A simple population protocol for fast robust approximate majority

Dana Angluin[1] James Aspnes[1] David Eisenstat[2]

[1]Department of Computer Science
Yale University

[2]Department of Computer Science
Princeton University

DISC 2007, September 24th, 2007

Outline

1. Population protocols
   - Model
   - Previous work
   - Fast robust approximate majority
2. Analysis of fast robust approximate majority
   - Overview
   - State change bound
   - Correctness
   - Interaction bound
   - Byzantine resistance
3. One application and open problems
   - One application
   - Open problems

Definition

- A **population protocol** (Angluin, Aspnes, Diamadi, Fischer and Peralta, PODC 2004):
    - Q — a finite set of **states**
    - $\delta : Q \times Q \to Q \times Q$ — a joint **transition function**
    - ...
- **Agents** have states in Q
- An execution **step**:
    - Select an **initiator** and a **responder** at random
    - Update their states according to δ

$$(q'_i, q'_r) = \delta(q_i, q_r)$$

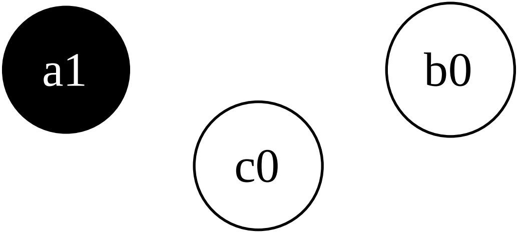
- n is the number of agents
- **Parallel time** is the number of steps per agent

Example: OR

- $Q = \{0, 1\}$
- The transition function $\delta(q_i, q_r) = (q_i, q_i \vee q_r)$:

$$10 \rightarrow 11$$

All other interactions have no effect.

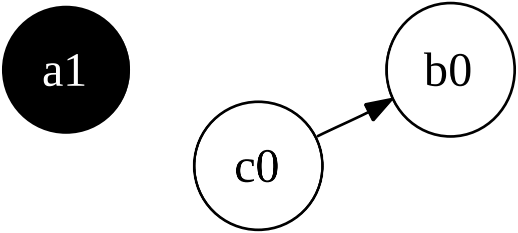


Example: OR

- $Q = \{0, 1\}$
- The transition function $\delta(q_i, q_r) = (q_i, q_i \vee q_r)$:

$$10 \rightarrow 11$$

All other interactions have no effect.

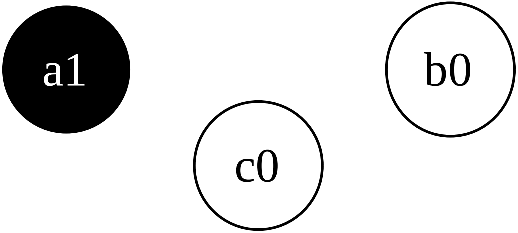


Example: OR

- $Q = \{0, 1\}$
- The transition function $\delta(q_i, q_r) = (q_i, q_i \vee q_r)$:

$$10 \rightarrow 11$$

All other interactions have no effect.

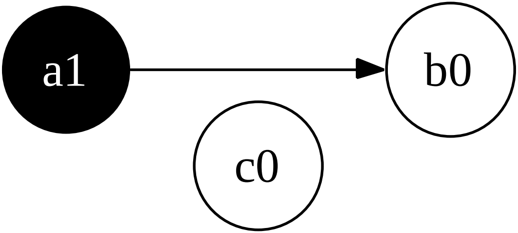


Example: OR

- $Q = \{0, 1\}$
- The transition function $\delta(q_i, q_r) = (q_i, q_i \vee q_r)$:

$$10 \rightarrow 11$$

All other interactions have no effect.

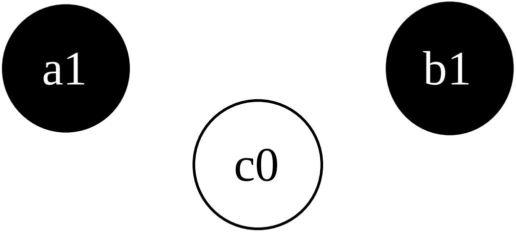


Example: OR

- $Q = \{0, 1\}$
- The transition function $\delta(q_i, q_r) = (q_i, q_i \vee q_r)$:

$$10 \rightarrow 11$$

All other interactions have no effect.

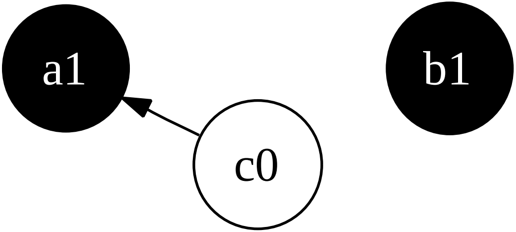


Example: OR

- $Q = \{0, 1\}$
- The transition function $\delta(q_i, q_r) = (q_i, q_i \vee q_r)$:

$$10 \to 11$$

All other interactions have no effect.

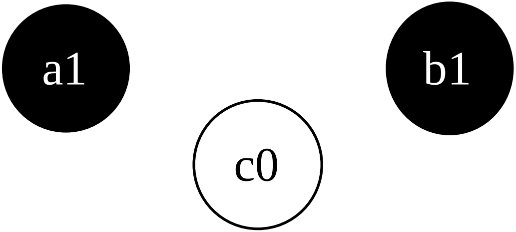


Example: OR

- $Q = \{0, 1\}$
- The transition function $\delta(q_i, q_r) = (q_i, q_i \vee q_r)$:

$$10 \rightarrow 11$$

All other interactions have no effect.

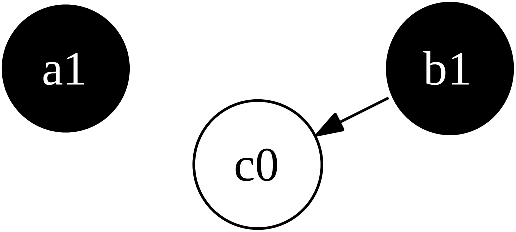


Example: OR

- $Q = \{0, 1\}$
- The transition function $\delta(q_i, q_r) = (q_i, q_i \vee q_r)$:

$$10 \to 11$$

All other interactions have no effect.

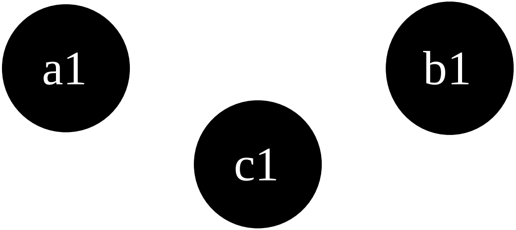


Example: OR

- $Q = \{0, 1\}$
- The transition function $\delta(q_i, q_r) = (q_i, q_i \vee q_r)$:

$$10 \rightarrow 11$$

All other interactions have no effect.



One motivation

- Chemical systems as distributed systems
  - What can they compute?
  - What can we say about their dynamics?
- Replace "agent" with "molecule" $\Rightarrow$ Chemical Master Equation (modulo details)
- Objection: in real life, some pairs of agents are more likely to interact than others
  - Agents in the same state are interchangeable
  - In a **well-stirred** chemical mixture, reaction *types* occur with the right probabilities (Gillespie, Physica A 1992)

Majority framework

- Agents start in one of two states, x or y
- The population must eventually agree on the majority value with probability 1
  - Assume that there is no tie
  - A map $o : Q \to \{x, y\}$ extracts output values from states
  - Termination is not required
- These conditions may be relaxed
  - Add a leader
  - Allow error with probability $n^{-\Theta(1)}$
  - Allow error when there are about as many x's as y's

Previous work

The original population protocol for majority (Angluin et al., PODC 2004)

- Works by canceling x's and y's and electing a leader to dictate the result
- Runs in (expected) parallel time $O(n \log n)$
- Adapted to handle stabilizing inputs (Angluin, Aspnes, Chan, Fischer, Jiang and Peralta, DCOSS 2005)
- Adapted to use one-way communication with queuing (Angluin, Aspnes, E. and Ruppert, OPODIS 2005)
- Adapted to handle $O(1)$ crash failures (Delporte-Gallet, Fauconnier, Guerraoui and Ruppert, DCOSS 2006)
- Modified to run in parallel time $O(n)$ (Angluin, Aspnes and E., DISC 2006)

Previous work (continued)

Majority on a simulated register machine (Angluin et al., DISC 2006)

- Requires a leader
- Uses the timing properties of epidemics to achieve partial synchrony with high probability
  - The **phase-clock** construction
- Works by alternating rounds of
  - Canceling x's and y's partially
  - Doubling the numbers of each

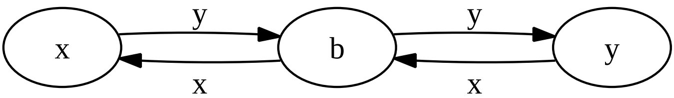
  until only the majority value remains
- Runs in parallel time $O(\log^2 n)$: $O(\log n)$ rounds, each of which takes $O(\log n)$ parallel time
- Fails with probability $n^{-\Theta(1)}$ unless the previous algorithm is used as a fail-safe

Fast robust approximate majority

- $Q = \{x, y, b\}$
- b is the **blank** state
- The transition function δ:

$$\begin{array}{ll} xy \rightarrow xb & xb \rightarrow xx \\ yx \rightarrow yb & yb \rightarrow yy \end{array}$$

  All other interactions have no effect.
- Why not $bx \rightarrow xx$ or $xy \rightarrow bb$? Requires two-way communication, can lose the last non-blank
- The transition graph of the responder:



Intuitions behind fast robust approximate majority

- Multiplicative increase, additive decrease
  - x's and y's recruit b's in proportion to their numbers BUT
  - An xy interaction is as likely as a yx
  - Small initial gap widens to total domination
- Analogy to the register machine algorithm
  - xy and yx interactions are like the canceling rounds
  - xb and yb interactions are like the doubling rounds
  - Faster because we don't wait $O(\log n)$ parallel time for the last agent to double
  - Approximate because the rate of each process is random
- Next: proof sketch

Overview

- Using martingales, we show that with high probability,
  - The number of state changes before converging is $O(n \log n)$
  - The total number of interactions before converging is $O(n \log n)$
  - The final outcome is correct if the initial disparity is $\omega(\sqrt{n \log n})$
- This algorithm is the fastest possible
  - Must wait $\Omega(n \log n)$ steps in expectation for all agents to interact
- Finally, we consider the effect of Byzantine agents

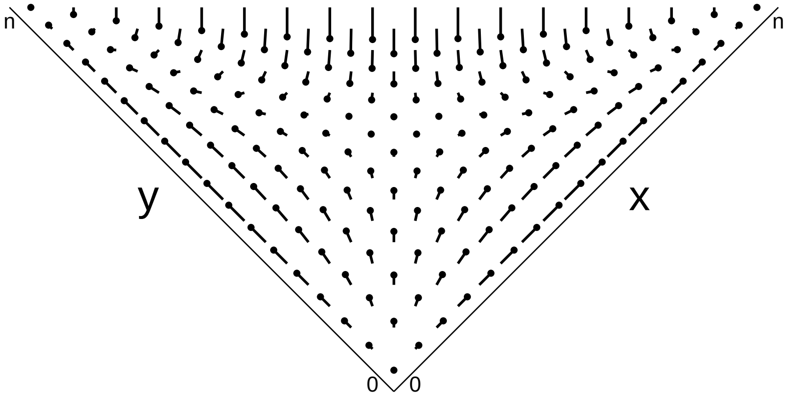
Bounding the number of state changes (1)

- Define

x	the number of x's
y	the number of y's
b	the number of b's
u	$x - y$
S^{vb}	the number of xb and yb interactions so far
S^{xy}	the number of xy and yx interactions so far

- Claim: $|S^{vb} - S^{xy}| \leq n - 1$
  - The left-hand side is how much the number of non-blank agents has changed

Configuration space



$|u| = |x - y|$ is a pretty good measure of progress

Bounding the number of state changes (2)

- Given an xy or yx interaction:
  - u increases by 1 with probability $1/2$
  - u decreases by 1 with probability $1/2$

  Like a random walk
- Given an xb or yb interaction:
  - u increases by 1 with probability $x/(x+y)$
  - u decreases by 1 with probability $y/(x+y)$

  The expected increase is roughly proportional to u: like exponential growth

Bounding the number of state changes (3)

- f is the potential function
- We design f to increase on xb and yb interactions and decrease less on xy and yx interactions
- For $|u|$ small, f should resemble the potential function for a random walk, u^2
- For $|u|$ large, f should resemble the potential function for exponential growth, $\log |u|$

$$f(u) = \log\left(\frac{3}{2}n + u^2\right)$$

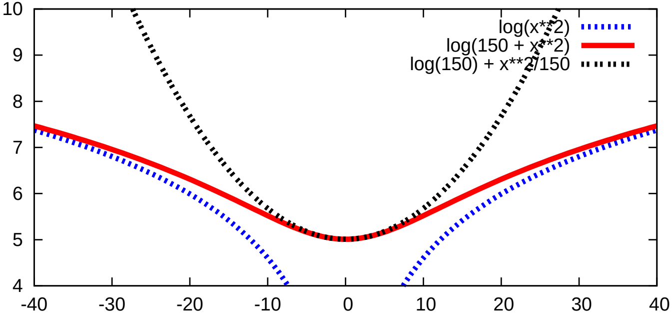
Bounding the number of state changes (3)

- f is the potential function
- We design f to increase on xb and yb interactions and decrease less on xy and yx interactions
- For $|u|$ small, f should resemble the potential function for a random walk, u^2
- For $|u|$ large, f should resemble the potential function for exponential growth, $\log |u|$

$$f(u) = \log\left(\frac{3}{2}n + u^2\right)$$

Bounding the number of state changes (4)

- f and the two functions it behaves like:



- f increases by $\sim 2/(3n)$ conditioned on xb or yb
- f decreases by $\sim 1/(12n)$ conditioned on xy or yx
- Martingales: $E[\Delta f]$ is $\Omega(n^{-1/2})$, so f attains its maximum in $\Theta(\log n)/\Omega(1/n) = O(n \log n)$ steps whp

Correctness of fast robust approximate majority

Fast robust approximate majority is correct given that the initial margin is $\omega(\sqrt{n \log n})$

- Couple (u_i) with an unbiased random walk (t_i) so that $|t_i| \leq |u_i|$
  - $\Pr[u \text{ increases}] \geq 1/2$ for $u \geq 0$
  - $\Pr[u \text{ decreases}] \geq 1/2$ for $u \leq 0$
- Suppose $t_0 = u_0 = x_0 - y_0 = \omega(\sqrt{n \log n})$
- Whp, random walk is positive for $\Theta(n \log n)$ steps $\Rightarrow$ x wins
- Argue symmetrically wrt y

Bounding the total number of interactions

- Why doesn't the bound on state changes suffice?
  - State changes are infrequent in the corners
- Solution: introduce auxiliary potential functions

b corner	$\log(x+y)$
x corner	$-\log(1+b+3y)$
y corner	$-\log(1+b+3x)$

- In their respective corners, these functions increase by $\Omega(1/n)$ in expectation
- The decrease elsewhere is bounded by the number of state changes $\Rightarrow$ the desired bound

Byzantine resistance

- Suppose there are $z = o(\sqrt{n})$ Byzantine agents
  - Can change their state at will
  - Cannot control the scheduling of interactions
  - No information about the future
- Weaker guarantees wrt convergence and correctness
- Our proof of convergence requires
  - $\sqrt{n}$ non-blank agents to start with
  - Redefining convergence to be when at most $O(\sqrt{n})$ agents have the wrong value
  - Truncating the execution after exponentially many steps (instead of never)
  - Allowing probability $O(n^{-c})$ of failure
- Correctness requires a slightly larger margin of $\omega(\sqrt{n}\log n)$

Proof sketch of Byzantine resistance

- Maximum "error" in potential function analysis is $o(1/n)$: not enough to cause trouble in the center
  - Byzantine interaction probability is $\frac{o(\sqrt{n}) \cdot O(n)}{n(n-1)} = o(1/\sqrt{n})$
  - Maximum potential function change is $O(1/\sqrt{n})$
- Strong pressure out of the b corner
- Strong pressure into the x and y corners
  - In both cases, Byzantine agents "winning" involves completing biased random walks in reverse $\Rightarrow$ not for exponentially many steps
  - The Byzantine agents are not numerous enough to keep the protocol in the center for long

- Make fast comparison exact whp
  - Make unary representation robust by using multiples of $\Theta(n^{2/3})$
  - Add "1/2" to avoid non-deterministic behavior for comparing equal quantities
- Together with other tricks, reduce amortized per-step overhead of
  - addition
  - subtraction
  - comparison
  - division by a constant

  to $O(\log n)$ parallel time per step—improved by several log factors

- Better proofs for fast robust approximate majority
- Obstacles:
    - Does not resemble a well-studied random process (coupon collector, random walk) throughout the configuration space
    - No closed-form solution to the analogous differential equations
- Any proof at all for several protocols described in the paper (have only empirical evidence)
    - Three or more values $\Rightarrow$ "Fast robust approximate plurality"
    - Phase-clock that stabilizes in $O(\log n)$ parallel time
    - Leader election in $O(\log n)$ parallel time

Thank you!