Proving in Coq (Pactole): A user perspective

David Ilcinkas
Joint work with Sébastien Bouchard

CNRS, Bordeaux, France

Descartes/Estate/BAD
November 9, 2021

Big (french) projects about DC in Coq

PADEC

- Dedicated to self-stabilizing algorithms
- Complex algorithms
- Precise analyses, including time complexities
- K. Altisen, P. Corbineau, S. Devismes

PACTOLE

- Dedicated to distributed computing by mobile robots
- Large variety of models and parameters
- **Positive** and **negative** results
- T. Balabonski, P. Courtieu, L. Rieg, S. Tixeuil, X. Urbain, ...

Problem: Ring exploration with stop

Model/context

- Team of robots
  - sensing the environment by taking a snapshot of it
  - that do not communicate
  - that are anonymous and oblivious
- Anonymous unoriented rings.

Goal: exploration with stop

- Each node must be visited by at least one robot.
- All robots must stop after finite time.

The Look-Compute-Move cycle

Look

The robot takes an instantaneous egocentric snapshot of the network and its robots, with multiplicity detection ("zero", "one", or "more than one" robots).

Compute

Based on this observation, it decides to stay idle or to move to some neighbouring node.

Move

In the latter case it instantaneously moves towards its destination.

Identical oblivious asynchronous robots

Identical

Robots have no IDs. They execute the same program.

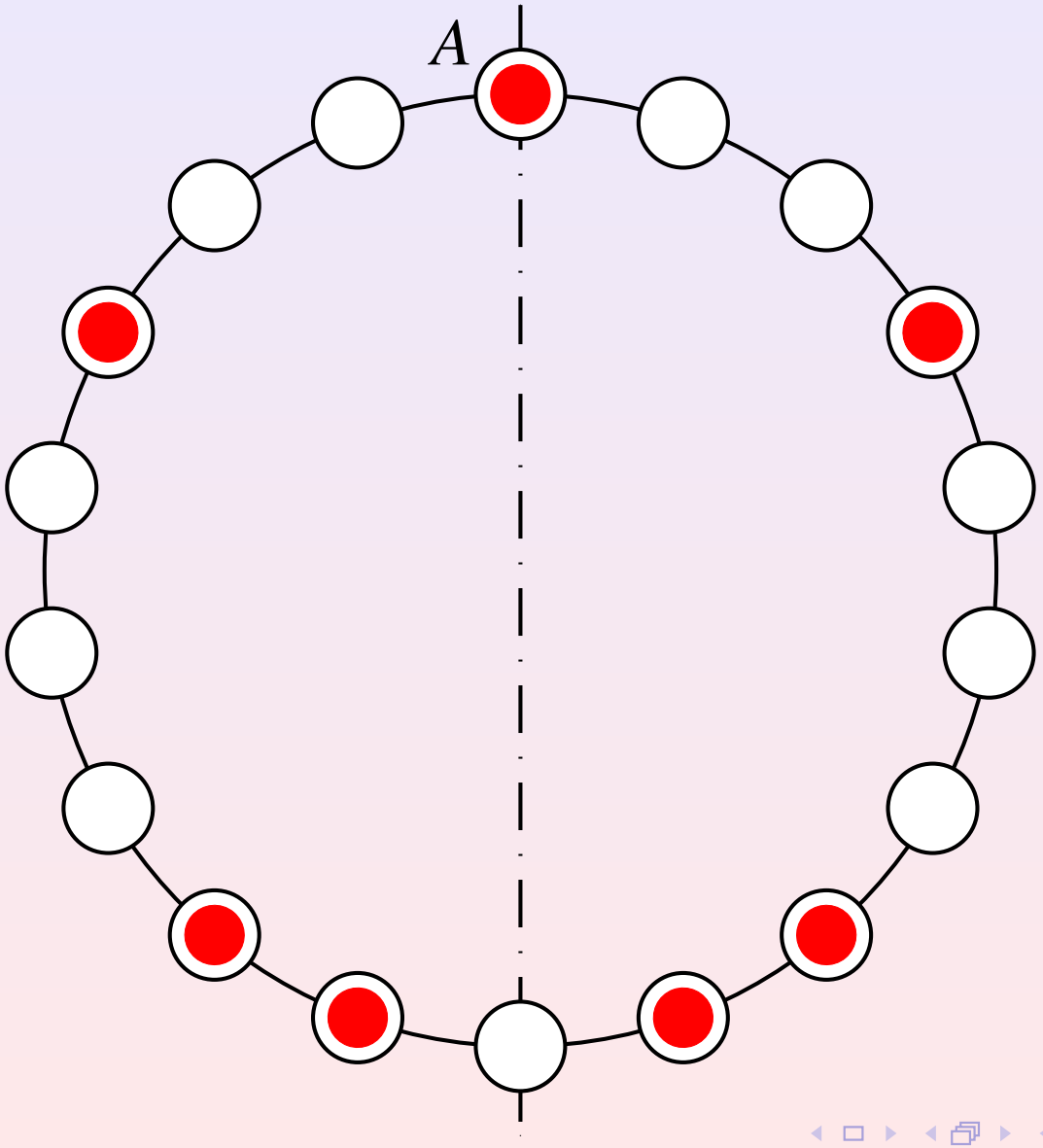
Oblivious

The robots have no memory of observations, computations and moves made in previous cycles.

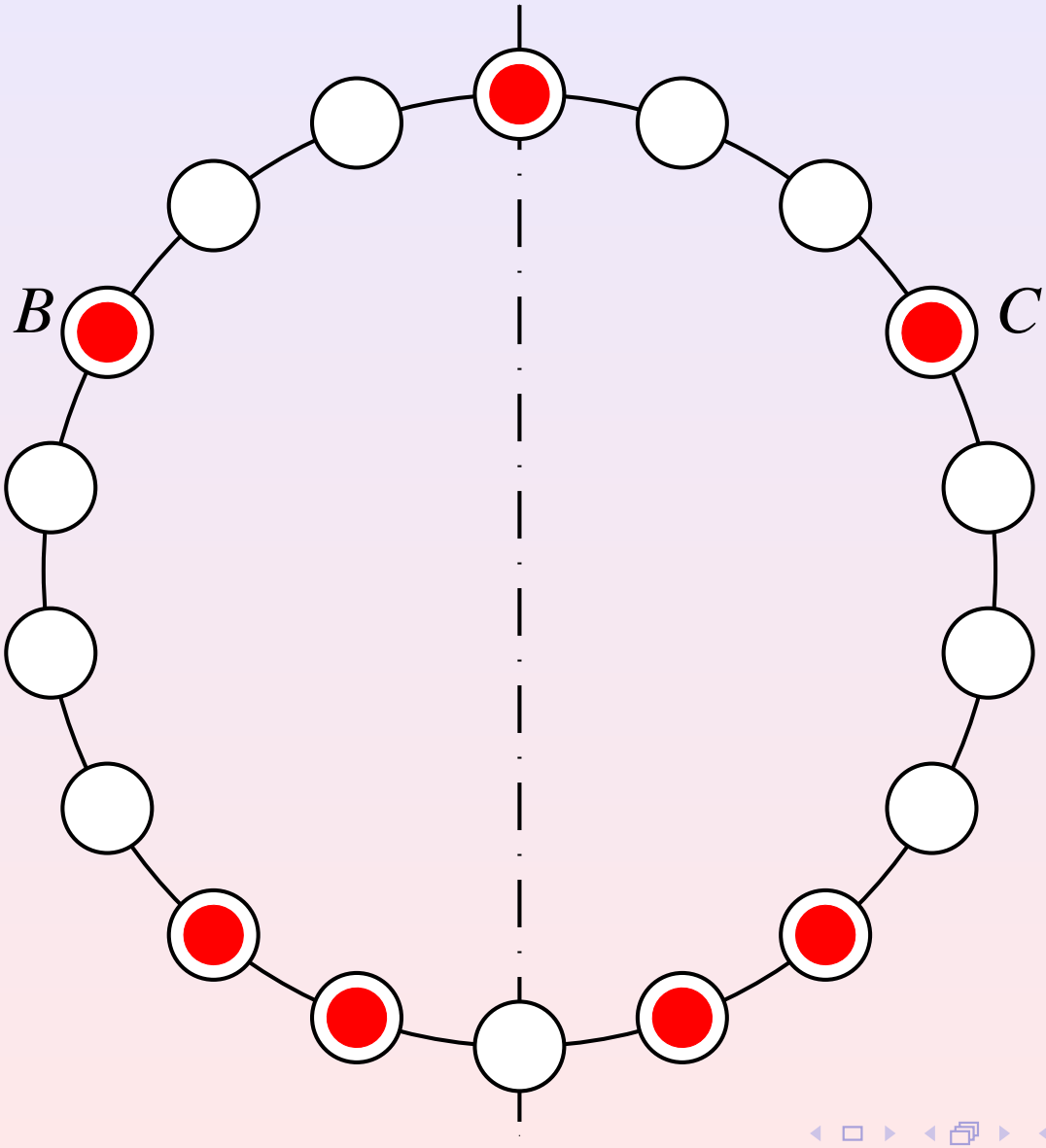
Asynchronous

The time between Look, Compute, and Move operations is finite but unbounded.

Some precisions



Some precisions



Problem definitions

Goal: exploration with stop

- Each node must be visited by at least one robot.
- All robots must stop after finite time.

Definition: Explo(k, n)

We say that exploration of a n-node ring is possible with k robots if there exists an algorithm enabling the robots to perform exploration with stop starting from any initial configuration of the k robots without multiplicity (at most one robot per node).

More formal definition: Explo(k, n)

exists Algo, forall Adv, Config,
if is_cycle(n,Config) and has_robots(k,Config) and is_flat(Config)
then exploStop(Algo, Adv, Config)

Results in PACTOLE

Already in PACTOLE

- not Explo($\mathbf{1}, n$)
- not Explo(k, n) **if** k **divides** n

Our "new" result

for every positive integer m,
not Explo(k, n) $\Longrightarrow$ not Explo($k * m, n * m$)

Results in PACTOLE

Already in PACTOLE

- not Explo($\mathbf{1}, n$)
- not Explo(k, n) **if** k **divides** n

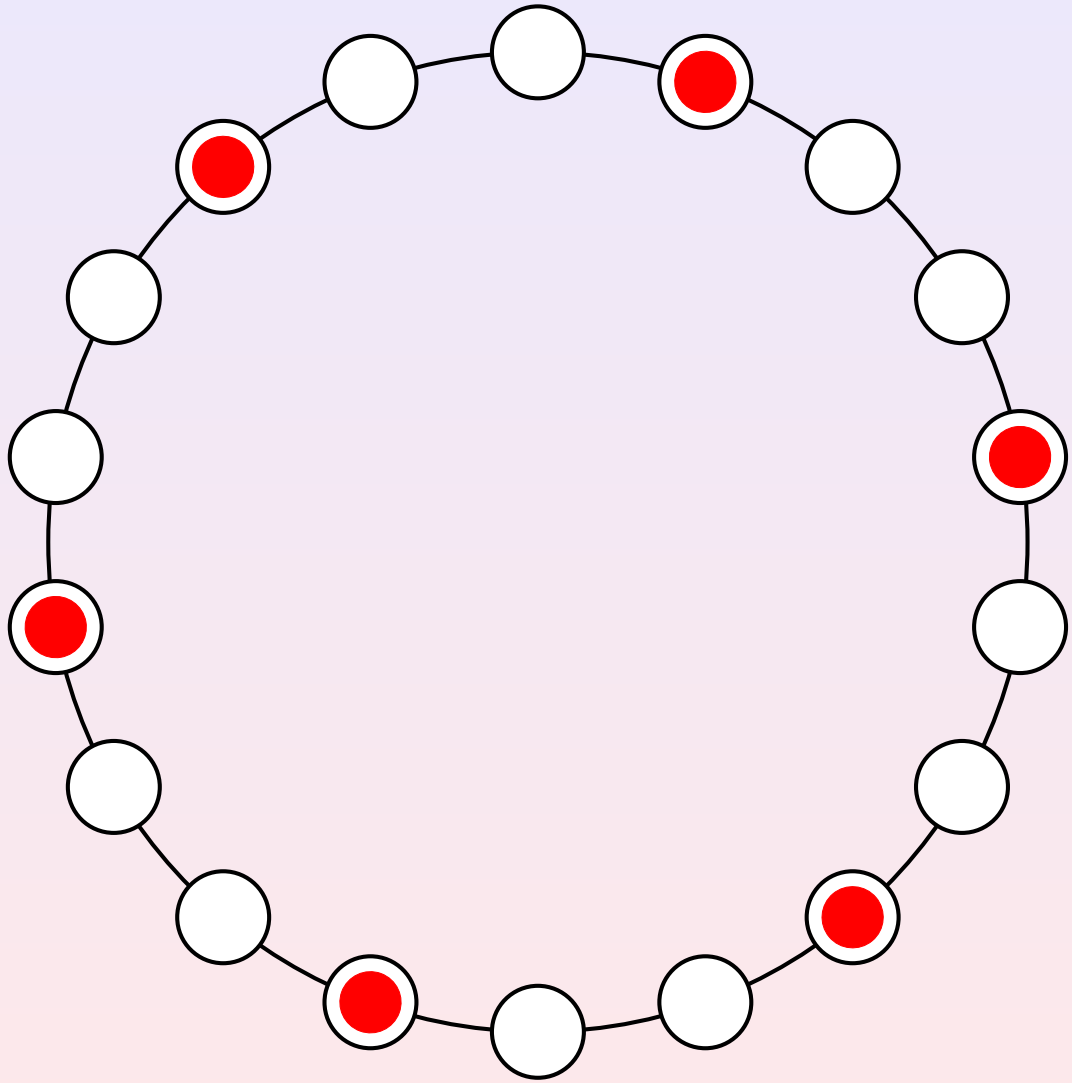
Our "new" result

for every positive integer m,
not Explo(k, n) $\Longrightarrow$ not Explo($k * m, n * m$)

When k divides n

Lemma

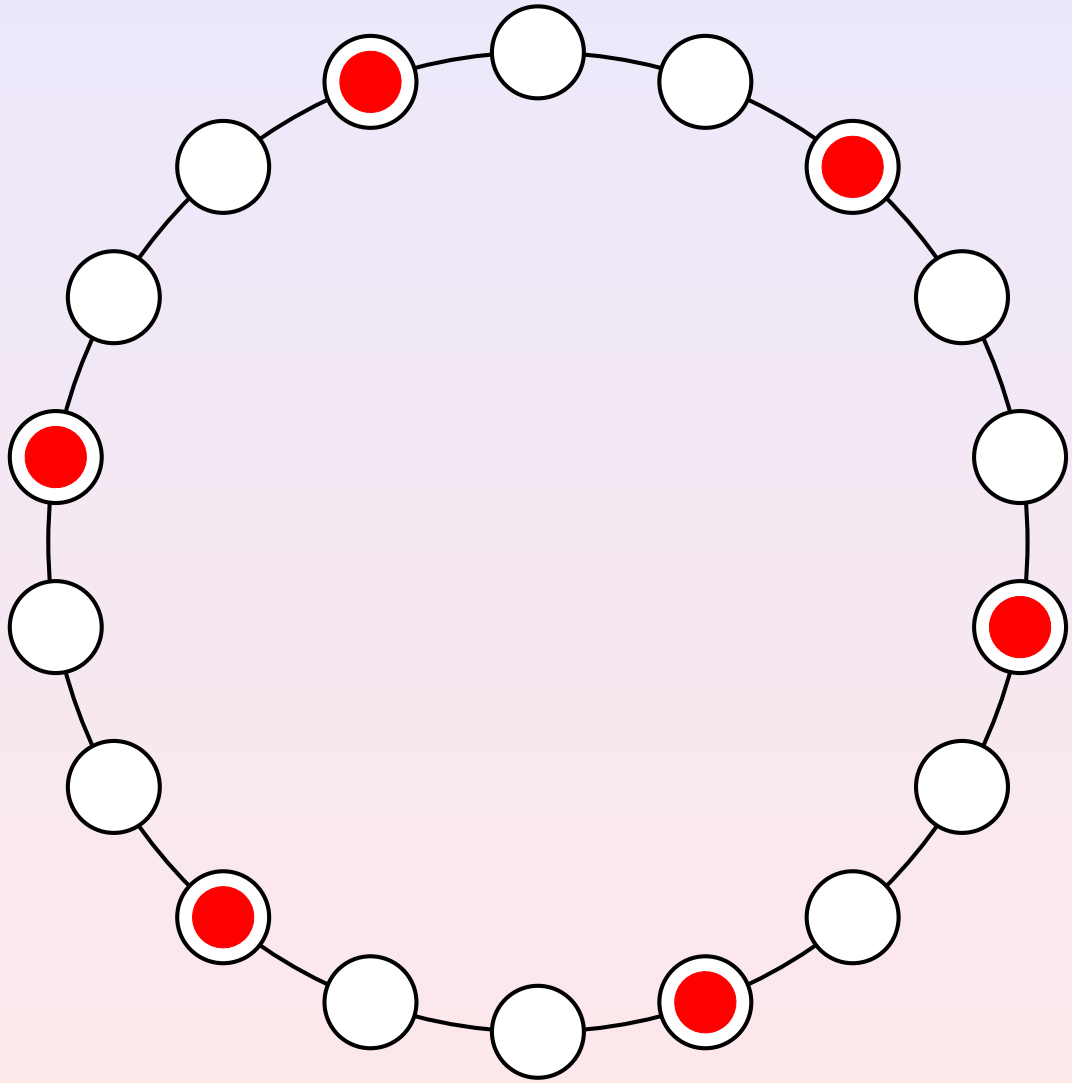
Impossible to stop (and sometimes to explore) when $k|n$.



When k divides n

Lemma

Impossible to stop (and sometimes to explore) when $k|n$.



Sketch of proof

Proved via: $\text{Explo}(k * m, n * m) \Longrightarrow \text{Explo}(k, n)$

Transformations from small _1 to big _m

- From _1 to _m: $i \rightarrow \{n \text{ or } k\} * j + i$, for $0 \leq j < m$
- From _m to _1: $i \rightarrow i$ modulo $\{n \text{ or } k\}$