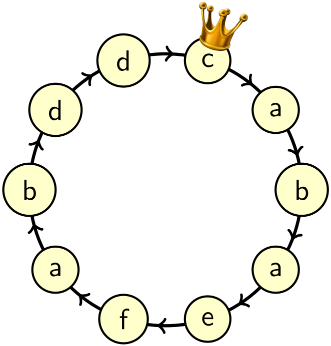
Leader Election in Asymmetric Labeled Unidirectional Rings

Karine Altisen, Ajoy K. Datta, Stéphane Devismes, **Anaïs Durand**, and Lawrence L. Larmore

May 30, 2017

Context



- Leader election
- Unidirectional rings
- Homonym processes
- Deterministic algorithm
- Asynchronous message-passing

State of the Art

Leader Election in Rings

	Deterministic solution
Anonymous processes	Impossible [Angluin, 80] [Lynch, 96]

Leader Election in Rings

	Deterministic solution		Probabilistic solution
Anonymous processes	Impossible [Angluin, 80] [Lynch, 96]	⟹	Possible [Xu and Srimani, 06] [Kutten *et al.*, 13]

State of the Art

Leader Election in Rings

	Deterministic solution	**Probabilistic** solution
Anonymous processes	Impossible [Angluin, 80] [Lynch, 96]	Possible [Xu and Srimani, 06] [Kutten *et al.*, 13]
Identified processes	Possible [LeLann, 77] [Chang and Roberts, 79] [Peterson, 82]	

Leader Election in Rings

+

Deterministic solution → **Probabilistic** solution

Anonymous processes

- Deterministic: Impossible [Angluin, 80] [Lynch, 96]
- Probabilistic: Possible [Xu and Srimani, 06] [Kutten *et al.*, 13]

Homonym processes [Yamashita and Kameda, 89]

Identified processes

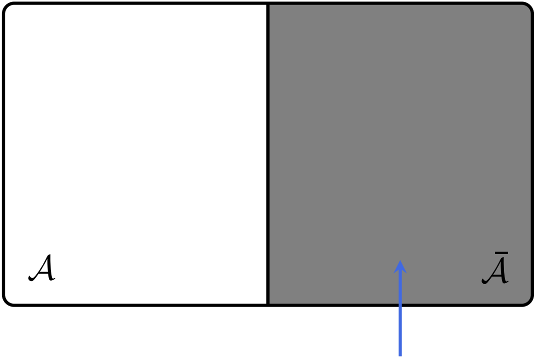
- Deterministic: Possible [LeLann, 77] [Chang and Roberts, 79] [Peterson, 82]

Leader Election in Rings of Homonym Processes

	PT/MT	Asynch.	Uni./Bi.	Known	Ring Class	# Msg	Time	Memory
[Delporte *et al.*, 14]	MT	✓	Bi.		# labels > greatest proper divisor of n	?	?	?
	PT	✓		n		$O(n \log n)$	?	?
[Dobrev, Pelc, 04]	PT	✗	Bi. + uni.	$m \leq n$	Decide if inputs are unambiguous	$O(n \log n)$	$O(M)$	$O(nb)$
		✓	Bi.	$M \geq n$		$O(nM)$	?	$O(Mb)$
[SSS 2016]	PT	✓	Uni.	k	∃ unique label and # proc with same label ≤ k	$O(kn)$	$O(kn)$	$O(\log k + b)$
[IPDPS 2017]	PT	✓	Uni.	k	Asymmetric labelling and # proc with same label ≤ k	$O(n^2 + kn)$	$O(kn)$	$O(knb)$
						$O(k^2n^2)$	$O(k^2n^2)$	$O(\log k + b)$

- Uni : Unidirectional / Bi : Bidirectional
- MT = Message-terminating: Processes do not explicitly terminate but only a finite number of messages are exchanged.
- PT = Process-terminating: Every process eventually halts.

Contributions

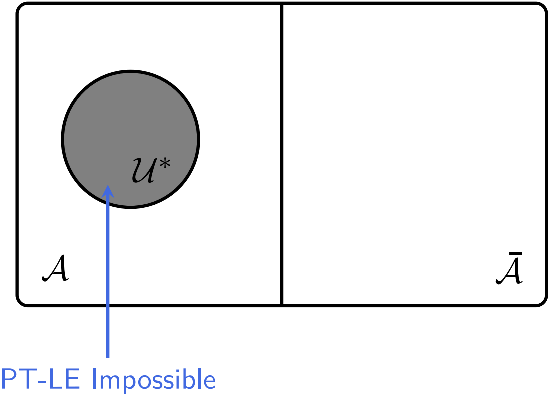


MT-LE Impossible [Angluin,80]

- MT-LE: Message-Terminating Leader Election
- PT-LE: Process-Terminating Leader Election

- $\mathcal{A}$: Rings with asymmetric labelling
- $\overline{\mathcal{A}}$: Rings with symmetric labelling
- $\mathcal{U}^*$: Rings with at least one unique label
- $\mathcal{K}_k$: Rings with no more than k processes with the same label

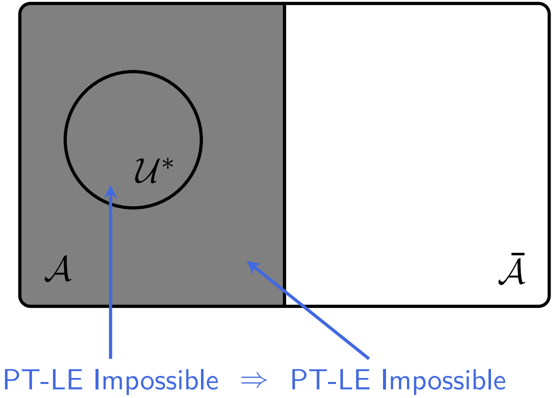
Contributions



- MT-LE: Message-Terminating Leader Election
- PT-LE: Process-Terminating Leader Election

- $\mathcal{A}$: Rings with asymmetric labelling
- $\overline{\mathcal{A}}$: Rings with symmetric labelling
- $\mathcal{U}^*$: Rings with at least one unique label
- $\mathcal{K}_k$: Rings with no more than k processes with the same label

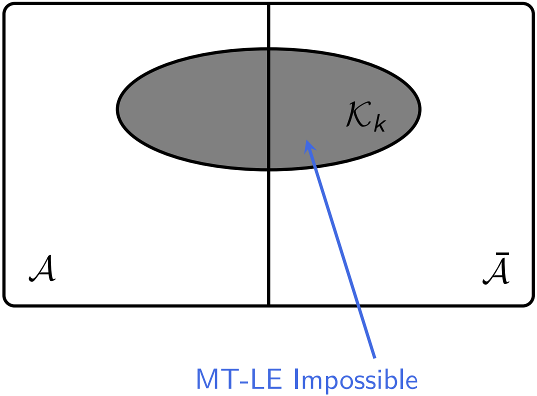
Contributions



- MT-LE: Message-Terminating Leader Election
- PT-LE: Process-Terminating Leader Election

- $\mathcal{A}$: Rings with asymmetric labelling
- $\overline{\mathcal{A}}$: Rings with symmetric labelling
- $\mathcal{U}^*$: Rings with at least one unique label
- $\mathcal{K}_k$: Rings with no more than k processes with the same label

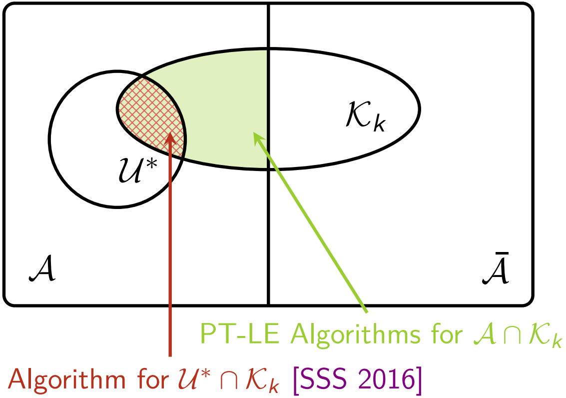
Contributions



- MT-LE: Message-Terminating Leader Election
- PT-LE: Process-Terminating Leader Election

- $\mathcal{A}$: Rings with asymmetric labelling
- $\overline{\mathcal{A}}$: Rings with symmetric labelling
- $\mathcal{U}^*$: Rings with at least one unique label
- $\mathcal{K}_k$: Rings with no more than k processes with the same label

Contributions

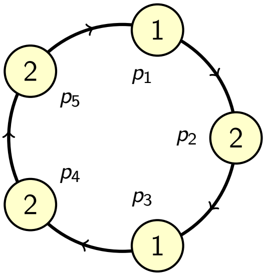


- MT-LE: Message-Terminating Leader Election
- PT-LE: Process-Terminating Leader Election

- $\mathcal{A}$: Rings with asymmetric labelling
- $\overline{\mathcal{A}}$: Rings with symmetric labelling
- $\mathcal{U}^*$: Rings with at least one unique label
- $\mathcal{K}_k$: Rings with no more than k processes with the same label

First PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

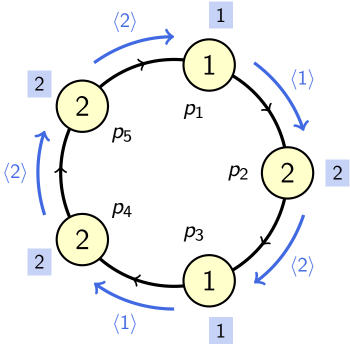
- **Chosen Leader:**
  process whose LabelSequence = LyndonWord(LabelSequence)
  Lyndon Word = smallest rotation in lexicographic order



- **Label Sequence at p_1:**
  $LS_{p_1} = 12212$
  Rotations:
  12212 $(= LS_{p_1})$
  21221 $(= LS_{p_2})$
  12122 $(= LS_{p_3})$ $LW \neq LS_{p_1}$
  21212 $(= LS_{p_4})$
  22121 $(= LS_{p_5})$

First PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

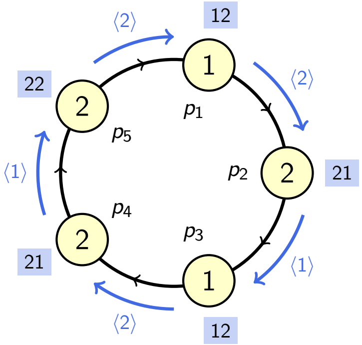
- **Chosen Leader:**
  process whose LabelSequence = LyndonWord(LabelSequence)
  Lyndon Word = smallest rotation in lexicographic order



- Local label aggregation

First PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

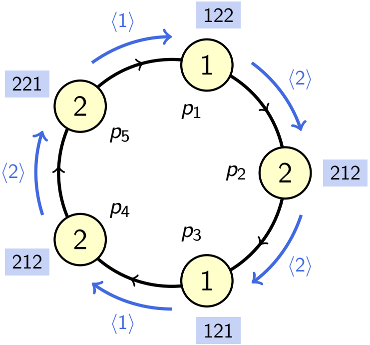
- **Chosen Leader:**
  process whose LabelSequence = LyndonWord(LabelSequence)
  Lyndon Word = smallest rotation in lexicographic order



- Local label aggregation

First PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

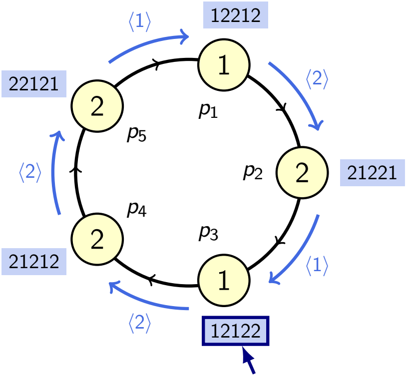
- **Chosen Leader:**
  process whose LabelSequence = LyndonWord(LabelSequence)
  Lyndon Word = smallest rotation in lexicographic order



- Local label aggregation

First PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

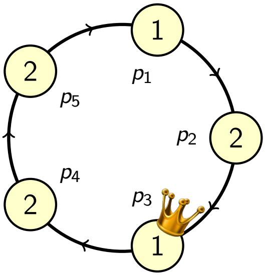
- **Chosen Leader:**
  process whose LabelSequence = LyndonWord(LabelSequence)
  Lyndon Word = smallest rotation in lexicographic order



- Local label aggregation
- ⚠ Do not know n
  $\Rightarrow$ Leader cannot detect its election

First PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

- **Chosen Leader:**
  process whose LabelSequence = LyndonWord(LabelSequence)
  Lyndon Word = smallest rotation in lexicographic order



$k = 3$

12122 1212212

Smallest repeating prefix = LabelSequence
= LyndonWord(Smallest repeating prefix)

- Local label aggregation
- ⚠ Do not know n
  ⇒ Leader cannot detect its election
- Termination detection = $(2k+1) \times$ the same label
  ⇒ at least 2 times the sequence of labels

First PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

- **Time complexity:** at most $(2k+2)n$ time units
- **Message complexity:** at most $n^2(2k+1)$ messages
- **Memory:** $(2k+1)nb + 2b + 3$ bits,
  where $b =$ number of bits to store an ID

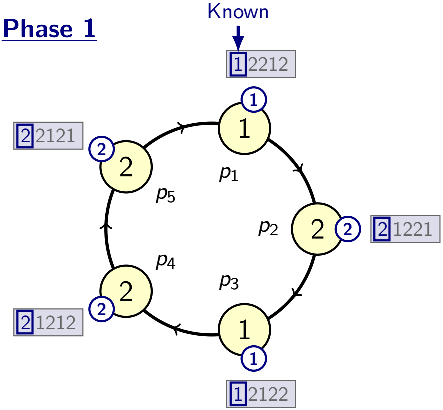
Asymptotically optimal time complexity

but

Large memory requirement

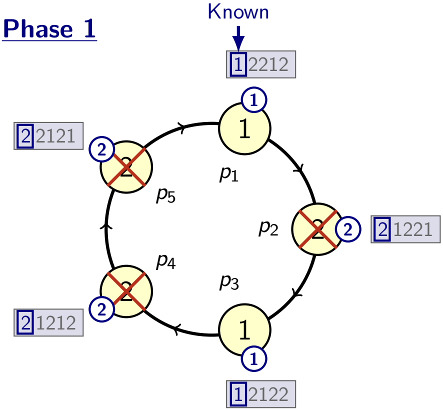
Second PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

- **Decrease memory usage** $\Rightarrow$ Peterson principle with radix sort



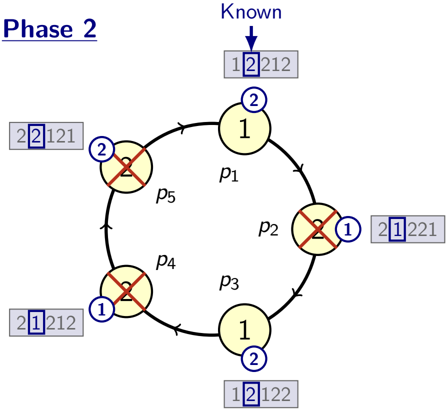
Second PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

- **Decrease memory usage** $\Rightarrow$ Peterson principle with radix sort



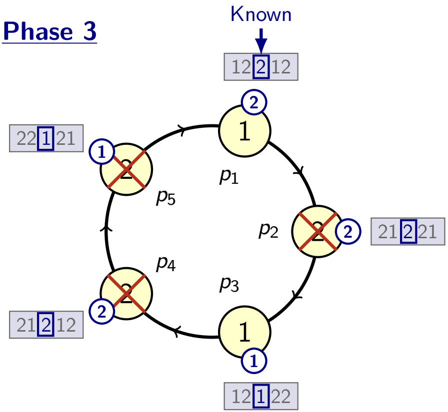
Second PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

- **Decrease memory usage** ⇒ Peterson principle with radix sort

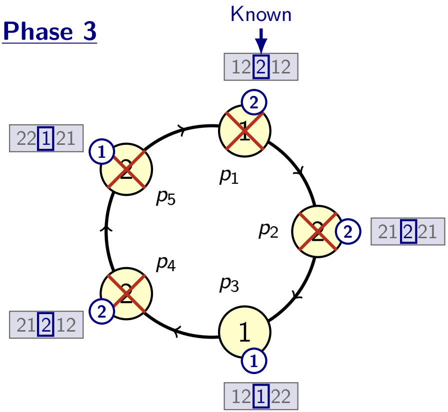


Second PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

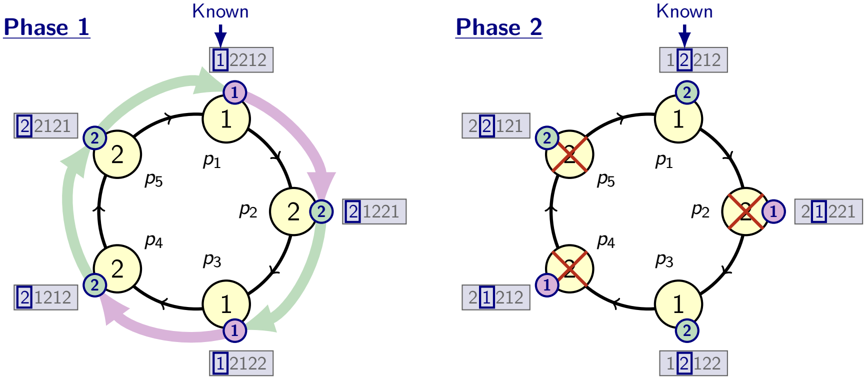
- **Decrease memory usage** $\Rightarrow$ Peterson principle with radix sort



- **Decrease memory usage** ⇒ Peterson principle with radix sort

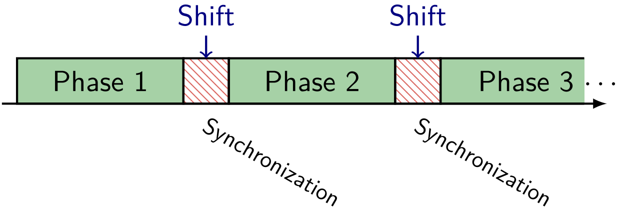


Phase Shift


Phase 1
Known
12212
22121
21221
21212
12122
p_1
p_2
p_3
p_4
p_5
Phase 2
Known
12212
22121
21221
21212
12122
p_1
p_2
p_3
p_4
p_5

Second PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

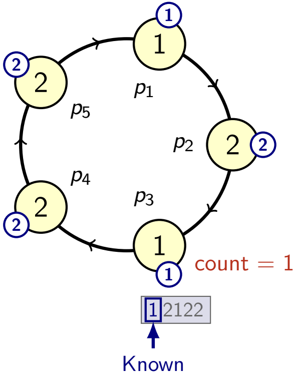
- **Execution**



Second PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

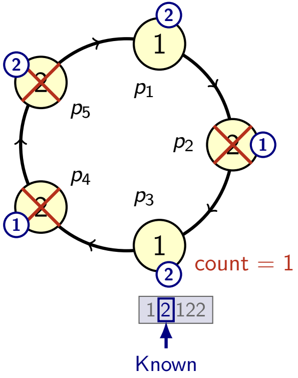
- **Termination Detection:** count = k+1
  count = # phases where Known = Label

Phase 1



- **Termination Detection:** count = k+1
  count = # phases where Known = Label

Phase 2

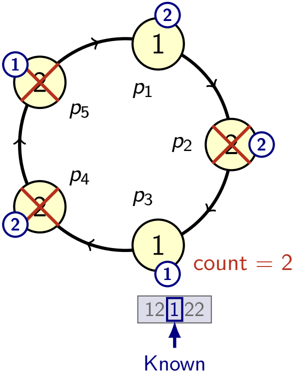


Second PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

- **Termination Detection:** count = k+1

  count = # phases where Known = Label

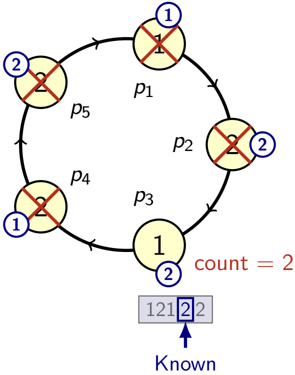
Phase 3



- **Termination Detection:** count = k+1

  count = # phases where Known = Label

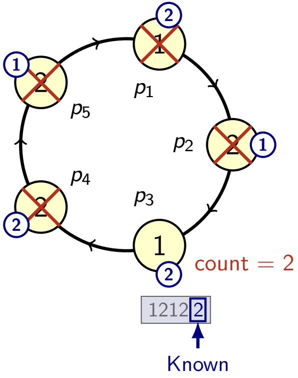
Phase 4



- **Termination Detection:** count = k+1

  count = # phases where Known = Label

Phase 5

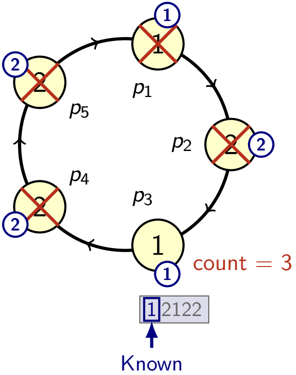


Second PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

- **Termination Detection:** count = k+1

  count = # phases where Known = Label

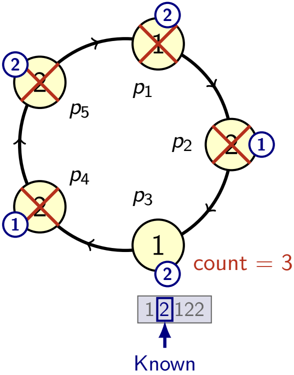
Phase 6



Second PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

- **Termination Detection:** count = k+1
  count = # phases where Known = Label

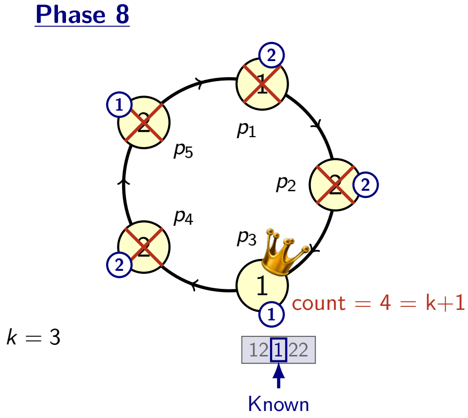
Phase 7



Second PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

- **Termination Detection:** count = k+1

  count = # phases where Known = Label



Second PT-LE Algorithm for $\mathcal{A} \cap \mathcal{K}_k$

- **Memory:** $2\lceil \log k \rceil + 3b + 5$ bits,
  where $b =$ number of bits to store an ID
- **Time complexity:** $O(k^2 n^2)$ time units
- **Message complexity:** $O(k^2 n^2)$ messages

Asymptotically optimal memory requirement
but
Large time complexity

Conclusion

Class	
$\overline{\mathcal{A}}$	Message-terminating leader election impossible
$\mathcal{K}_k$	Message-terminating leader election impossible
$\mathcal{U}^*$	Process-terminating leader election impossible
$\mathcal{A}$	Process-terminating leader election impossible

Class	Lower Bound on Time	Time	Nbr of Msgs	Memory
$\mathcal{U}^* \cap \mathcal{K}_k$ [SSS 2016]	$\Omega(kn)$	$O(kn)$	$O(n^2 + kn)$	$O(\log k + b)$
$\mathcal{A} \cap \mathcal{K}_k$	$\Omega(kn)$	$O(kn)$	$O(n^2 k)$	$O(knb)$
		$O(k^2 n^2)$	$O(k^2 n^2)$	$O(\log k + b)$

$b = \#$ bits to store a label

- $\mathcal{A}$: Rings with asymmetric labelling
- $\overline{\mathcal{A}}$: Rings with symmetric labelling
- $\mathcal{U}^*$: Rings with at least one unique label
- $\mathcal{K}_k$: Rings with no more than k processes with the same label

Thank you for your attention.

Do you have any questions ?