Algorithms and Lower Bounds for the Number of Opinions in Wait-free RV

P. Fraigniaud¹ S. Rajsbaum² M. Roy³ C. Travers⁴

¹LIAFA, CNRS France

²UNAM, Mexico

³ LAAS, CNRS France

³LaBRI, France

DRV - Bertinoro May 2016











Distributed languages

A distributed language $\mathcal{L} \subseteq \bigcup_{i \leq \ell} A^i$ is a collection of vectors $[a_1, \ldots, a_\ell]$ representing the valid states of the system

Distributed languages

A distributed language $\mathcal{L} \subseteq \bigcup_{i \leq \ell} A^i$ is a collection of vectors $[a_1, \ldots, a_\ell]$ representing the valid states of the system

 $\frac{\text{AGREEMENT}}{[a_1, \dots, a_\ell] \in \text{AGREEMENT}} \iff a_i = a_j \forall i, j$

 $\begin{array}{l} \underline{\text{CONSENSUS}} \\ [(v_1, d_1), \dots, (v_{\ell}, d_{\ell})] \in \text{CONSENSUS} \\ \longleftrightarrow \begin{cases} 1. \text{agreement} & d_i = d_j = d \ \forall i, j \\ 2. \text{validity} & \exists i : v_i = d \end{cases} \end{array}$

A monitor for a language ${\mathcal L}$ consists in 3 components

- A set of opinions U
- A mapping μ : multiset of $\mathbf{U} \rightarrow {\{\mathbf{YES}, \mathbf{NO}\}}$
- A distributed monitoring protocol

Let $v \in A^{\ell}$ be the state of the system Each monitors **i** gets as input a *partial vector* $v'_i \in (\{\bot\} \cup A)^{\ell}$:

•
$$v'_i[j] \neq \bot \implies v'_i[j] = v[j]$$

• for each
$$j, 1 \leq j \leq \ell, \exists i : v'_i[j] = v[j]$$

Monitoring protocol

Let $v \in A^{\ell}$ be the state of the system Each monitors i gets as input a *partial vector* $v'_i \in (\{\bot\} \cup A)^{\ell}$:

•
$$v'_i[j] \neq \bot \implies v'_i[j] = v[j]$$

• for each $j, 1 \leq j \leq \ell, \exists i : v'_i[j] = v[j]$

communicates with the others, and emits an opinion ui.



- Number of monitors $n = \ell$ number of state components
- Each monitor **i** gets as intput v[i]

Indulgent algorithms

Systems are synchronous ... with arbitrary asynchronous period



Wait-free shared memory



Shared memory

• *n* Processes $\{p_1, \ldots, p_n\}$

Wait-free shared memory



Shared memory

- *n* Processes $\{p_1, \ldots, p_n\}$
- Asynchronous communications
 - read() from/write() to any memory cell
 - finite but unbounded delay between steps

Wait-free shared memory



Shared memory

- *n* Processes $\{p_1, \ldots, p_n\}$
- Asynchronous communications
 - read() from/write() to any memory cell
 - finite but unbounded delay between steps
- Each process is subject to crash-failure

Shared memory models

- SRSW safe bit
- MRSW safe bit
- MRSW regular bit
- MRSW regular
- MRSW atomic
- MRMW atomic
- Atomic snapshot

Shared memory models

- SRSW safe bit
- MRSW safe bit
- MRSW regular bit
- MRSW regular
- MRSW atomic
- MRMW atomic
- Atomic snapshot 🔶

Wait-free construction

an array A of SWMR registers

- update(*i*, *v*) : writes *v* to *A*[*i*]
- scan() : takes instantaneous snapshot of A

Shared memory vs. Message passing

Atomic registers can be simulated in the asynchronous message-passing model when a majority of the processes are correct

$$\underline{\mathsf{Opinions}} = \{\mathbf{0}, \mathbf{1}\}$$

Monitoring protocol
 $update (i, a_i)$ at monitor i with input a_i : $update (i, a_i)$ scan() memory to get set a'if agree(a')then decide 1else decide 0.

Interpretation on input multiset $S \subseteq \{0, 1\}$: **AND**(S)

AND interpretation: characterization

Not every distributed language can be monitored with

- two opinions
- AND interpretation

Theorem (Fraigniaud, Rajsbaum, T. disc'11) \mathcal{L} has a monitor with **AND** interpretation $\iff \mathcal{L}$ is projection-closed.

Let \mathcal{L} be a distributed language. How many **opinions** are needed to wait-free monitor \mathcal{L} ?

Let \mathcal{L} be a distributed language. How many **opinions** are needed to wait-free monitor \mathcal{L} ?

 $\# \mathsf{opinions}(\mathcal{L}) = \mathsf{number}$ of opinions necessary and sufficient to wait-free monitor \mathcal{L}

Theorem (Fraigniaud, Rajsbaum, T. RV '14) Every distributed language on n processes can be monitored with n + 1 opinions

Theorem (Fraigniaud, Rajsbaum, T. RV '14) Every distributed language on n processes can be monitored with n + 1 opinions

Theorem (Fraigniaud, Rajsbaum, T. RV '14) For every $1 \le k \le n$, there is a n-process language \mathcal{L}_k such that $\# opinions(\mathcal{L}_k) = k + 1$

Monitoring CONSENSUS

$\begin{array}{l} \underline{\text{CONSENSUS}}\\ (s_1, t_1), \dots, (s_{\ell}, t_{\ell}) \in \text{CONSENSUS} \\ \iff \begin{cases} 1. \text{agreement} & t_i = t_j = t \; \forall i, j \\ 2. \text{validity} & \exists i : s_i = t \end{cases} \end{array}$

Monitoring CONSENSUS

$\begin{array}{l} \underline{\text{CONSENSUS}}\\ (s_1, t_1), \dots, (s_{\ell}, t_{\ell}) \in \text{CONSENSUS} \\ \iff \begin{cases} 1. \text{agreement} & t_i = t_j = t \; \forall i, j \\ 2. \text{validity} & \exists i : s_i = t \end{cases} \end{array}$

Examples:

- $(1,1), (2,1), (3,2) \notin \mathcal{L}$
- (1, 3), (2, 3) ∉ L
- $(1, 2), (2, 2), (3, 2) \in \mathcal{L}$

Monitoring CONSENSUS

$\begin{array}{l} \underline{\text{CONSENSUS}}\\ (s_1, t_1), \dots, (s_{\ell}, t_{\ell}) \in \text{CONSENSUS} \\ \iff \begin{cases} 1. \text{agreement} & t_i = t_j = t \; \forall i, j \\ 2. \text{validity} & \exists i : s_i = t \end{cases} \end{array}$

Examples:

- $(1,1), (2,1), (3,2) \notin \mathcal{L}$
- (1, 3), (2, 3) ∉ L
- $(1, 2), (2, 2), (3, 2) \in \mathcal{L}$

Theorem #opinions(CONSENSUS) = 3

Traffic-light monitor

Set of opinions = $\{green, orange, red\}$

Traffic-light monitor

$\underline{\mathsf{Set of opinions}} = \{ \mathbf{green}, \mathbf{orange}, \mathbf{red} \}$

```
\frac{\text{Monitoring-protocol}}{update(i, (s_i, t_i))} \text{ at monitor } \mathbf{i} \text{ with input } (s_i, t_i):
\frac{update(i, (s_i, t_i))}{snapshot()} \text{ to get } (s', t')
\mathbf{case} \text{ agree}(t') \text{ and } \text{ valid}(s', t') \text{ then decide green}
\text{ agree}(t') \text{ but not } \text{ valid}(s', t') \text{ then decide orange}
\text{ not agree}(t') \text{ then decide red}
\mathbf{end case}
```

Traffic-light monitor

Set of opinions = $\{green, orange, red\}$

 $\begin{array}{l} \underline{\text{Monitoring-protocol}} \text{ at monitor } \mathbf{i} \text{ with input } (s_i, t_i):\\ \hline update(i, (s_i, t_i))\\ snapshot() \text{ to get } (s', t')\\ \textbf{case} \text{ agree}(t') \text{ and } \text{valid}(s', t') \text{ then decide green}\\ & \text{ agree}(t') \text{ but not } \text{valid}(s', t') \text{ then decide orange}\\ & \text{ not agree}(t') \text{ then decide red}\\ \hline \textbf{end case} \end{array}$

Interpretation on multiset $S \subseteq \{$ red, orange, green $\}$ if red $\in S$ or green $\notin S$ then return NO else return YES

 $\frac{\text{Interpretation}}{\mu: \text{ multiset}} \subseteq \{\text{red, orange, green}\} \longrightarrow \{\text{YES}, \text{NO}\}$

Interpretation

 μ : multiset \subseteq {red, orange, green} \longrightarrow {YES, NO}

 $\begin{array}{l} \underline{ \text{Interpretation } \mu \text{ on multiset } S \subseteq \{ \textbf{red, orange, green} \} \\ \hline \textbf{if red} \in S \text{ or green} \notin S \text{ then return NO} \\ \hline \textbf{else return YES} \end{array}$

Interpretation

 μ : multiset \subseteq {red, orange, green} \longrightarrow {YES, NO}

 $\begin{array}{l} \underline{ \text{Interpretation } \mu \text{ on multiset } S \subseteq \{ \textbf{red, orange, green} \} \\ \hline \textbf{if red} \in S \text{ or green} \notin S \text{ then return NO} \\ \hline \textbf{else return YES} \end{array}$

Interpretation

 μ : multiset \subseteq {red, orange, green} \longrightarrow {YES, NO}

 $\begin{array}{l} \hline \\ \hline \\ \hline \\ \textbf{Interpretation} \\ \mu \\ \textbf{ on multiset } S \subseteq \{ \textbf{red, orange, green} \} \\ \hline \\ \hline \\ \textbf{if red} \\ \in S \\ \textbf{ or green} \\ \not \in S \\ \textbf{ then return NO} \\ \hline \\ \\ \textbf{else return YES} \\ \end{array}$

 Monitor i decides red whenever it sees disagreement. This cannot be fixed even if a'_i ⊊ a.
Traffic-light monitor: correctness

Interpretation

 μ : multiset \subseteq {red, orange, green} \longrightarrow {YES, NO}

 $\begin{array}{l} \hline \\ \hline \\ \hline \\ \textbf{Interpretation} \\ \mu \\ \textbf{ on multiset } S \subseteq \{ \textbf{red, orange, green} \} \\ \hline \\ \hline \\ \textbf{if red} \\ \in S \\ \textbf{ or green} \\ \not \in S \\ \textbf{ then return NO} \\ \hline \\ \\ \textbf{else return YES} \\ \end{array}$

- Monitor i decides red whenever it sees disagreement.
 This cannot be fixed even if a'_i ⊊ a.
- No **red** means agreement. No **green** however indicates validity violation.

Traffic-light monitor: correctness

Interpretation

 μ : multiset \subseteq {red, orange, green} \longrightarrow {YES, NO}

 $\begin{array}{l} \hline \\ \hline \\ \hline \\ \textbf{Interpretation} \\ \mu \\ \textbf{ on multiset } S \subseteq \{ \textbf{red, orange, green} \} \\ \hline \\ \hline \\ \textbf{if red} \\ \in S \\ \textbf{ or green} \\ \not \in S \\ \textbf{ then return NO} \\ \hline \\ \\ \textbf{else return YES} \\ \end{array}$

- Monitor i decides red whenever it sees disagreement. This cannot be fixed even if a'_i ⊊ a.
- No **red** means agreement. No **green** however indicates validity violation.
- green and no red imply agreement and validity.

$\# opinions(CONSENSUS) \ge 3$

$\# opinions(CONSENSUS) \geq 3$

$$\underbrace{ \substack{ \notin \mathcal{L} \\ (0,1) \\ \bullet \end{array}}_{ \in \mathcal{L}} \underbrace{ \begin{array}{c} \in \mathcal{L} \\ (1,1) \\ \notin \mathcal{L} \end{array} }_{ \notin \mathcal{L}} \underbrace{ \begin{array}{c} \notin \mathcal{L} \\ (0,2) \\ \bullet \end{array} }_{ \bullet }$$

$\# opinions(CONSENSUS) \geq 3$



$\# opinions(CONSENSUS) \geq 3$



#opinions(CONSENSUS) ≥ 3



#opinions(CONSENSUS) ≥ 3



Contradiction : $\mu(\{\text{green}, \text{red}\}) = \text{YES}, \text{NO}$

$$\begin{array}{l} \underline{\text{K-SET AGREEMENT}} \\ \{(\textit{id}_1, (\textit{s}_1, \textit{t}_1)), \dots, (\textit{id}_\ell, (\textit{s}_\ell, \textit{t}_\ell)\} \in \text{K-SET-AGREEMENT} \\ \iff \begin{cases} 1.\text{k-agreement} & |\{\textit{t}_1, \dots, \textit{t}_\ell\}| \leq k \\ 2.\text{validity} & \{\textit{t}_1, \dots, \textit{t}_\ell\} \subseteq \{\textit{s}_1, \dots, \textit{s}_\ell\} \end{cases} \end{array}$$

$$\begin{array}{l} \underline{\text{K-SET AGREEMENT}} \\ \{(\textit{id}_1, (\textit{s}_1, \textit{t}_1)), \dots, (\textit{id}_{\ell}, (\textit{s}_{\ell}, \textit{t}_{\ell})\} \in \text{K-SET-AGREEMENT} \\ \\ \iff \begin{cases} 1.\text{k-agreement} & |\{\textit{t}_1, \dots, \textit{t}_{\ell}\}| \leq k \\ 2.\text{validity} & \{\textit{t}_1, \dots, \textit{t}_{\ell}\} \subseteq \{\textit{s}_1, \dots, \textit{s}_{\ell}\} \end{cases} \end{array}$$

Examples:

- $\{(1, 1), (2, 1), (3, 2)\} \in 2$ -set-agreement
- $\{(1, 1), (2, 1), (3, 0)\} \notin 2$ -Set-Agreement
- $\{(1, 1), (2, 2), (3, 3)\} \notin 2\text{-set-agreement}$

How many opinions for K-SET-AGREEMENT?

• 1-SET-AGREEMENT (CONSENSUS) : #opinions = 3

How many opinions for K-SET-AGREEMENT?

- 1-SET-AGREEMENT (CONSENSUS) : #opinions = 3
- <u>K-SET-AGREEMENT</u> : #opinions = ???

How many opinions for K-SET-AGREEMENT?

- 1-SET-AGREEMENT (CONSENSUS) : #opinions = 3
- <u>K-SET-AGREEMENT</u> : #opinions = ???

Theorem #opinions(K-SET-AGREEMENT) = min(2K, n) + 1

Assume :

• **k** < *n*/2

• There is a monitoring protocol *M* and an interpretation μ that use *x* opinions

We show that $x \ge 2k + 1$





• $\{(0,1)\} \notin \mathcal{L}$ $\mu(\{ullet\}) = \mathbf{No}$





•
$$\{(0,1)\} \notin \mathcal{L}$$
 $\mu(\{ullet\}) = \mathsf{No}$





•
$$\{(0,1)\} \notin \mathcal{L}$$

• $\{(1,1)\} \in \mathcal{L}$

 $\mu(\{\bullet\}) = No$ $\mu(\{\bullet\}) = Yes$



• $\{(0,1)\} \notin \mathcal{L}$ $\mu(\{\bullet\}) = No$ • $\{(1,1)\} \in \mathcal{L}$ $\mu(\{\bullet\}) = Yes$



 $\begin{array}{ll} \bullet \ \{(0,1)\} \notin \mathcal{L} & \mu(\{\bullet\}) = \mathsf{No} \\ \bullet \ \{(1,1)\} \in \mathcal{L} & \mu(\{\bullet\}) = \mathsf{Yes} \end{array} \end{array}$



• $\{(0,1)\} \notin \mathcal{L}$ $\mu(\{\bullet\}) = \mathsf{No}$ • $\{(1,1)\} \in \mathcal{L}$ $\mu(\{\bullet\}) = \mathsf{Yes}$



- $\{(0,1)\} \notin \mathcal{L}$ $\mu(\{\bullet\}) = \mathsf{No}$
- $\{(1,1)\} \in \mathcal{L}$ $\mu(\{\bullet\}) =$ Yes
- $\{(0,1),(1,1)\} \in \mathcal{L}$ $\mu(\{\bullet,\bullet\}) =$ Yes



















(A Variant of) Sperner lemma



#opinions > 2k + 1 (end)



#opinions > 2k + 1 (end)



#opinions > 2k + 1 (end)



#opinions > 2k + 1 (end)



#opinions > 2k + 1 (end)



Alternating sequence

$$2k + 1 \begin{cases} \{(0,1)\} \\ \{(0,1),(1,1)\} \\ \{(0,1),(1,1),(1,2)\} \\ \{(0,1),(1,1),(1,2),(2,2)\} \\ \{(0,1),(1,1),(1,2),(2,2),(2,3)\} \end{cases}$$
Alternating sequence



Opinions for k-SET-AGREEMENT

Opinions = $\{\text{red}\} \cup \{(\text{orange}, \ell) : 1 \le \ell \le k\} \cup \{(\text{green}, \ell) : 1 \le \ell \le k\}$

 $\# \mathsf{Opinions} = 2\mathbf{k} + \mathbf{1}$

```
\frac{\text{Monitoring protocol}}{update(i, (s_i, t_i))} \text{ at monitor } i \text{ with input } (s_i, t_i):
\frac{update(i, (s_i, t_i))}{snapshot() \text{ to get } (s', t')}
case
endcase
```

```
\frac{\text{Monitoring protocol}}{update(i, (s_i, t_i))} \text{ at monitor } i \text{ with input } (s_i, t_i):
\frac{update(i, (s_i, t_i))}{snapshot() \text{ to get } (s', t')}
d \leftarrow \# \text{decided values}(t')
case
endcase
```

```
\frac{\text{Monitoring protocol}}{update(i, (s_i, t_i))} \text{ at monitor } i \text{ with input } (s_i, t_i):
\frac{update(i, (s_i, t_i))}{snapshot()} \text{ to get } (s', t')
d \leftarrow \# \text{decided values}(t')
\textbf{case } d \leq k \text{ and valid}(s', t') \text{ then decide } (d, \textbf{green})
\textbf{endcase}
```

```
 \begin{array}{l} \underline{\text{Monitoring protocol}}_{update(i, (s_i, t_i))} \\ \underline{\text{snapshot}()}_{snapshot()} \text{ to get } (s', t') \\ d \leftarrow \# \text{decided values}(t') \\ \textbf{case } d \leq k \text{ and valid}(s', t') \\ d \leq k \text{ and not valid}(s', t') \text{ then decide } (d, \textbf{green}) \\ d \leq k \text{ and not valid}(s', t') \text{ then decide } (d, \textbf{orange}) \\ \textbf{endcase} \end{array}
```

```
\begin{array}{l} \underline{\text{Monitoring protocol}}_{update(i, (s_i, t_i))} \\ \underline{\text{Monitoring protocol}}_{update(i, (s_i, t_i))} \\ \underline{\text{snapshot}}() \text{ to get } (s', t') \\ \underline{d} \leftarrow \# \text{decided values}(t') \\ \underline{\text{case }} d \leq k \text{ and valid}(s', t') \\ \underline{d} \leq k \text{ and valid}(s', t') \text{ then decide } (d, \text{green}) \\ \underline{d} \leq k \text{ and not valid}(s', t') \text{ then decide } (d, \text{orange}) \\ \underline{d} > k \\ \underline{\text{then decide } (\text{red})}; \\ \underline{\text{endcase}} \end{array}
```

 $\begin{aligned} & \text{Opinions} = \\ & \{\text{red}\} \cup \{(\text{orange}, \ell) : 1 \leq \ell \leq k\} \cup \{(\text{green}, \ell) : 1 \leq \ell \leq k\} \end{aligned}$

Opinions = $\{\text{red}\} \cup \{(\text{orange}, \ell) : 1 \le \ell \le k\} \cup \{(\text{green}, \ell) : 1 \le \ell \le k\}$

Let S a multiset of opinions. Define $level(\mathbf{c},S) = \max\{d : (\mathbf{c},d) \in S\}$ $\begin{aligned} & \text{Opinions} = \\ & \{\text{red}\} \cup \{(\text{orange}, \ell) : 1 \leq \ell \leq k\} \cup \{(\text{green}, \ell) : 1 \leq \ell \leq k\} \end{aligned}$

Let S a multiset of opinions. Define $level(\mathbf{c},S) = \max\{d : (\mathbf{c},d) \in S\}$



 $\begin{aligned} \mathsf{Opinions} &= \\ \{\mathsf{red}\} \cup \{(\mathsf{orange}, \ell) : 1 \leq \ell \leq k\} \cup \{(\mathsf{green}, \ell) : 1 \leq \ell \leq k\} \end{aligned}$

Let S a multiset of opinions. Define $level(\mathbf{c}, S) = \max\{d : (\mathbf{c}, d) \in S\}$

• red $\in S$ $\mu(S) = NO$ • red $\notin S \land level(orange, S) > level(green, S)$ $\mu(S) = NO$ $\begin{aligned} \mathsf{Opinions} &= \\ \{\mathsf{red}\} \cup \{(\mathsf{orange}, \ell) : 1 \leq \ell \leq k\} \cup \{(\mathsf{green}, \ell) : 1 \leq \ell \leq k\} \end{aligned}$

Let S a multiset of opinions. Define $level(\mathbf{c}, S) = \max\{d : (\mathbf{c}, d) \in S\}$

• $\operatorname{red} \in S$ $\mu(S) = \operatorname{NO}$ • $\operatorname{red} \notin S \land \operatorname{level}(\operatorname{orange}, S) > \operatorname{level}(\operatorname{green}, S)$ $\mu(S) = \operatorname{NO}$ • $\operatorname{red} \notin S \land \operatorname{level}(\operatorname{orange}, S) \leq \operatorname{level}(\operatorname{green}, S)$ $\mu(S) = \operatorname{YES}$

Definition

The alternation number #altern (\mathcal{L}) of a language \mathcal{L} is the largest integer **k** for which there exists $a_1 \subset a_2 \subset \ldots a_k$ such that for every *i* either

 $a_i \in \mathcal{L}$ and $a_{i+1} \notin \mathcal{L}$ or $a_i \notin \mathcal{L}$ and $a_{i+1} \in \mathcal{L}$.

Theorem

There exists a protocol which, for every $k \ge 1$, monitors every language with alternation number k, by using at most k + 1 opinions.

Theorem

There exists a protocol which, for every $k \ge 1$, monitors every language with alternation number k, by using at most k + 1 opinions.

Theorem

For any $n \ge 1$, and every k, $1 \le k \le n$, there are some languages on n processes with alternation number k, for which monitoring requires $\Omega(k)$ opinions.

- Non anonymous monitoring: The outputs of the monitors are opinions. The interpretation does not use who expresses which opinion.
- Monitors received one input and collectively check whether the set of inputs satisfied some property. What if monitors receive instead a sequence of inputs?

Thanks!