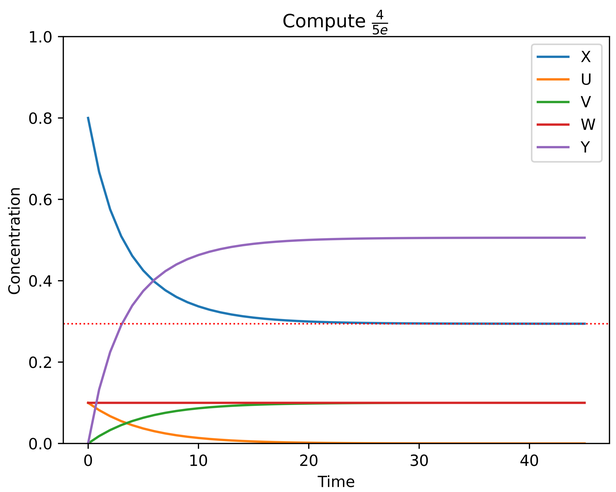
$$e^{-1} = \lim_{t\to\infty} e^{e^{-t}-1},$$

into a differential system such that the sum of derivatives is zero.

The protocol to compute $\frac{1}{ke}$ for all $k > 1$ uses the following transitions:

$$U + W \to V + W,$$
$$X + U \to Y + U.$$

By initializing the states (X, U, V, W, Y) to $(\frac{4}{5}, \frac{1}{10}, 0, \frac{1}{10}, 0)$, we compute $\frac{4}{5e}$.



Multiplication with LPPs

Given the protocols (Q_x, I_x, S_x) and (Q_y, I_y, S_y) that compute α and β respectively, we will define a new system (Q_{xy}, I_{xy}, S_{xy}) that computes the product $\gamma = \alpha\beta$.

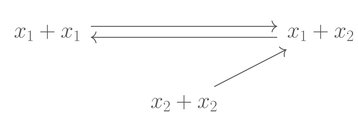
Steps :

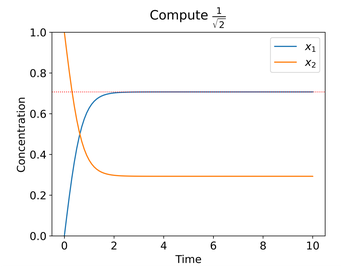
1. $Q_{xy} = \{xy \mid \forall (x, y) \in Q_x \times Q_y\}$
2. $I_{xy} = \{xy \mid \forall (x, y) \in I_x \times I_y\}$
3. S_{xy}: For each xy in Q_{xy}, we have $(xy)' = x'y + y'x$. Multiply the terms of $(xy)'$ by $1 = \sum Q_x = \sum Q_y$ to get a homogeneous degree 4 polynomial.
4. Transform the differential system in bimolecular transitions.

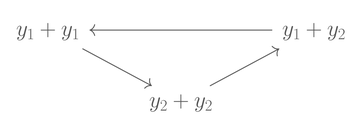
Computing a Product

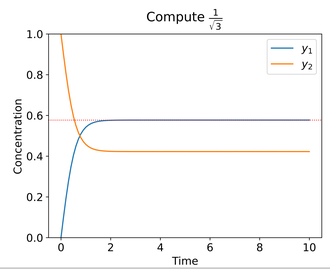
The Factors

Note that the two transition sets below compute $\frac{1}{\sqrt{2}}$ and $\frac{1}{\sqrt{3}}$ respectively.









The Product

Let $a = x_1y_1$, $b = x_1y_2$, $c = x_2y_1$, and $d = x_2y_2$. Appplying the above steps yield: