

# Leader Election Algorithm in Anonymous Rings: Franklin Goes Probabilistic

Rena Bakhshi

Theoretical Computer Science Section  
Department of Computer Science  
Vrije Universiteit Amsterdam

Milan, September 9, 2008

## Joint work with

- Wan Fokkink
  - Vrije Universiteit Amsterdam
- Jun Pang
  - Université du Luxembourg
- Jaco van de Pol
  - University of Twente

Thanks to

- Bert Lisser
  - CWI, Amsterdam

## Leader Election

- Franklin's Leader Election Algorithm

- Leader Election in Anonymous Networks

- Our Leader Election Algorithm

## Verification

- Verification with  $\mu$ CRL and CADP

- Partial order reduction with  $\mu$ CRL

- Performance evaluation with PRISM

## Leader Election

- Franklin's Leader Election Algorithm

- Leader Election in Anonymous Networks

- Our Leader Election Algorithm

## Verification

- Verification with  $\mu$ CRL and CADP

- Partial order reduction with  $\mu$ CRL

- Performance evaluation with PRISM

# (Deterministic) Leader Election

To break symmetry in a distributed system.

## Assumptions

- Each node has a unique identity
- The identities are ordered
- System is fully asynchronous

Elect a node with max (or min) identity.

## Definition

- The algorithm is decentralized
- Each node has the same local algorithm
- Upon termination one node is 'leader' and the rest are 'lost'.

# Franklin's Leader Election Algorithm

A undirected ring.

Each active node:

- sends its identity to its neighbors
- compares its identity with the nearest active neighbors ids
- if the identity is not the largest, node becomes passive

Passive nodes pass on messages

Repeat until node with the largest id receives its own message

Worst-case message complexity:  $O(n \log n)$ .

# Anonymous Networks

In some cases, nodes don't have (unique) identities:

## Example

- FireWire bus
  - to send ids can be costly
- Lego Mindstorms robots
  - CPUs don't have identities

## Assumptions

- Nodes are indistinguishable
  - No unique identities
  - Execute the same local algorithm
- Asynchronous communication

# Leader Election in Anonymous Networks

Impossibility results:

- The knowledge of network size is needed
- Election with deterministic algorithm is impossible

F. Fich, E. Ruppert, “Hundreds of impossibility results for distributed computing”, *Distributed Computing*, 16(2-3):121–163, 2003.

Probabilistic algorithms can be used

- Node can pick random identity
- Message with a hop counter to detect its source.

# Itai-Rodeh Election Algorithm

Based on Chang-Roberts algorithm.

A directed ring. All nodes know the ring size  $n$ .

A node selects a random identity, and sends it out

- When  $u$  receives  $v$ 
  - if  $u < v$ ,  $u$  becomes passive and passes on  $v$
  - if  $u > v$ ,  $u$  purges the message
- When  $u$  receives  $u$ 
  - if hop counter is  $n$ , it becomes the leader
  - otherwise, passes on the message with a 'dirty' bit

If several nodes picked the same largest identity

- they start a new election round

Round number is a part of:

- the state of a node
- message

# Probabilistic Finite-State Leader Election Algorithm

Itai-Rodeh algorithm with rounds is infinite-state algorithm.

- Avoid round numbers using FIFO channels

Our contribution

- use undirected ring with round numbers modulo 2

## Assumptions

- An anonymous, undirected ring
- Asynchronous communication
- Message order is not preserved between any pair of nodes
- Reliable channels
- Fair scheduler for message queues
  - Sent message will eventually be processed at its destination.
- All nodes know the ring size  $n$ .

# Our Algorithm

Each node is either active, passive or leader

An active node maintains two parameters:

- identity, not necessarily unique
- the number of the current election round (modulo 2)

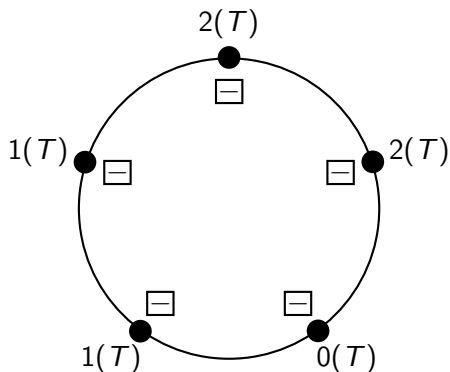
All messages are

- of the form  $\langle id, hop, bit \rangle$ 
  - $id$  is the originator identity
  - $bit$  is the election round of the owner (modulo 2)
  - $hop \leq n$  is hop counter, used to detect identity clashes
- travelling in both directions

Passive nodes simply pass on messages

- increasing hop counter by one

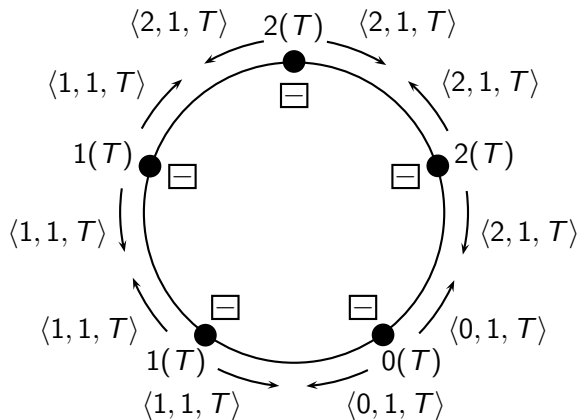
# The Algorithm



Initially, all nodes are active and their round number  $bit = T$ .

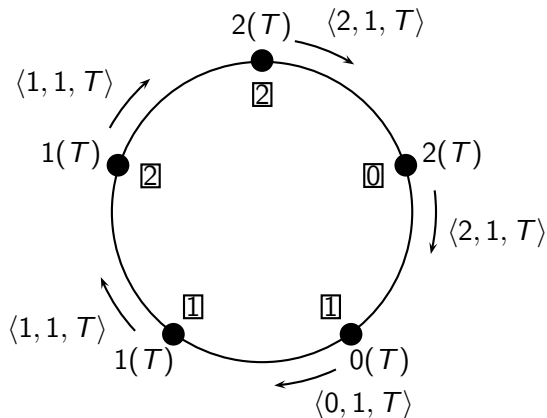
At the start of a round an active node picks a random identity

# The Algorithm



and sends the message  $\langle id, 1, bit \rangle$  in both directions

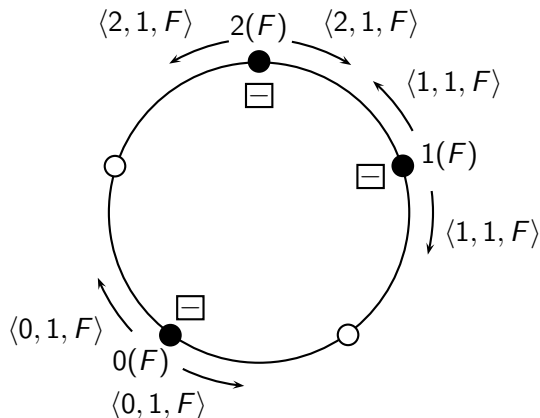
# The Algorithm



Upon receipt of a message  $\langle id, hop < n, bit \rangle$ , an *active* node

- stores it, and
- waits for a message from the other direction

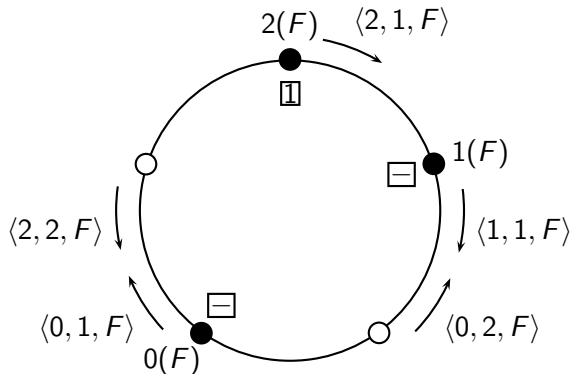
# The Algorithm



Upon receipt of messages from both sides, an *active* node

- becomes passive, if any of the ids is larger than its own
- otherwise, it starts a new election round with an inverted round number and a new identity

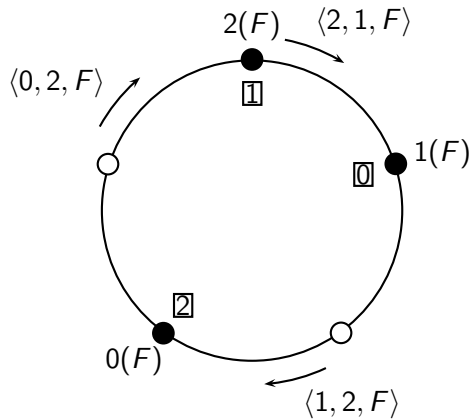
# The Algorithm



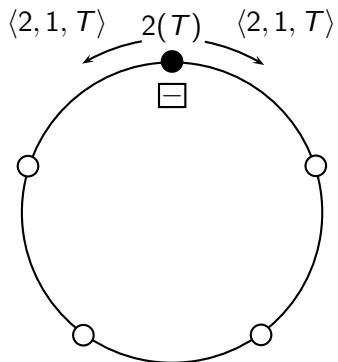
Upon receipt of a message  $\langle id, hop, bit \rangle$

- a *passive* node passes on a message  $\langle id, hop + 1, bit \rangle$

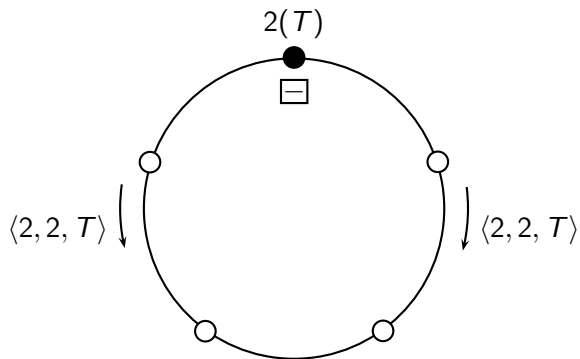
# The Algorithm



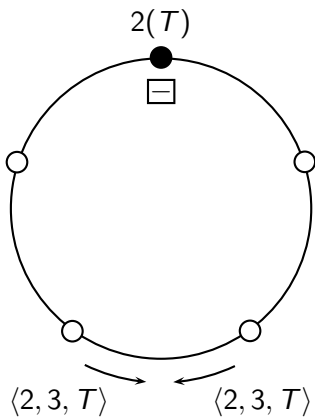
# The Algorithm



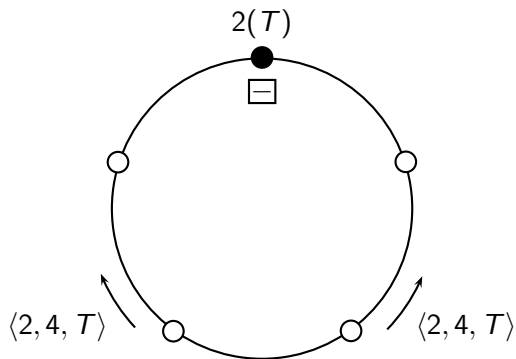
# The Algorithm



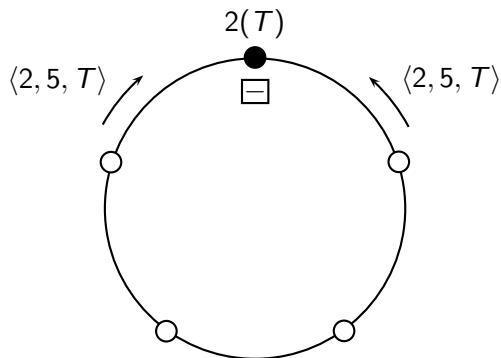
# The Algorithm



# The Algorithm



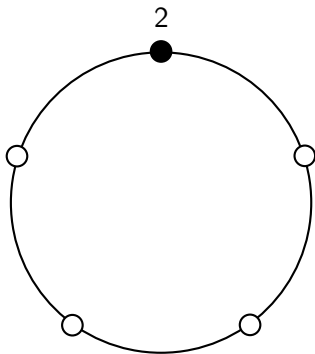
# The Algorithm



Upon receipt of a message  $\langle id, hop = n, bit \rangle$

- an *active* node becomes the leader

# The Algorithm



# Why Round Numbers?

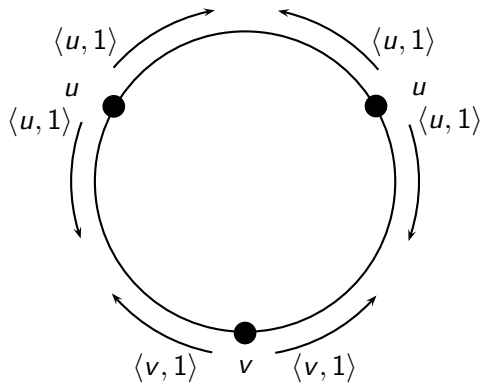
When channels are FIFO, round numbers are not needed.

For instance, shown for Itai-Rodeh leader election algorithm

- Wan Fokkink and Jun Pang “Variations on Itai-Rodeh Leader Election for Anonymous Rings”,  
J. of Universal Computer Science, 12(8):981–1006, 2006.

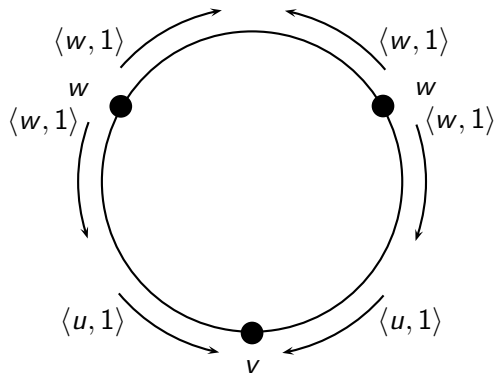
But for our algorithm, the message order is not preserved between any pair of nodes

# Why Round Numbers?



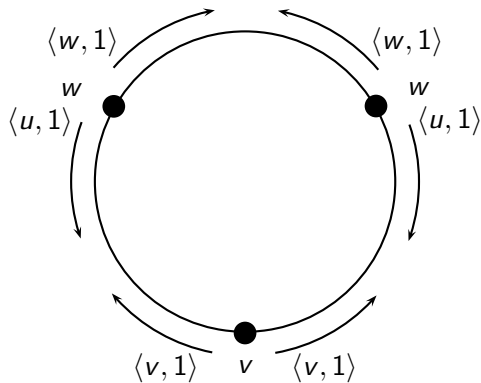
$$u > v > w$$

# Why Round Numbers?



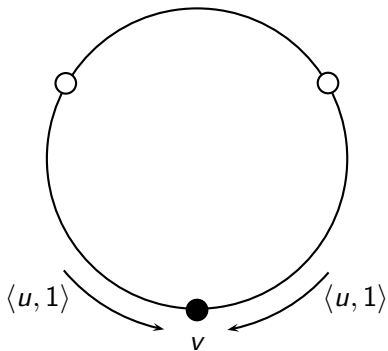
$$u > v > w$$

# Why Round Numbers?



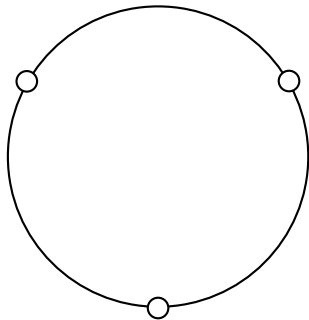
$$u > v > w$$

# Why Round Numbers?



$$u > v > w$$

# Why Round Numbers?



$$u > v > w$$

# Why Undirected Algorithm?

Dolev, Klawe and Rodeh (also Peterson) adapted Franklin's idea to a directed ring.

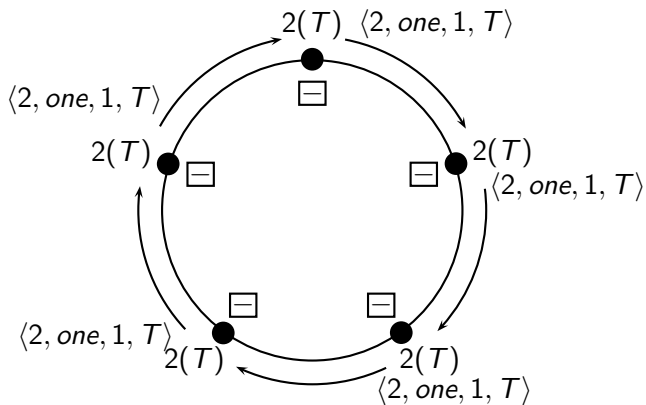
- An active node only progresses to the next election round if its identity is larger than the identities of two left active consecutive nodes.

In our probabilistic version of DKR algorithm

- identities are picked at random
- hop counts are used to decide the leader
- labels *one* and *two* are used to identify the message originator
- a round number modulo 2

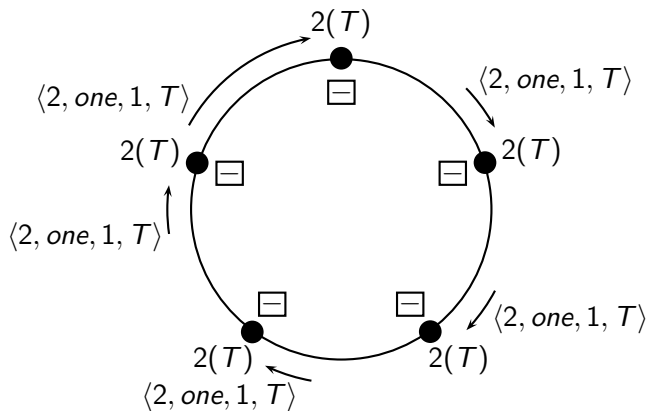
But round numbers modulo 2 are not sufficient for probabilistic DRK algorithm.

# Probabilistic Dolev-Klawe-Rodeh



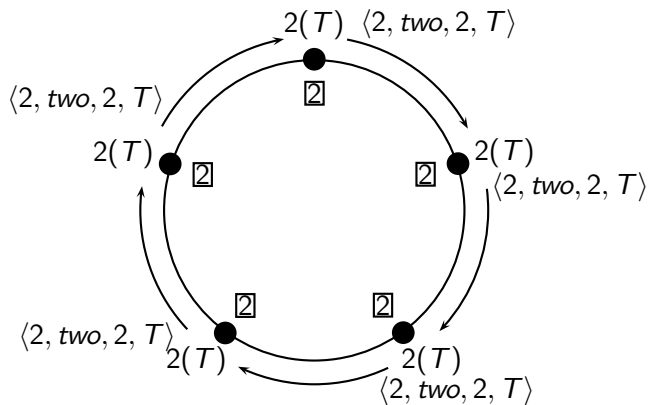
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



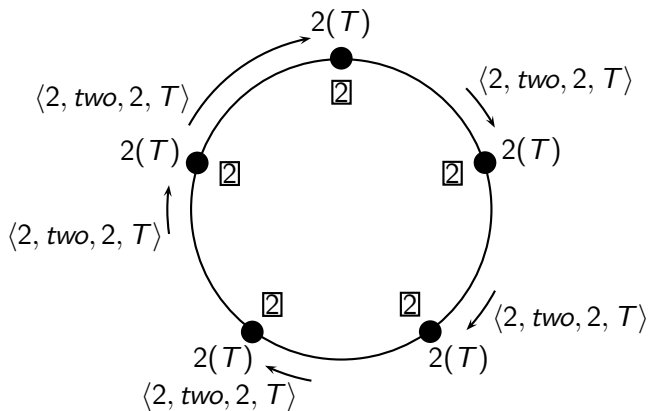
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



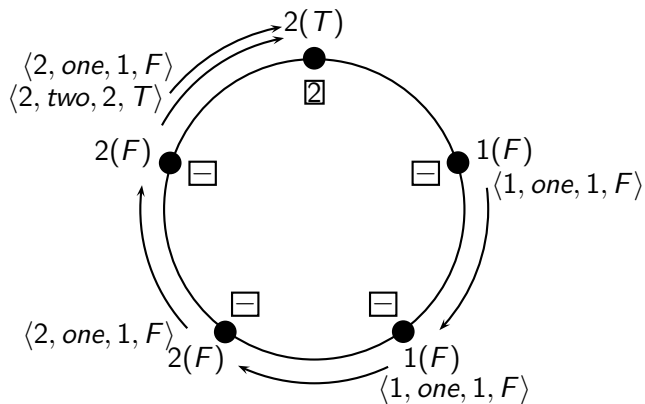
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



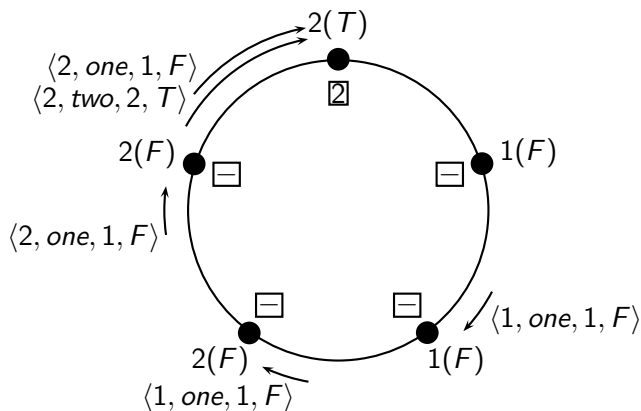
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



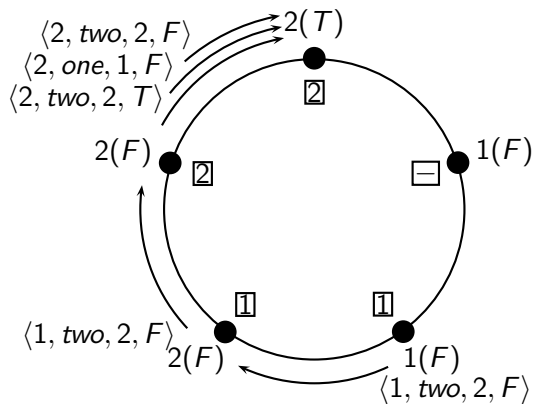
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



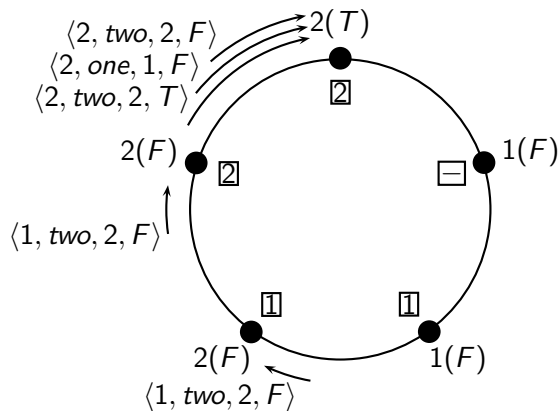
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



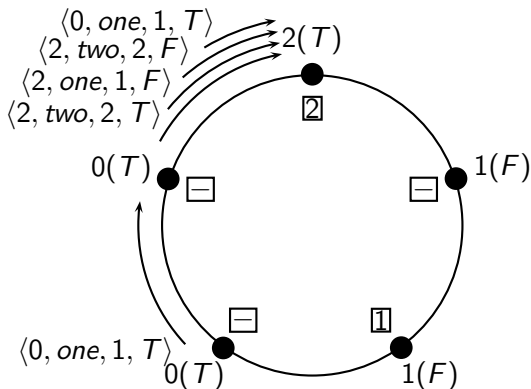
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



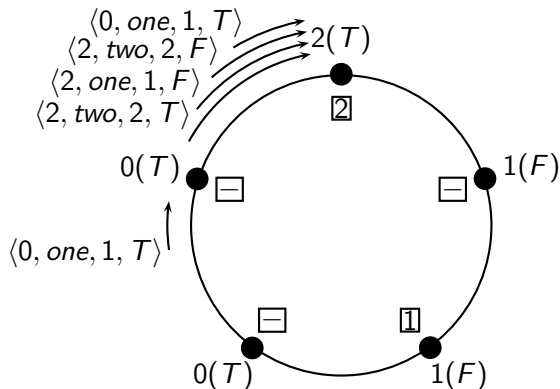
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



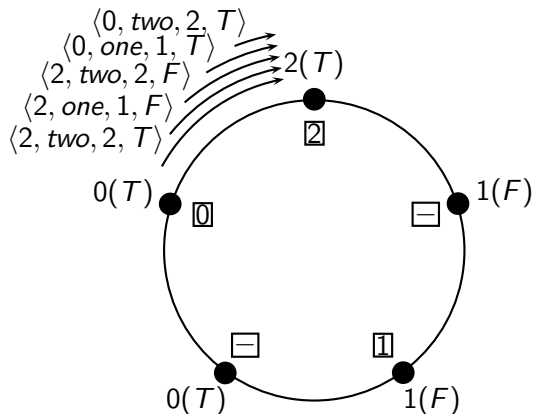
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



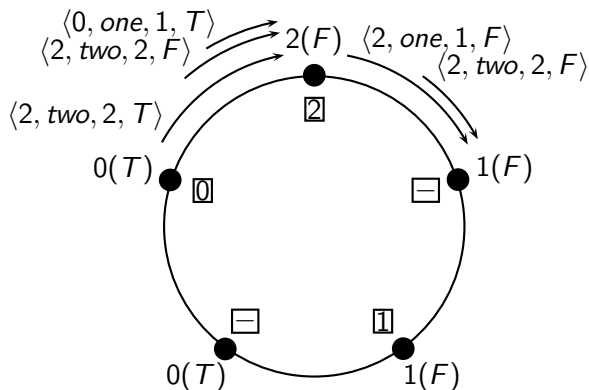
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



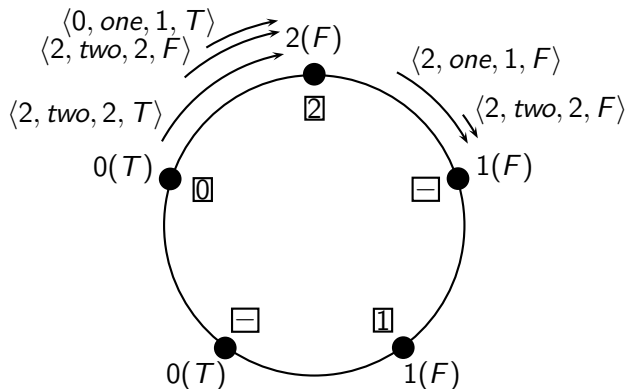
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



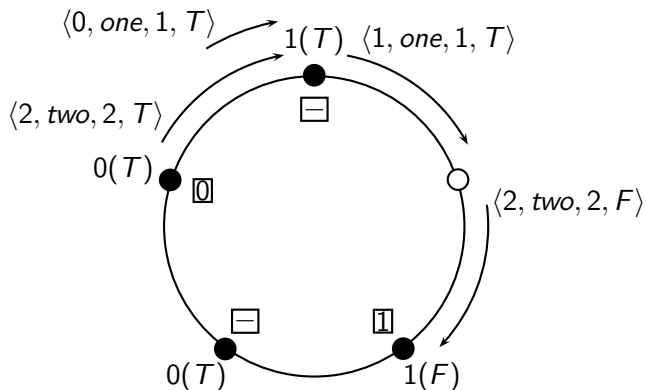
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



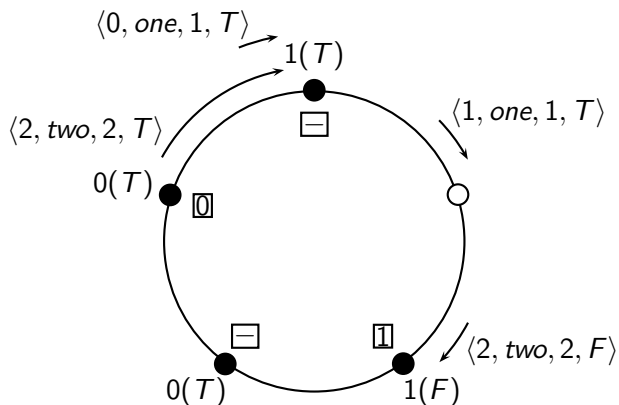
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



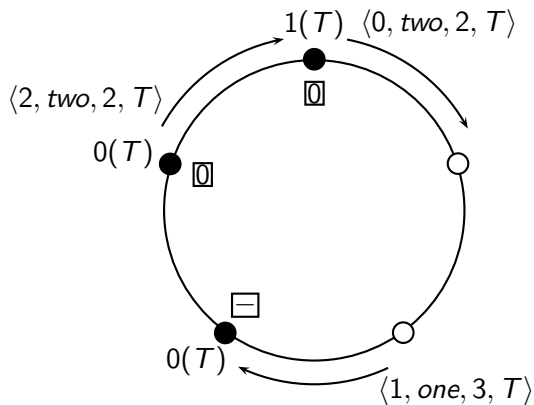
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



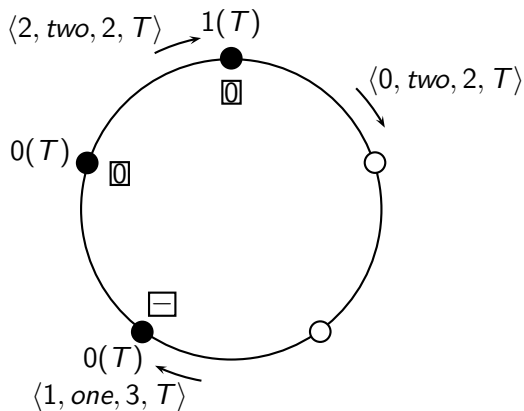
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



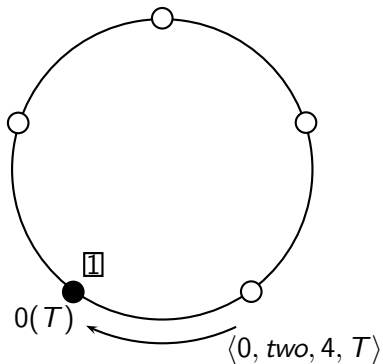
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



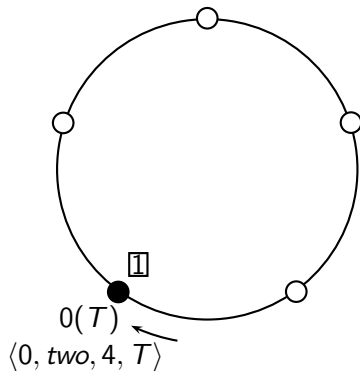
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



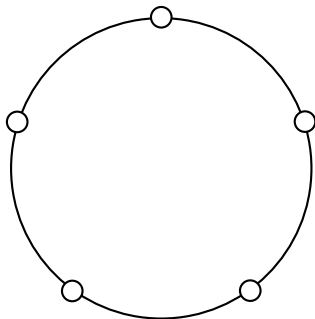
Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



Message  $\langle id, label, hop, round \rangle$

# Probabilistic Dolev-Klawe-Rodeh



Message  $\langle id, label, hop, round \rangle$

## Correctness

The probabilistic Franklin algorithm for anonymous undirected rings terminates with probability one, and upon termination a unique leader has been elected.

Shown using model checking analysis with  $\mu$ CRL and CADP

- up to ring size 6
- distributed  $\mu$ CRL used for 6 nodes (2 identities) and 5 nodes (3 identities)
- branching bisimulation equivalence abstracts away from infinite executions
- minimized state space of the algorithm has two states and “leader” action as transition

## Generated State Space Statistics

- State space for two identities

<b># Procs</b>	<b>States</b>	<b>Transitions</b>
2	657	1,368
3	15,445	43,968
4	380,609	1,396,512
5	9,819,065	44,242,920
6	260,753,105	1,393,967,976

- State space for three identities

<b># Procs</b>	<b>States</b>	<b>Transitions</b>
2	1,525	3,564
3	55,009	168,102
4	2,095,777	8,182,092
5	84,381,157	401,681,445

# Partial Order Reduction

To reduce the state space for model checking

- Each node first reads from the left and then from the right
- Static analysis
  - Elimination of constant node parameters (`constelm`)
- Finding and marking confluent summands (`confcheck`)
- Symbolic confluence reduction (`confelm`)
- On-the-fly  $\tau$ -reduction (`--confluence ctau` option with `instantiator`)

# Partial Order Reduction

- State space for two identities

Strategy	#	2	3	4	5	6
normal	s.	385	7,613	152,065	3,162,337	67,758,817
	t.	664	17,880	459,488	11,736,100	298,484,184
confelm	s.	205	2,875	40,881	606,783	9,280,633
	t.	340	6,342	114,384	2,069,040	37,381,488
on-the-fly	ext. s.	165	1,819	21,409	263,963	3,348,345
	int. s.	181	2,343	30,039	395,723	5,350,021
	t.	276	4,086	60,576	902,820	13,449,324

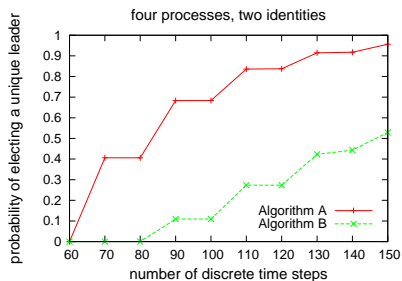
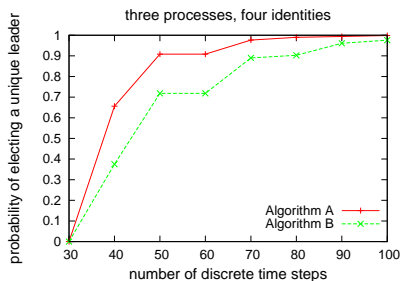
- State space for three identities

Strategy	#	2	3	4	5
normal	s.	877	26,299	802,489	25,919,965
	t.	1,680	65,853	2,560,848	100,868,445
confelm	s.	469	9,874	214,957	4,952,449
	t.	876	23,310	637,884	17,778,660
on-the-fly	ext. s.	385	6,400	116,785	2,242,609
	int. s.	433	8,518	170,131	3,524,305
	t.	732	15,570	353,508	8,137,080

# Performance Evaluation with PRISM

An active node chooses a fresh identity at the start of:

- A: each election round.
- B: new election round only if an identity clash is detected.



Thank you!

Thank you for your attention!

Questions?