Characterizing consensus in the Heard-Of model

Igor Walukiewicz

joint work with A.R. Balasubramanian

Consesus problem:

Every process starts with the same value of its input variable. We require:

- termination: every process eventually sets its dec value,
- agreement: all dec variables have the same value,
- every dec variable is set at most once,
- the value of dec variables is one of the initial values of input variables.

FLP theorem: consensus is impossible in fully asynchronous systems in the presence of faults

2/3 algorithm

the multiset of values received from other processes

send (inp)

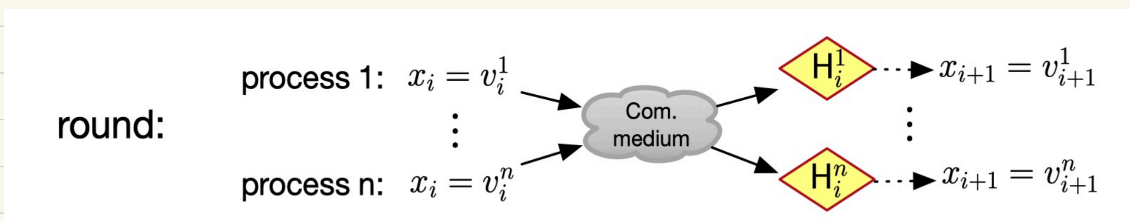
- **if** $\texttt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > \frac{2}{3}n$ **then** $x_1 := inp := \mathrm{smor}(\mathsf{H})$;
- **if** $\texttt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > \frac{2}{3}n$ **then** $x_1 := inp := \mathrm{smor}(\mathsf{H})$;

smallest most frequent value in H

send x_1

- **if** $\texttt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > \frac{2}{3}n$ **then** $dec := \mathrm{smor}(\mathsf{H})$;

Communication predicate: eventually $\psi^1 = (\varphi_= \wedge \varphi_{\frac{2}{3}}, true)$ and later $\psi^2 = (\varphi_{\frac{2}{3}}, \varphi_{\frac{2}{3}})$



2/3 algorithm

the multiset of values received from other processes

---

send (inp)

  if uni(H) $\wedge$ $|\mathsf{H}| > \frac{2}{3}n$ **then** $x_1 := inp :=$ smor(H);

  if mult(H) $\wedge$ $|\mathsf{H}| > \frac{2}{3}n$ **then** $x_1 := inp :=$ smor(H);

smallest most frequent value in H

send x_1

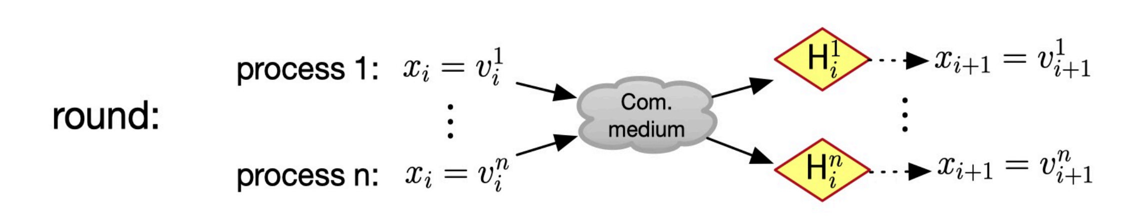
  if uni(H) $\wedge$ $|\mathsf{H}| > \frac{2}{3}n$ **then** $dec :=$ smor(H);

Communication predicate: eventually $\psi^1 = (\varphi_= \wedge \varphi_{\frac{2}{3}}, true)$ and later $\psi^2 = (\varphi_{\frac{2}{3}}, \varphi_{\frac{2}{3}})$

---

every process receives the same multiset

H contains values from $\geq 2/3$ of processes



2/3 algorithm

the multiset of values received from other processes

send (inp)

  if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > \frac{2}{3}n$ **then** $x_1 := inp := \mathrm{smor}(\mathsf{H})$;

  if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > \frac{2}{3}n$ **then** $x_1 := inp := \mathrm{smor}(\mathsf{H})$;

smallest most frequent value in H

send x_1

  if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > \frac{2}{3}n$ **then** $dec := \mathrm{smor}(\mathsf{H})$;

Communication predicate: eventually $\psi^1 = (\varphi_= \wedge \varphi_{\frac{2}{3}}, true)$ and later $\psi^2 = (\varphi_{\frac{2}{3}}, \varphi_{\frac{2}{3}})$

every process receives the same multiset

H contains values from $\geq 2/3$ of processes

- At $(\varphi_= \wedge \varphi_{2/3}, true)$ phase, every process sets inp to the same value
- At $(\varphi_{2/3}, \varphi_{2/3})$ phase later, every process sets dec to this value

Q : can we change 2/3 to 1/2 ?

2/3 algorithm

the multiset of values received from other processes

send (inp)

if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > \frac{2}{3}n$ **then** $x_1 := inp := \mathrm{smor}(\mathsf{H})$;
if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > \frac{2}{3}n$ **then** $x_1 := inp := \mathrm{smor}(\mathsf{H})$;

smallest most frequent value in H

send x_1

if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > \frac{2}{3}n$ **then** $dec := \mathrm{smor}(\mathsf{H})$;

Communication predicate: eventually $\psi^1 = (\varphi_= \wedge \varphi_{\frac{2}{3}}, true)$ and later $\psi^2 = (\varphi_{\frac{2}{3}}, \varphi_{\frac{2}{3}})$

every process receives the same multiset

H contains values from $\geq 2/3$ of processes

Q : can we change 2/3 to 1/2 ?

$f = \{\overbrace{a, \ldots, a}^{1/2}, \overbrace{b, \ldots, b}^{1/2}\}$ H_a - more a's H_b - more b's

$f' = \{\overbrace{a, \ldots, a}^{1/2+\varepsilon}, \overbrace{b, \ldots, b}^{1/2-\varepsilon}\}$ send $H = \{\overbrace{a, \ldots, a}^{1/2+\varepsilon}\}$ to p_1 and nothing to others. So p_1 dec on a.

In the next phase send $H_{bb} = \{\overbrace{a, \ldots, a}^{\varepsilon}, \overbrace{b, \ldots, b}^{1/2-\varepsilon}\}$ to every process. They decide on b.

Heard-off model

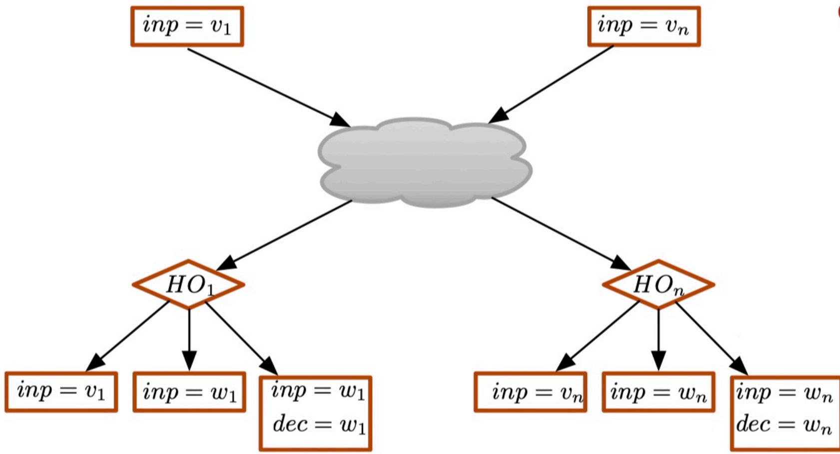
Introduced by Bernadette Charron-Bost · André Schiper in 2009

A round based model for non synchronous computing.

Unified treatment of different types of faults through transmission faults.

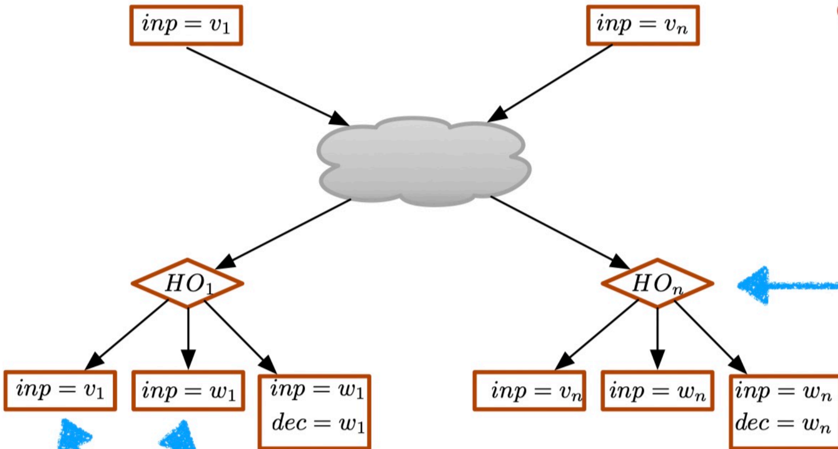
A model is relatively simple and concise:
a good candidate to develop verification methods

- [Charron-Bost, Stefan Merz,..] Efficient encoding the model in Isabelle, and TLA
- [Drăgoi, Henzinger, Zufferey,..] A semi-automatic proof method, a domain-specific language based on HO-model.
- [Ognjen Maric, Christoph Sprenger, David Basin, *Cut-off Bounds for Consensus Algorithms*], see later
- [R. Bloem, S. Jacobs, A. Khalimov, I. Konnov, S. Rubin, H. Veith, and J. Widder. *Decidability of Parameterized Verification*], a book, 2015



- No operations on variables
- No failure of components
- No process identities

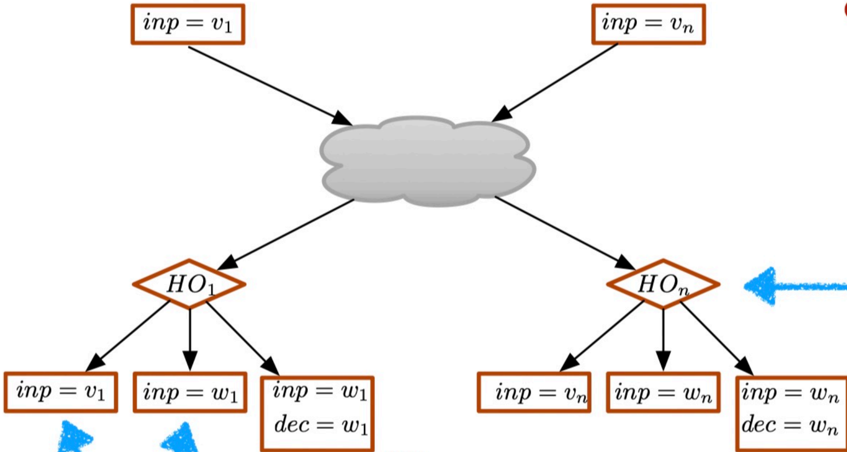
- No operations on variables
- No failure of components
- No process identities



HO_i is a multiset of values

Value of ***inp*** either stays the same or changes to some received value

- No operations on variables
- No failure of components
- No process identities

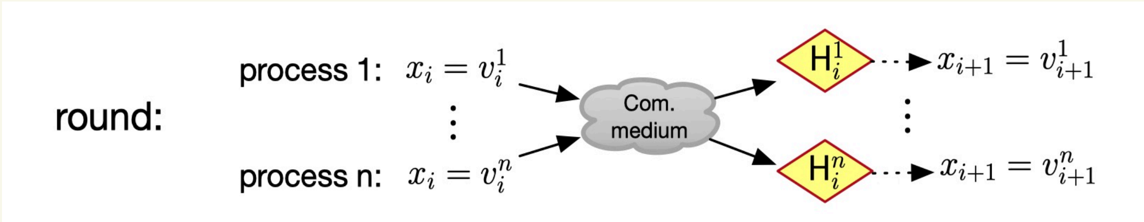


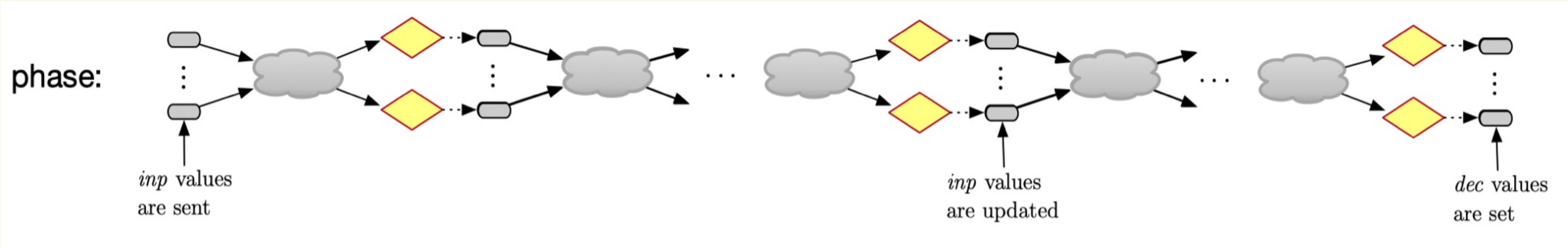
HO_i is a multiset of values

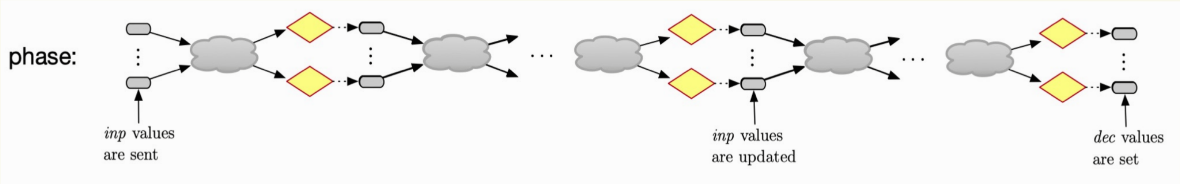
Value of ***inp*** either stays the same or changes to some received value

A process can also definitely decide on some value by setting ***dec***

Heard-of algorithm







send inp
  if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > \mathit{thr}_u^1 \cdot n$ then $x_1 := \mathsf{op}_0^1(\mathsf{H})$;
  $\vdots$
  if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > \mathit{thr}_m^{1,k} \cdot n$ then $x_1 := \mathsf{op}_k^1(\mathsf{H})$;

input sent, x_1 set

send x_{i-1}
  if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > \mathit{thr}_u^i \cdot n$ then $x_i := \mathsf{op}_0^i(\mathsf{H})$;
  $\vdots$
  if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > \mathit{thr}_m^{i,k} \cdot n$ then $x_i := \mathsf{op}_k^i(\mathsf{H})$;

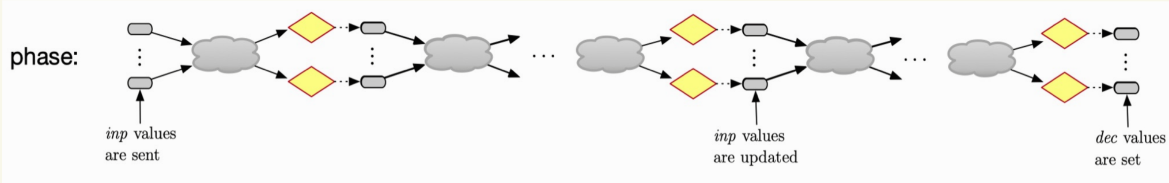
x_{i-1} sent, x_i set

send $x_{\mathbf{ir}-1}$
  if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > \mathit{thr}_u^{\mathbf{ir}} \cdot n$ then $x_{\mathbf{ir}} := \mathit{inp} := \mathsf{op}_0^{\mathbf{ir}}(\mathsf{H})$;
  $\vdots$
  if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > \mathit{thr}_m^{\mathbf{ir},k} \cdot n$ then $x_{\mathbf{ir}} := \mathit{inp} := \mathsf{op}_k^{\mathbf{ir}}(\mathsf{H})$;

round ri where inp is set [update]

send x_{l-1}
  if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > \mathit{thr}_u^l \cdot n$ then $\mathit{dec} := \mathsf{op}_0^l(\mathsf{H})$;
  $\vdots$
  if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > \mathit{thr}_m^{l,k} \cdot n$ then $\mathit{dec} := \mathsf{op}_k^l(\mathsf{H})$;

last round where dec is set [decision]



send inp

  if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^1 \cdot n$ **then** $x_1 := \mathsf{op}_0^1(\mathsf{H})$;

  $\vdots$

  if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{1,k} \cdot n$ **then** $x_1 := \mathsf{op}_k^1(\mathsf{H})$;

send x_{i-1}

  if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^i \cdot n$ **then** $x_i := \mathsf{op}_0^i(\mathsf{H})$;

  $\vdots$

  if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{i,k} \cdot n$ **then** $x_i := \mathsf{op}_k^i(\mathsf{H})$;

send $x_{\mathbf{ir}-1}$

  if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^{\mathbf{ir}} \cdot n$ **then** $x_{\mathbf{ir}} := inp := \mathsf{op}_0^{\mathbf{ir}}(\mathsf{H})$;

  $\vdots$

  if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{\mathbf{ir},k} \cdot n$ **then** $x_{\mathbf{ir}} := inp := \mathsf{op}_k^{\mathbf{ir}}(\mathsf{H})$;

send x_{l-1}

  if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^l \cdot n$ **then** $dec := \mathsf{op}_0^l(\mathsf{H})$;

  $\vdots$

  if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{l,k} \cdot n$ **then** $dec := \mathsf{op}_k^l(\mathsf{H})$;

---

send (inp)

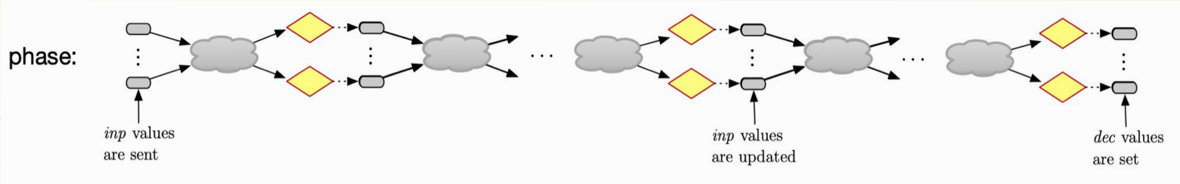
  if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > \frac{2}{3}n$ **then** $x_1 := inp := \mathrm{smor}(\mathsf{H})$;

  if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > \frac{2}{3}n$ **then** $x_1 := inp := \mathrm{smor}(\mathsf{H})$;

send x_1

  if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > \frac{2}{3}n$ **then** $dec := \mathrm{smor}(\mathsf{H})$;

Communication predicate: eventually $\psi^1 = (\varphi_= \wedge \varphi_{\frac{2}{3}}, true)$ and later $\psi^2 = (\varphi_{\frac{2}{3}}, \varphi_{\frac{2}{3}})$



send *inp*

  if uni(H) $\wedge$ $|\mathsf{H}| > thr_u^1 \cdot n$ **then** $x_1 := \mathsf{op}_0^1(\mathsf{H})$;

  $\vdots$

  if mult(H) $\wedge$ $|\mathsf{H}| > thr_m^{1,k} \cdot n$ **then** $x_1 := \mathsf{op}_k^1(\mathsf{H})$;

send x_{i-1}

  if uni(H) $\wedge$ $|\mathsf{H}| > thr_u^i \cdot n$ **then** $x_i := \mathsf{op}_0^i(\mathsf{H})$;

  $\vdots$

  if mult(H) $\wedge$ $|\mathsf{H}| > thr_m^{i,k} \cdot n$ **then** $x_i := \mathsf{op}_k^i(\mathsf{H})$;

send $x_{\mathbf{ir}-1}$

  if uni(H) $\wedge$ $|\mathsf{H}| > thr_u^{\mathbf{ir}} \cdot n$ **then** $x_{\mathbf{ir}} := inp := \mathsf{op}_0^{\mathbf{ir}}(\mathsf{H})$;

  $\vdots$

  if mult(H) $\wedge$ $|\mathsf{H}| > thr_m^{\mathbf{ir},k} \cdot n$ **then** $x_{\mathbf{ir}} := inp := \mathsf{op}_k^{\mathbf{ir}}(\mathsf{H})$;

send x_{l-1}

  if uni(H) $\wedge$ $|\mathsf{H}| > thr_u^l \cdot n$ **then** $dec := \mathsf{op}_0^l(\mathsf{H})$;

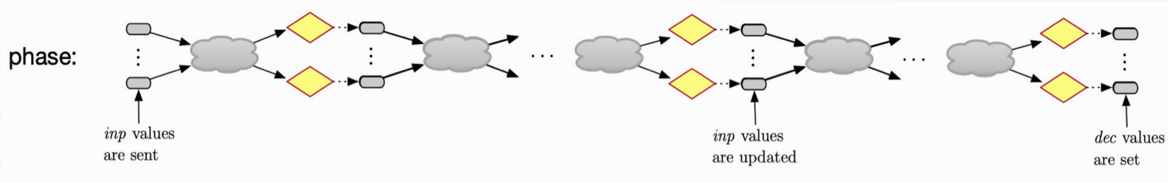
  $\vdots$

  if mult(H) $\wedge$ $|\mathsf{H}| > thr_m^{l,k} \cdot n$ **then** $dec := \mathsf{op}_k^l(\mathsf{H})$;

Wanted:

- Given an algorithm over a fixed set of values, decide if it solves the consensus

- What operations and tests are allowed?
- Do we have a cut-of principle? restriction to k-proc
- Do we have a 0/1 principle? restriction to {0,1} values



send *inp*

$\quad$ if $\texttt{uni}(\mathsf{H}) \land |\mathsf{H}| > thr_u^1 \cdot n$ then $x_1 := \mathsf{op}_0^1(\mathsf{H})$;

$\quad \vdots$

$\quad$ if $\texttt{mult}(\mathsf{H}) \land |\mathsf{H}| > thr_m^{1,k} \cdot n$ then $x_1 := \mathsf{op}_k^1(\mathsf{H})$;

Semantics: phase round

$$(f, d) \xrightarrow{\psi} (f', d') \quad \text{and} \quad f \stackrel{\varphi}{\Longrightarrow}_i f' .$$

$f : \{1, .., n\} \to D \cup \{?\}$ $\quad$ $d : \{1, .., n\} \to D \cup \{?\}$

D finite and linearly ordered

send x_{i-1}

$\quad$ if $\texttt{uni}(\mathsf{H}) \land |\mathsf{H}| > thr_u^i \cdot n$ then $x_i := \mathsf{op}_0^i(\mathsf{H})$;

$\quad \vdots$

$\quad$ if $\texttt{mult}(\mathsf{H}) \land |\mathsf{H}| > thr_m^{i,k} \cdot n$ then $x_i := \mathsf{op}_k^i(\mathsf{H})$;

send $x_{\mathbf{ir}-1}$

$\quad$ if $\texttt{uni}(\mathsf{H}) \land |\mathsf{H}| > thr_u^{\mathbf{ir}} \cdot n$ then $x_{\mathbf{ir}} := inp := \mathsf{op}_0^{\mathbf{ir}}(\mathsf{H})$;

$\quad \vdots$

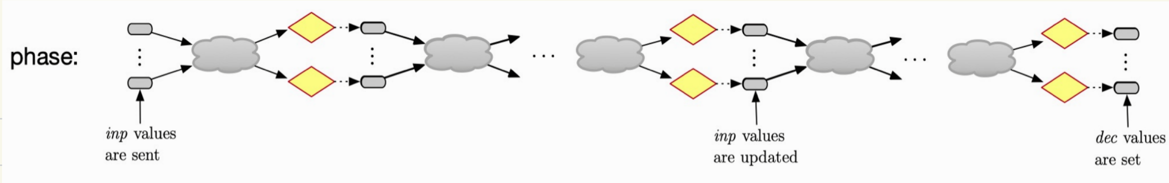
$\quad$ if $\texttt{mult}(\mathsf{H}) \land |\mathsf{H}| > thr_m^{\mathbf{ir},k} \cdot n$ then $x_{\mathbf{ir}} := inp := \mathsf{op}_k^{\mathbf{ir}}(\mathsf{H})$;

send x_{l-1}

$\quad$ if $\texttt{uni}(\mathsf{H}) \land |\mathsf{H}| > thr_u^l \cdot n$ then $dec := \mathsf{op}_0^l(\mathsf{H})$;

$\quad \vdots$

$\quad$ if $\texttt{mult}(\mathsf{H}) \land |\mathsf{H}| > thr_m^{l,k} \cdot n$ then $dec := \mathsf{op}_k^l(\mathsf{H})$;



send *inp*
  if uni(H) $\wedge$ |H| $>$ $thr^1_u \cdot n$ **then** $x_1 := \mathsf{op}^1_0(\mathsf{H})$;
  $\vdots$
  if mult(H) $\wedge$ |H| $>$ $thr^{1,k}_m \cdot n$ **then** $x_1 := \mathsf{op}^1_k(\mathsf{H})$;

send x_{i-1}
  if uni(H) $\wedge$ |H| $>$ $thr^i_u \cdot n$ **then** $x_i := \mathsf{op}^i_0(\mathsf{H})$;
  $\vdots$
  if mult(H) $\wedge$ |H| $>$ $thr^{i,k}_m \cdot n$ **then** $x_i := \mathsf{op}^i_k(\mathsf{H})$;

send $x_{\mathbf{ir}-1}$
  if uni(H) $\wedge$ |H| $>$ $thr^{\mathbf{ir}}_u \cdot n$ **then** $x_{\mathbf{ir}} := inp := \mathsf{op}^{\mathbf{ir}}_0(\mathsf{H})$;
  $\vdots$
  if mult(H) $\wedge$ |H| $>$ $thr^{\mathbf{ir},k}_m \cdot n$ **then** $x_{\mathbf{ir}} := inp := \mathsf{op}^{\mathbf{ir}}_k(\mathsf{H})$;

send x_{l-1}
  if uni(H) $\wedge$ |H| $>$ $thr^l_u \cdot n$ **then** $dec := \mathsf{op}^l_0(\mathsf{H})$;
  $\vdots$
  if mult(H) $\wedge$ |H| $>$ $thr^{l,k}_m \cdot n$ **then** $dec := \mathsf{op}^l_k(\mathsf{H})$;

Semantics:

phase   round

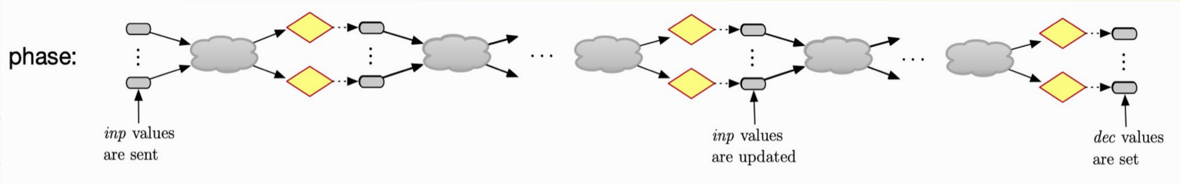
$(f,d) \xrightarrow{\psi} (f',d')$ and $f \xRightarrow{\varphi}_i f'$.

- $f : \{1,..n\} \to D \cup \{?\}$ $d : \{1,..n\} \to D \cup \{?\}$
  D finite and linearly ordered

- $f \xRightarrow{\varphi}_i f'$ if $\exists\, (H_1,..,H_n) \models \varphi \wedge \forall p\ \ H_p \subseteq mset(f)$
  $f'(p) = update_i(H_p)$

  min(H) min value in $H - \{?\}$
  smor(H) smallest most frequent value in $H - \{?\}$
  $f'(p) = ?$ if no test holds for H_p

Rem: tests count ? value but operations ignore it



send inp
- if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^1 \cdot n$ then $x_1 := \mathtt{op}_0^1(\mathsf{H})$;
- $\vdots$
- if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{1,k} \cdot n$ then $x_1 := \mathtt{op}_k^1(\mathsf{H})$;

send x_{i-1}
- if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^i \cdot n$ then $x_i := \mathtt{op}_0^i(\mathsf{H})$;
- $\vdots$
- if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{i,k} \cdot n$ then $x_i := \mathtt{op}_k^i(\mathsf{H})$;

send $x_{\mathbf{ir}-1}$
- if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^{\mathbf{ir}} \cdot n$ then $x_{\mathbf{ir}} := inp := \mathtt{op}_0^{\mathbf{ir}}(\mathsf{H})$;
- $\vdots$
- if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{\mathbf{ir},k} \cdot n$ then $x_{\mathbf{ir}} := inp := \mathtt{op}_k^{\mathbf{ir}}(\mathsf{H})$;

send x_{l-1}
- if $\mathtt{uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^l \cdot n$ then $dec := \mathtt{op}_0^l(\mathsf{H})$;
- $\vdots$
- if $\mathtt{mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{l,k} \cdot n$ then $dec := \mathtt{op}_k^l(\mathsf{H})$;

Semantics:

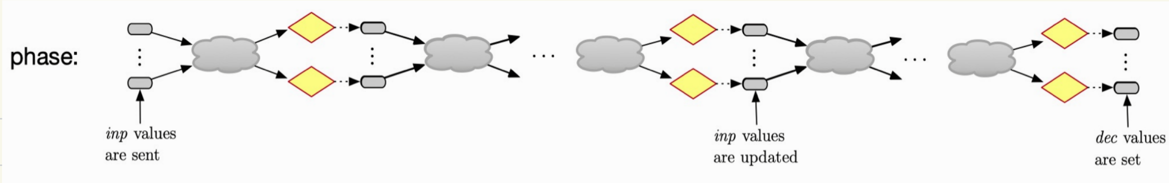
phase $\qquad\qquad$ round

$(f,d) \xrightarrow{\psi} (f',d')$ and $f \xRightarrow{\varphi}_i f'$.

- $f : \{1,..,n\} \to D \cup \{?\}$ $\qquad d : \{1,..,n\} \to D \cup \{?\}$
  D finite and linearly ordered

- $f \xRightarrow{\varphi}_i f'$ if $\exists (H_1, \dots, H_n) \models \varphi \wedge \forall p$ $H_p \subseteq mset(f)$
  $f'(p) = update_i(H_p)$
  min(H) min value in $H - \{?\}$
  smor(H) smallest most frequent value in $H - \{?\}$
  $f'(p) = ?$ if no test holds for H_p

- $(f,d) \xrightarrow{\psi} (f',d')$ if $f_0 \xRightarrow{\varphi_1} f_1 \xRightarrow{\varphi_2} \dots \xRightarrow{\varphi_l} f_l$
  - $f_0 = f$
  - $f'(p) = f_{ir}(p)$ if $f_{ir}(p) \neq ?$ and $f'(p) = f(p)$ otherwise
  - $d'(p) = d(p)$ if $d(p) \neq ?$ and $d'(p) = f_l(p)$ otherwise



send inp

  if $\mathsf{uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^1 \cdot n$ **then** $x_1 := \mathsf{op}_0^1(\mathsf{H})$;

  $\vdots$

  if $\mathsf{mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{1,k} \cdot n$ **then** $x_1 := \mathsf{op}_k^1(\mathsf{H})$;

send x_{i-1}

  if $\mathsf{uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^i \cdot n$ **then** $x_i := \mathsf{op}_0^i(\mathsf{H})$;

  $\vdots$

  if $\mathsf{mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{i,k} \cdot n$ **then** $x_i := \mathsf{op}_k^i(\mathsf{H})$;

send $x_{\mathbf{ir}-1}$

  if $\mathsf{uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^{\mathbf{ir}} \cdot n$ **then** $x_{\mathbf{ir}} := inp := \mathsf{op}_0^{\mathbf{ir}}(\mathsf{H})$;

  $\vdots$

  if $\mathsf{mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{\mathbf{ir},k} \cdot n$ **then** $x_{\mathbf{ir}} := inp := \mathsf{op}_k^{\mathbf{ir}}(\mathsf{H})$;

send x_{l-1}

  if $\mathsf{uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^l \cdot n$ **then** $dec := \mathsf{op}_0^l(\mathsf{H})$;

  $\vdots$

  if $\mathsf{mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{l,k} \cdot n$ **then** $dec := \mathsf{op}_k^l(\mathsf{H})$;

Semantics: phase $(f,d) \xrightarrow{\psi} (f',d')$ and round $f \xRightarrow{\varphi}_i f'$.

$f : \{1,\dots,n\} \to D \cup \{?\}$ $d : \{1,\dots,n\} \to D \cup \{?\}$

D finite and linearly ordered

$f \xRightarrow{\varphi}_i f'$ if $\exists (H_1,\dots,H_n) \models \varphi \wedge \forall p$ $H_p \subseteq mset(f)$
$f'(p) = update_i(H_p)$

Communication predicate $(\mathsf{G}\overline{\psi}) \wedge (\mathsf{F}(\psi_1 \wedge \mathsf{F}(\psi_2 \wedge \dots (\mathsf{F}\psi_k)\dots)))$

phase predicates

$\psi = (\varphi_1,\dots,\varphi_\ell)$

round predicates

φ_i is $(|H| > thr \cdot n)$ or $(eq \wedge |H| > thr \cdot n)$

A Characterization

▶ **Theorem 22.** *An algorithm solves consensus iff it is:*

- *syntactically safe,*
- *there are $i \leq j$ with ψ_i a unifier and ψ_j a decider.*

$(\mathsf{G}\overline{\psi}) \wedge (\mathsf{F}(\psi_1 \wedge \mathsf{F}(\psi_2 \wedge \ldots (\mathsf{F}\psi_k)\ldots)))$

We fix $D = \{a, b\}$ (we show 0/1-principle with our proof)

Some notation for $f : \{1,..,n\} \to D \cup \{?\}$ $\quad bias(\theta) = (a,..,a,\underbrace{b,..,b}_{\theta \cdot n})$ $\quad bias^?(\theta) = (?,..,?,\underbrace{b,..,b}_{\theta \cdot n})$

$solo = (b,..,b)$ $\quad solo^? = (?,..,?)$

A round i is <u>solo safe</u> wrt. ψ if $thr_u^i \leq thr_i(\psi)$ $\quad (solo \xRightarrow{\psi}_i solo)$

A round i is <u>preserving</u> wrt. ψ if either:

- no uni instruction, or
- no mult instruction, or
- $thr_i(\psi) < \max(thr_u^i, thr_m^{i,u})$

<u>Rem</u> $bias(\theta) \xRightarrow{\psi}_i solo^?$ is possible

A Characterization

▶ **Theorem 22.** *An algorithm solves consensus iff it is:*
- *syntactically safe,*
- *there are $i \le j$ with ψ_i a unifier and ψ_j a decider.*

$(\mathsf{G}\overline{\psi}) \wedge (\mathsf{F}(\psi_1 \wedge \mathsf{F}(\psi_2 \wedge \ldots (\mathsf{F}\psi_k)\ldots)))$

A round i is <u>solo safe</u> wrt. ψ if $thr_u^i \le thr_i(\psi)$

A round i is <u>preserving</u> wrt. ψ if either:
- no uni instruction, or
- no mult instruction, or
- $thr_i(\psi) < \max(thr_u^i, thr_m^{i,k})$

<u>Rem</u> $bias(\theta) \overset{\psi}{\Longrightarrow}_i$ solo? is possible

ψ is a <u>decider</u> if all rounds are solo safe wrt. ψ

ψ is a <u>unifier</u> if · $\exists$ round i s.t. $\underbrace{\boxed{1}\ ..\ \boxed{i-1}}_{\text{non-preserving}}\ \overset{\text{equalizer}}{\boxed{i}}\ \underbrace{\boxed{i+1}\ ..\ \boxed{\ell}}_{\text{solo safe}}$

· $thr_1(\psi) \ge thr_m^{1,k}$ and either $thr_1(\psi) \ge thr_u^1$ or $thr_1(\psi) \ge \overline{thr}$

where $\overline{thr} = \max(1 - thr_u^1, 1 - thr_m^{1,k}/2)$

A Characterization

▶ **Theorem 22.** *An algorithm solves consensus iff it is:*

- *syntactically safe,*
- *there are $i \leq j$ with ψ_i a unifier and ψ_j a decider.*

$\texttt{send}\ inp$
$\quad \texttt{if uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^1 \cdot n\ \texttt{then}\ x_1 := \mathsf{op}_0^1(\mathsf{H});$
$\quad \vdots$
$\quad \texttt{if mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{1,k} \cdot n\ \texttt{then}\ x_1 := \mathsf{op}_k^1(\mathsf{H});$

▶ **Definition 9.** *An algorithm is* syntactically safe *when:*

1. *First round has a* `mult` *instruction.*
2. *Every round has a* `uni` *instruction.*
3. *In the first round the operation in every* `mult` *instruction is* smor.
4. $thr_m^{1,k}/2 \geq 1 - thr_u^{\mathbf{ir}+1}$, *and* $thr_u^1 \geq 1 - thr_u^{\mathbf{ir}+1}$.

$\texttt{send}\ x_{i-1}$
$\quad \texttt{if uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^i \cdot n\ \texttt{then}\ x_i := \mathsf{op}_0^i(\mathsf{H});$
$\quad \vdots$
$\quad \texttt{if mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{i,k} \cdot n\ \texttt{then}\ x_i := \mathsf{op}_k^i(\mathsf{H});$

$\texttt{send}\ x_{\mathbf{ir}-1}$
$\quad \texttt{if uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^{\mathbf{ir}} \cdot n\ \texttt{then}\ x_{\mathbf{ir}} := inp := \mathsf{op}_0^{\mathbf{ir}}(\mathsf{H});$
$\quad \vdots$
$\quad \texttt{if mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{\mathbf{ir},k} \cdot n\ \texttt{then}\ x_{\mathbf{ir}} := inp := \mathsf{op}_k^{\mathbf{ir}}(\mathsf{H});$

$\texttt{send}\ x_{l-1}$
$\quad \texttt{if uni}(\mathsf{H}) \wedge |\mathsf{H}| > thr_u^l \cdot n\ \texttt{then}\ dec := \mathsf{op}_0^l(\mathsf{H});$
$\quad \vdots$
$\quad \texttt{if mult}(\mathsf{H}) \wedge |\mathsf{H}| > thr_m^{l,k} \cdot n\ \texttt{then}\ dec := \mathsf{op}_k^l(\mathsf{H});$

A Characterization

▶ **Theorem 22.** *An algorithm solves consensus iff it is:*

- *syntactically safe,*
- *there are $i \leq j$ with ψ_i a unifier and ψ_j a decider.*

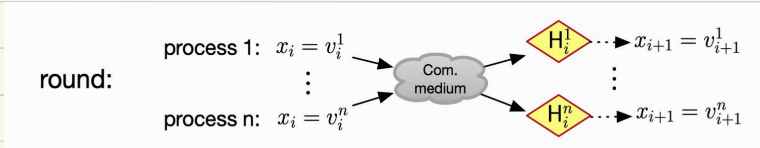
▶ **Definition 9.** *An algorithm is* syntactically safe *when:*

1. *First round has a* `mult` *instruction.*
2. *Every round has a* `uni` *instruction.*
3. *In the first round the operation in every* `mult` *instruction is* smor.
4. $thr_m^{1,k}/2 \geq 1 - thr_u^{\mathbf{ir}+1}$*, and* $thr_u^1 \geq 1 - thr_u^{\mathbf{ir}+1}$.

▶ **Lemma 23.** *An algorithm that is not syntactically safe cannot solve consensus. A syntactically safe algorithm has the agreement property.*

▶ **Lemma 24.** *A syntactically safe algorithm has the termination property iff it satisfies the second condition from Theorem 10.*

Elements of the proof



Lemma 0 If $f \overset{e}{\Rightarrow}_i f'$ and $a, b \in f'$ then $f \overset{e}{\Rightarrow}_i \mathrm{bias}(\theta)$ for every θ.

Proof

θ not an equalizer if $(\overset{a}{H_1}, \dots, \overset{b}{H_n}) \models e$ then $(H_1', \dots, H_n') \models e$

for all $H_i' \in \{H_1, H_n\}$ $\square$

Elements of the proof

Lemma 29 If round i has mult instruction then there is $\theta \geq 1/2$ s.t.
$bias(\theta) \overset{\psi}{\Longrightarrow} bias(\theta')$ for arb. θ'.

Proof

If the operation is smor then take $\theta = 1/2$.
Construct $H_a > thr_m^i \cdot n$ with more a than b. Similarly for H_b

If the operation is min, take $\theta > \max(thr_i(\psi), thr_u^i, 1/2)$
Complete H gives a
When H contains only b's it is big enough to give b. $\square$

Elements of the proof

▶ **Lemma 23.** *An algorithm that is not syntactically safe cannot solve consensus. A syntactically safe algorithm has the agreement property.*

Proof of the second statement:

Consider $(bias(\theta), ?) \xrightarrow{\overline{\Psi}} \cdots \xrightarrow{\overline{\Psi}} (bias(\theta'), d')$ first time some process, say p, decides

Recall $1 - thr_u^{ir+1} \leq thr_m^{?,k}$ and $1 - thr_u^{ir+1} \leq thr_u^{?}$.

First we show that round $ir+1$ cannot have mult instruction.

Suppose $d'(p) = a$.

This implies $\theta' < 1 - thr_u^{ir+1}$ and $d'(q) \in \{a, ?\}$, for all q.

Hence $\theta' < thr_u^{?}$, so b cannot be later obtained by uni instruction.

$\theta' < thr_m^{?,k}/2$ so b cannot be later obtained by mult instruction with smor,

(this is required for safe alg) ☐

Extensions

Timestamps

$\texttt{send}\ (inp, ts)$

- $\texttt{if}\ cond_1^1(\mathsf{H})\ \texttt{then}\ x_1 := \mathrm{maxts}(\mathsf{H});$
- $\vdots$
- $\texttt{if}\ cond_1^l(\mathsf{H})\ \texttt{then}\ x_1 := \mathrm{maxts}(\mathsf{H});$

smallest of the values of the most recent timestamp

▶ **Definition 13.** *An algorithm is* syntactically t-safe *when:*

1. *Every round has a* $\texttt{uni}$ *instruction.*
2. *First round has a* $\texttt{mult}$ *instruction.*
3. $thr_m^{1,k} \geq 1 - thr_u^{\mathbf{ir}+1}$ *and* $thr_u^1 \geq 1 - thr_u^{\mathbf{ir}+1}$.

▶ **Definition 14.** *A predicate* ψ *is a* strong unifier ψ *if it is a unifier in a sense of Definition 8 and* $thr_u^1 \leq thr_1(\psi)$.

▶ **Theorem 25.** *An algorithm satisfies consensus iff it is syntactically t-safe according to Definition 13, and it satisfies:*

sT *There are* $i \leq j$ *such that* ψ^i *is a strong unifier and* ψ^j *is a decider.*

Extensions

Coordinators

Three types of rounds
- lr (leader receive)
- ls (leader send)
- every (as before)

▶ **Theorem 26.** *An algorithm satisfies consensus iff the first round and the* $(\mathbf{ir}+1)^{th}$ *round are not of type* `ls`, *it is syntactically safe according to Definition 9, and it satisfies the condition:*

cT *There are* $i \leq j$ *such that* ψ^i *is a c-unifier and* ψ^j *is a c-decider.*

Coordinators + timestamps

▶ **Theorem 27.** *An algorithm satisfies consensus iff the first round and the* $(\mathbf{ir}+1)^{th}$ *round are not of type* `ls`, *it has the structural properties from Definition 13, and it satisfies:*

scT *There are* $i \leq j$ *such that* ψ^i *is a strong c-unifier and* ψ^j *is a c-decider.*

Examples

send (inp)
if uni(H) $\wedge |\mathsf{H}| > \frac{2}{3}n$ then $x_1 := inp :=$ smor(H);
if mult(H) $\wedge |\mathsf{H}| > \frac{2}{3}n$ then $x_1 := inp :=$ smor(H);
send x_1
if uni(H) $\wedge |\mathsf{H}| > \frac{2}{3}n$ then $dec :=$ smor(H);
Communication predicate: eventually $\psi^1 = (\varphi_= \wedge \varphi_{\frac{2}{3}}, \mathit{true})$ and later $\psi^2 = (\varphi_{\frac{2}{3}}, \varphi_{\frac{2}{3}})$

Because of the condition
$thr_m^{1,u}/2 \geq 1 - thr_u^{ir+1}$

it is not possible to have
1/2 threshold.

With timestamps the condition is weakened to $thr_m^{1,u} \geq 1 - thr_u^{ir+1}$

send (inp, ts)
if uni(H) $\wedge |\mathsf{H}| > 1/2 \cdot |\Pi|$ then $x_1 :=$ maxts(H);
if mult(H) $\wedge |\mathsf{H}| > 1/2 \cdot |\Pi|$ then $x_1 :=$ maxts(H);
send x_1
if uni(H) $\wedge |\mathsf{H}| > 1/2 \cdot |\Pi|$ then $x_2 := inp :=$ smor(H);
send x_2
if uni(H) $\wedge |\mathsf{H}| > 1/2 \cdot |\Pi|$ then $dec :=$ smor(H);
Communication predicate: $\mathsf{F}(\psi^1)$ where $\psi^1 := (\varphi_= \wedge \varphi_{1/2},\ \varphi_{1/2},\ \varphi_{1/2})$

Examples

```
send (inp, ts)
    if uni(H) ∧ |H| > 1/2 · |Π| then x_1 := maxts(H);
    if mult(H) ∧ |H| > 1/2 · |Π| then x_1 := maxts(H);
send x_1
    if uni(H) ∧ |H| > 1/2 · |Π| then x_2 := inp := smor(H);
send x_2
    if uni(H) ∧ |H| > 1/2 · |Π| then dec := smor(H);
```

Communication predicate: $\mathsf{F}(\psi^1)$ where $\psi^1 := (\varphi_= \wedge \varphi_{1/2},\ \varphi_{1/2},\ \varphi_{1/2})$

Another example with coordinators. Equalizer in round 2.

```
send (inp, ts)
    if uni(H) ∧ |H| > 1/2 · |Π| then x_1 := maxts(H);
    if mult(H) ∧ |H| > 1/2 · |Π| then x_1 := maxts(H);
send x_1
    if uni(H) ∧ |H| > 1/2 · |Π| then x_2 := smor(H);
    if mult(H) ∧ |H| > 1/2 · |Π| then x_2 := smor(H);
send x_2
    if uni(H) ∧ |H| > 1/2 · |Π| then x_3 := inp := smor(H);
send x_3
    if uni(H) ∧ |H| > 1/2 · |Π| then dec := smor(H);
```

Communication predicate: $\mathsf{F}(\psi^1 \wedge \mathsf{F}\psi^2)$
where: $\psi^1 = (\varphi_{1/2},\ \varphi_= \wedge \varphi_{1/2},\ \varphi_{1/2},\ \mathit{true})$ and $\psi^2 = (\varphi_{1/2},\ \varphi_{1/2},\ \varphi_{1/2},\ \varphi_{1/2})$

Examples

Paxos

send (inp, ts) **lr**
- **if** uni(H) $\wedge$ $|H| > 1/2 \cdot |\Pi|$ **then** $x_1 :=$ maxts(H);
- **if** mult(H) $\wedge$ $|H| > 1/2 \cdot |\Pi|$ **then** $x_1 :=$ maxts(H);

send x_1 **ls**
- **if** uni(H) **then** $x_2 := inp :=$ smor(H);

send x_2 **lr**
- **if** uni(H) $\wedge$ $|H| > 1/2 \cdot |\Pi|$ **then** $x_3 :=$ smor(H);

send x_3 **ls**
- **if** uni(H) **then** $dec :=$ smor(H);

Communication predicate: $\mathsf{F}(\psi^1)$ where $\psi^1 := (\varphi_{1/2},\ \varphi_{\mathtt{ls}},\ \varphi_{1/2},\ \varphi_{\mathtt{ls}})$

3-round Paxos

send (inp, ts) **lr**
- **if** uni(H) $\wedge$ $|H| > 1/2 \cdot |\Pi|$ **then** $x_1 :=$ maxts(H);
- **if** mult(H) $\wedge$ $|H| > 1/2 \cdot |\Pi|$ **then** $x_1 :=$ maxts(H);

send x_1 **ls**
- **if** uni(H) **then** $x_2 := inp :=$ smor(H);

send x_2 **every**
- **if** uni(H) $\wedge$ $|H| > 1/2 \cdot |\Pi|$ **then** $dec :=$ smor(H);

Communication predicate: $\mathsf{F}(\psi^1)$ where $\psi^1 := (\varphi_{1/2},\ \varphi_{\mathtt{ls}},\ \varphi_{1/2})$

Conclusions

- We wanted to do verification but arrived at a characterization
- Distributed algorithms are not algorithms:
  - for a given setting there is very little freedom for a solution
  - consensus must happen in two rounds: unifier followed by decider

Challenges

- We still do not know how to do verification, or even specify properties.
- Finding the best algorithms.
- What are reasonable extensions of this model?
  Ben-Or's algorithm.