Randomized Dining Philosophers

Presented by Xin Zhang

YORK
UNIVERSITÉ
UNIVERSITY

Outline

Introduction

Safety property-mutual exclusion

- Semaphore
- Monitor and condition
- Message passing...

Deadlock free-make progress

When some philosophers are hungry, then there will be at least philosopher to eat from then on.

Problem of These Models

More information than necessary

- Semaphore needs to know not only your neighbor, but your neighbor's neighbor;
- Monitor needs to know all processes;
- Message monitor needs to know the number of philosophers, and synchronization

Behavior differently

- Philosphers need to pick up forks in different order
- Id is needed for each philosopher

What We really Want

- Safety property
- Deadlock free

What We really Want

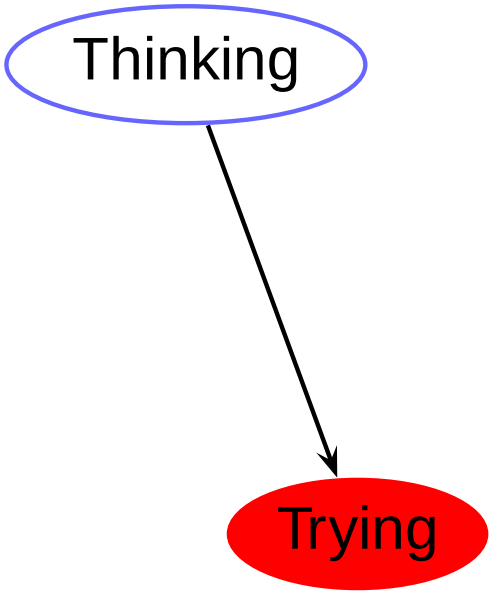
- Safety property
- Deadlock free
- Liveness property: starvation free
- Truly distributed: no central memory or central process
- Symmetric: identical, no id

Randomized Dining Philosophers Algorithm

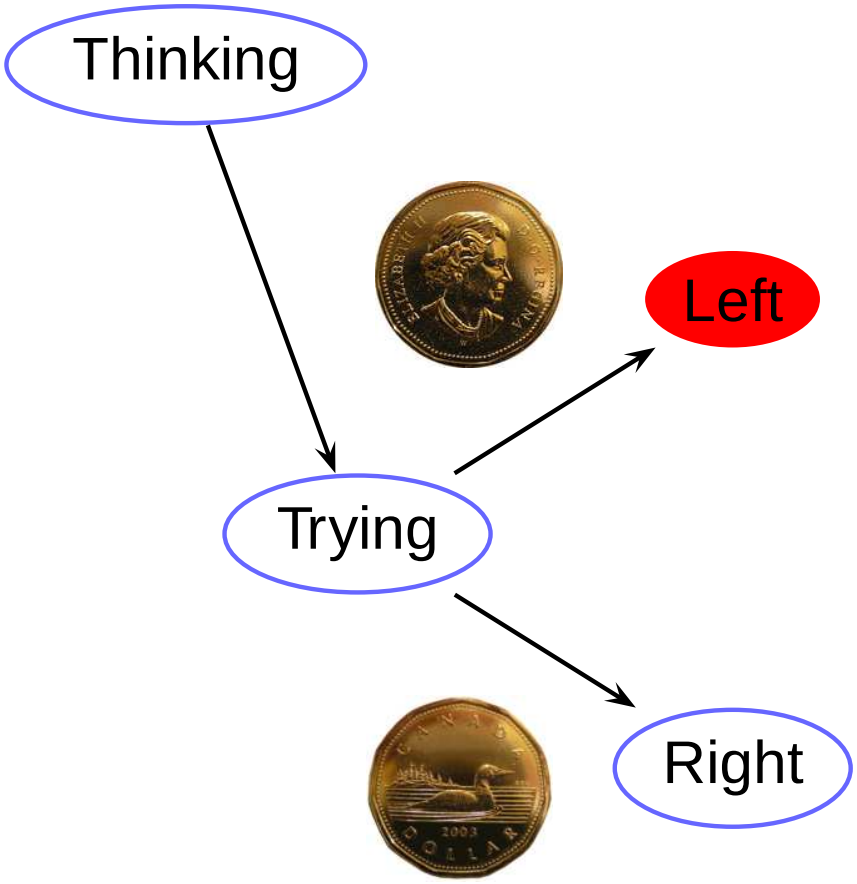
- Proposed by Daniel Lehmann & Michael O. Rabin
- Use probabilistic algorithm to randomly choose which fork to pick up first
- Wait for first fork
- Then for second, if can’t, give up the first one and retry

Control Flow

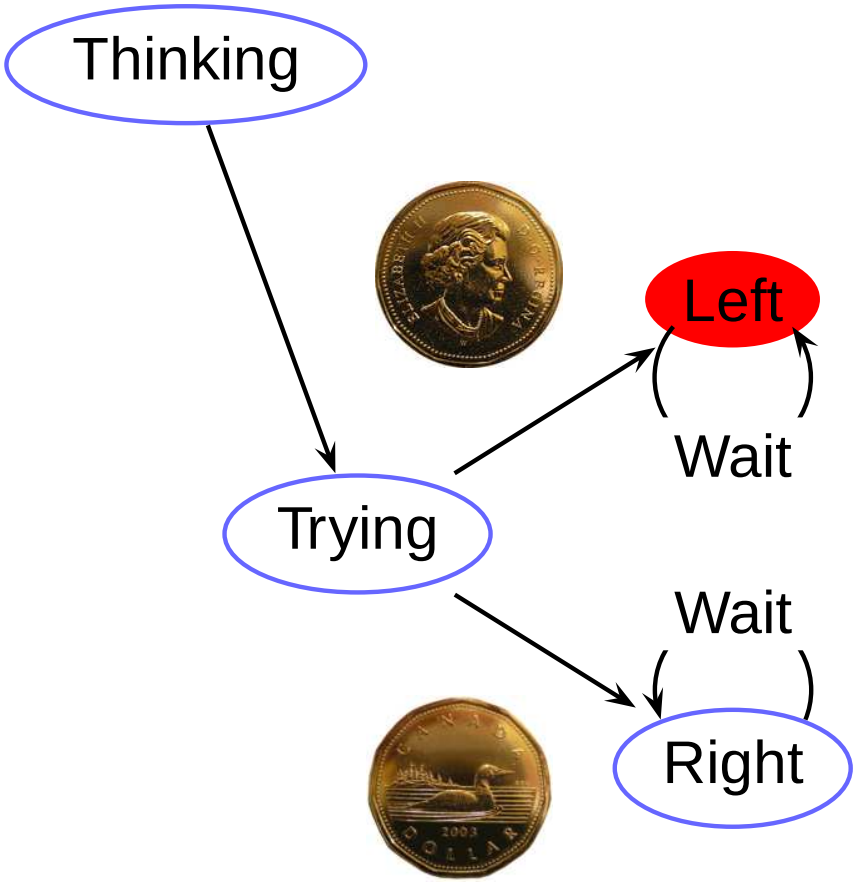
Control Flow



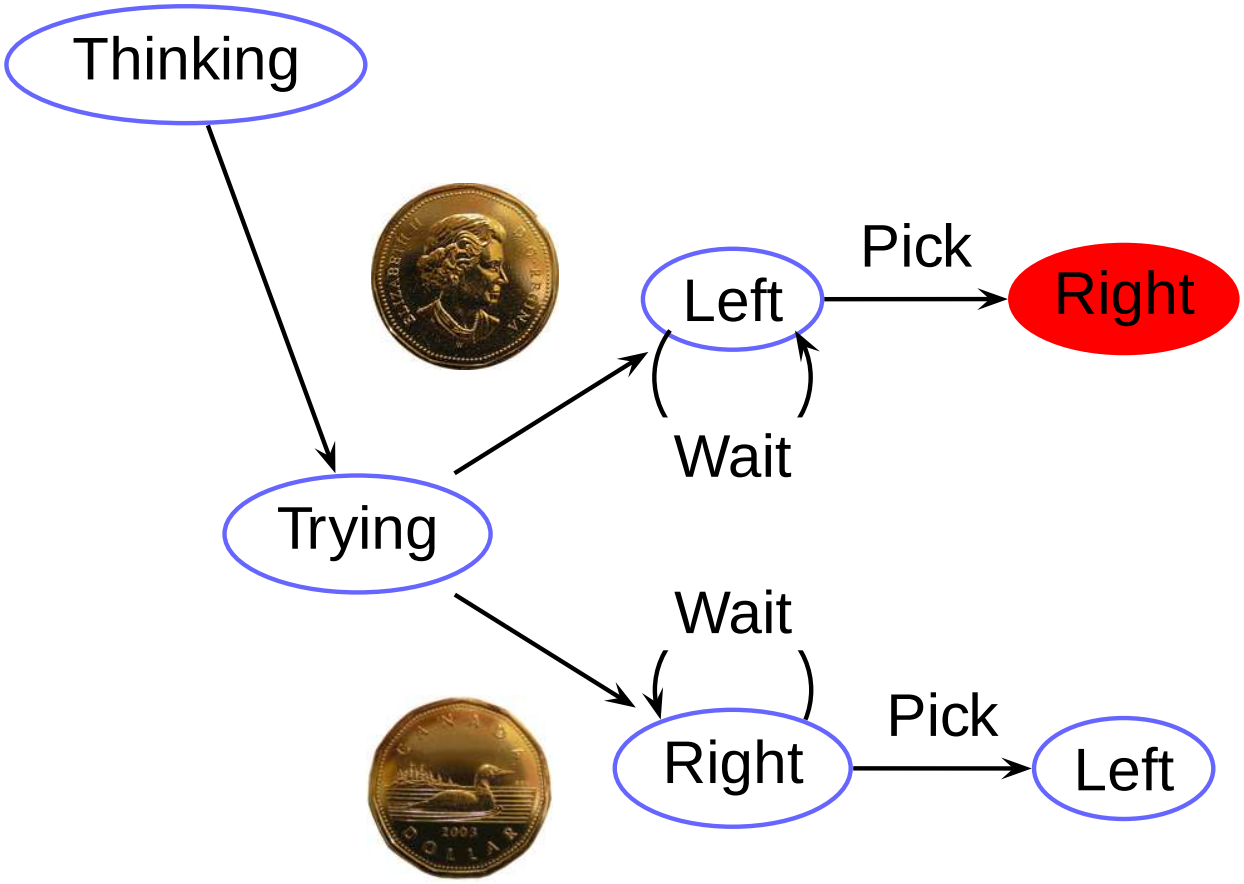
Control Flow



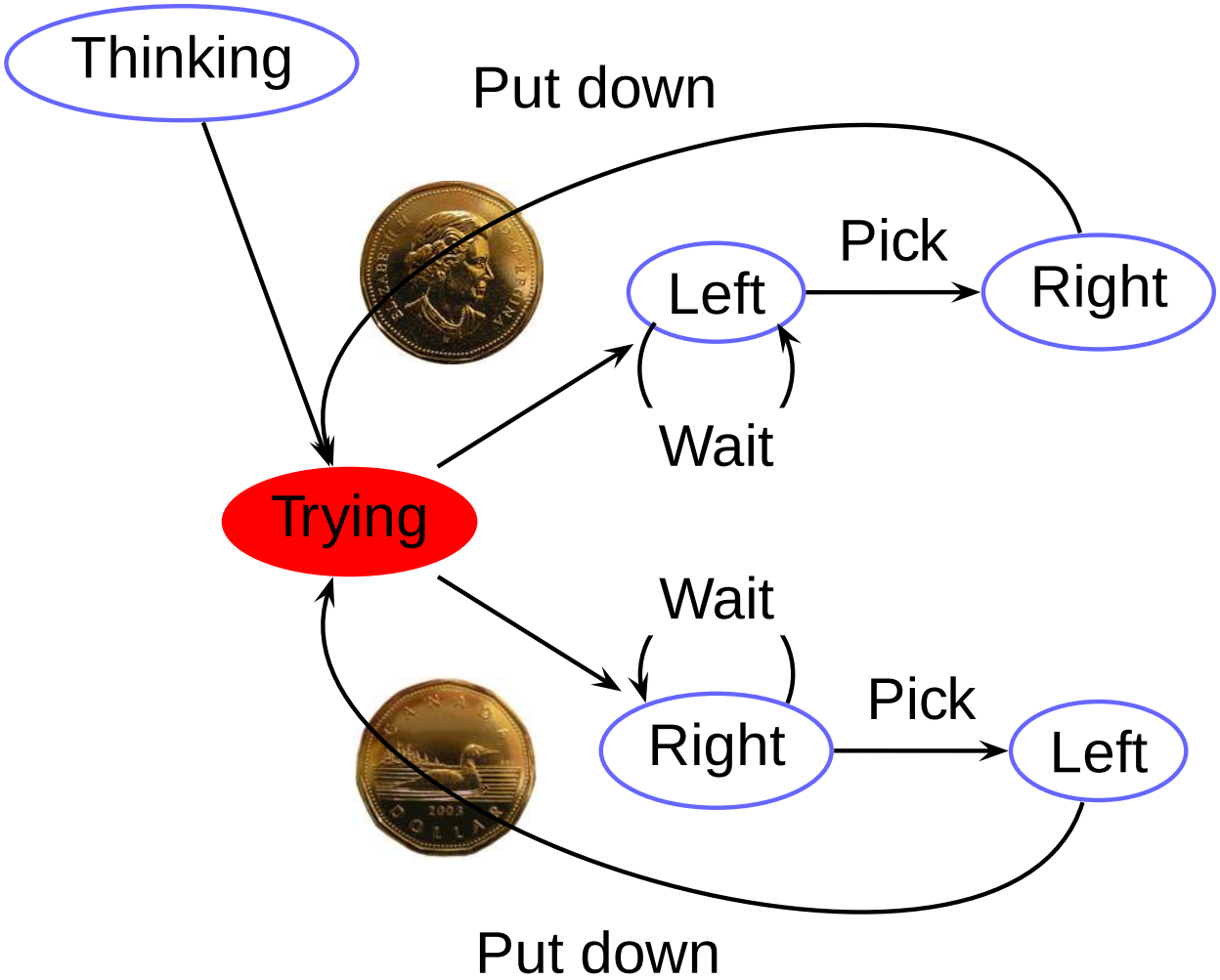
Control Flow



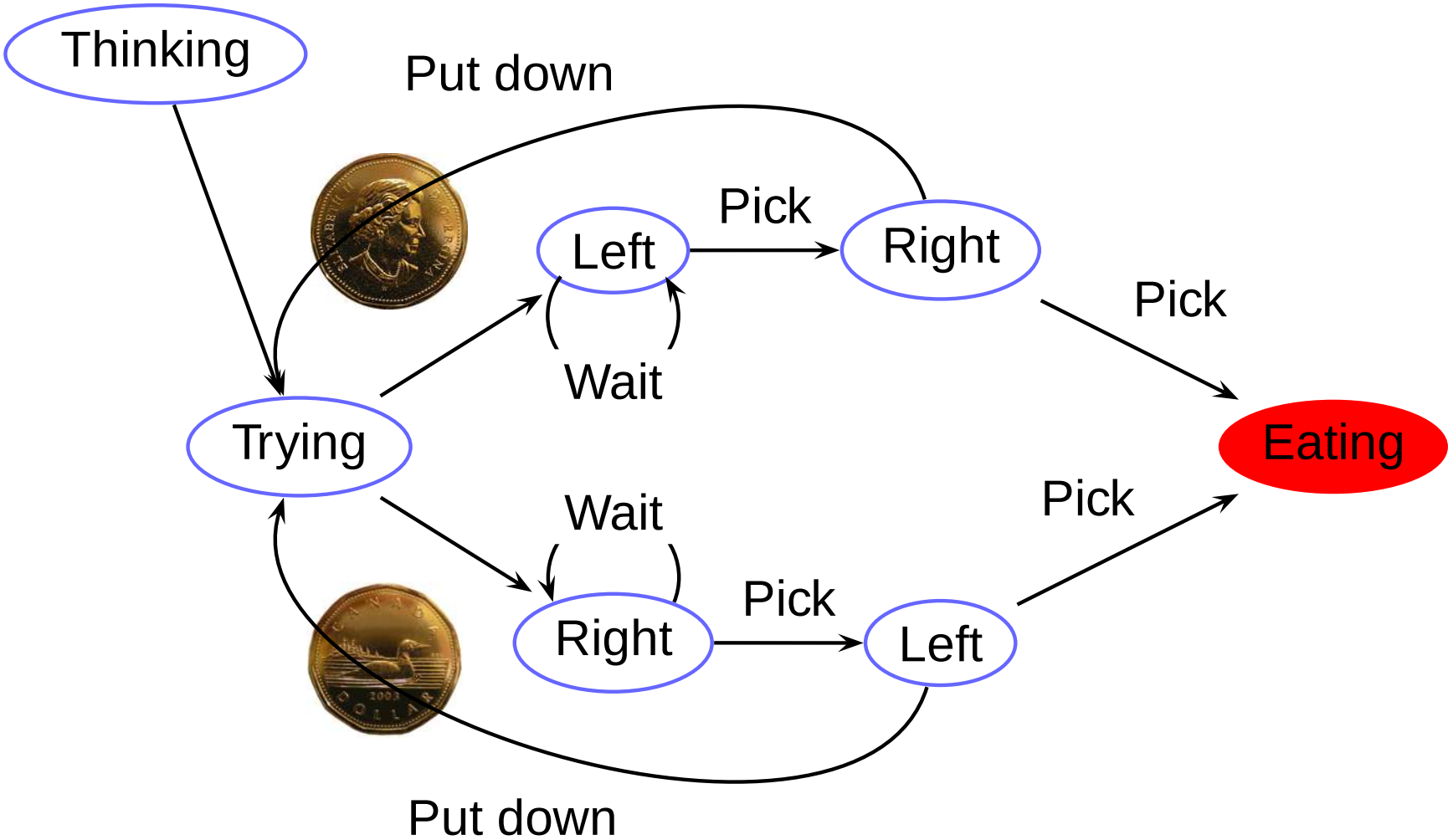
Control Flow



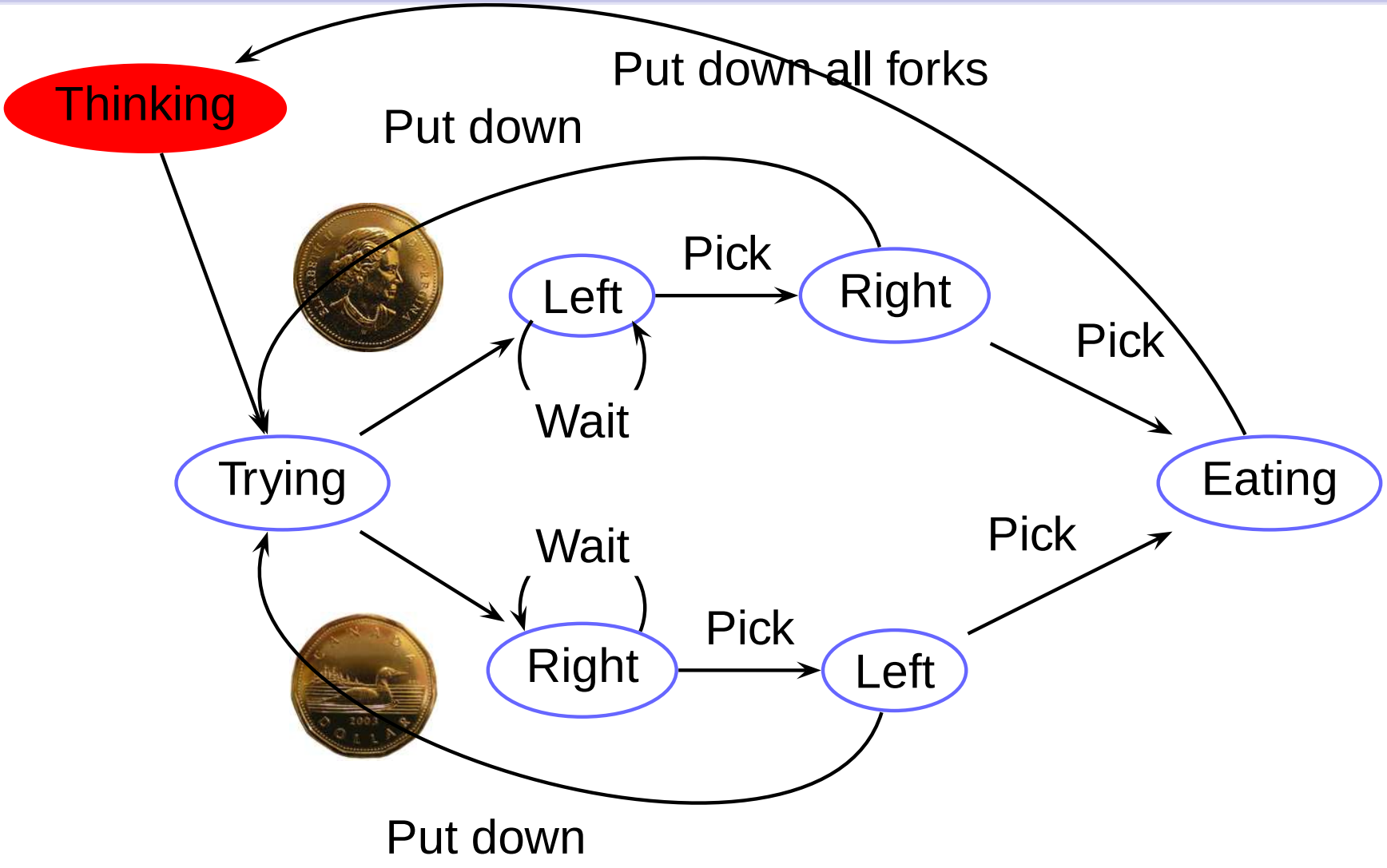
Control Flow



Control Flow



Control Flow



Improvement

- How about livenenss property?

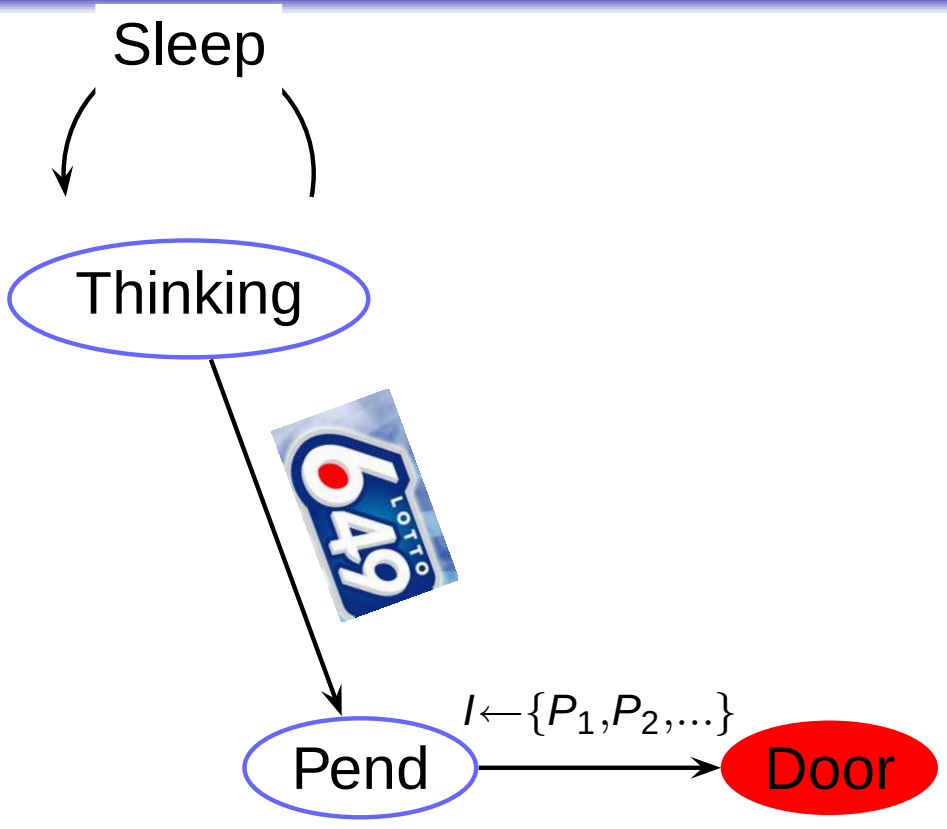
Improvement

- How about livenenss property?
- Add one more variable to indicate if the neighbor ate or not when pick the first fork
- Alternatively, use “Doorway Concept” and message passing between the philospher and neighbor

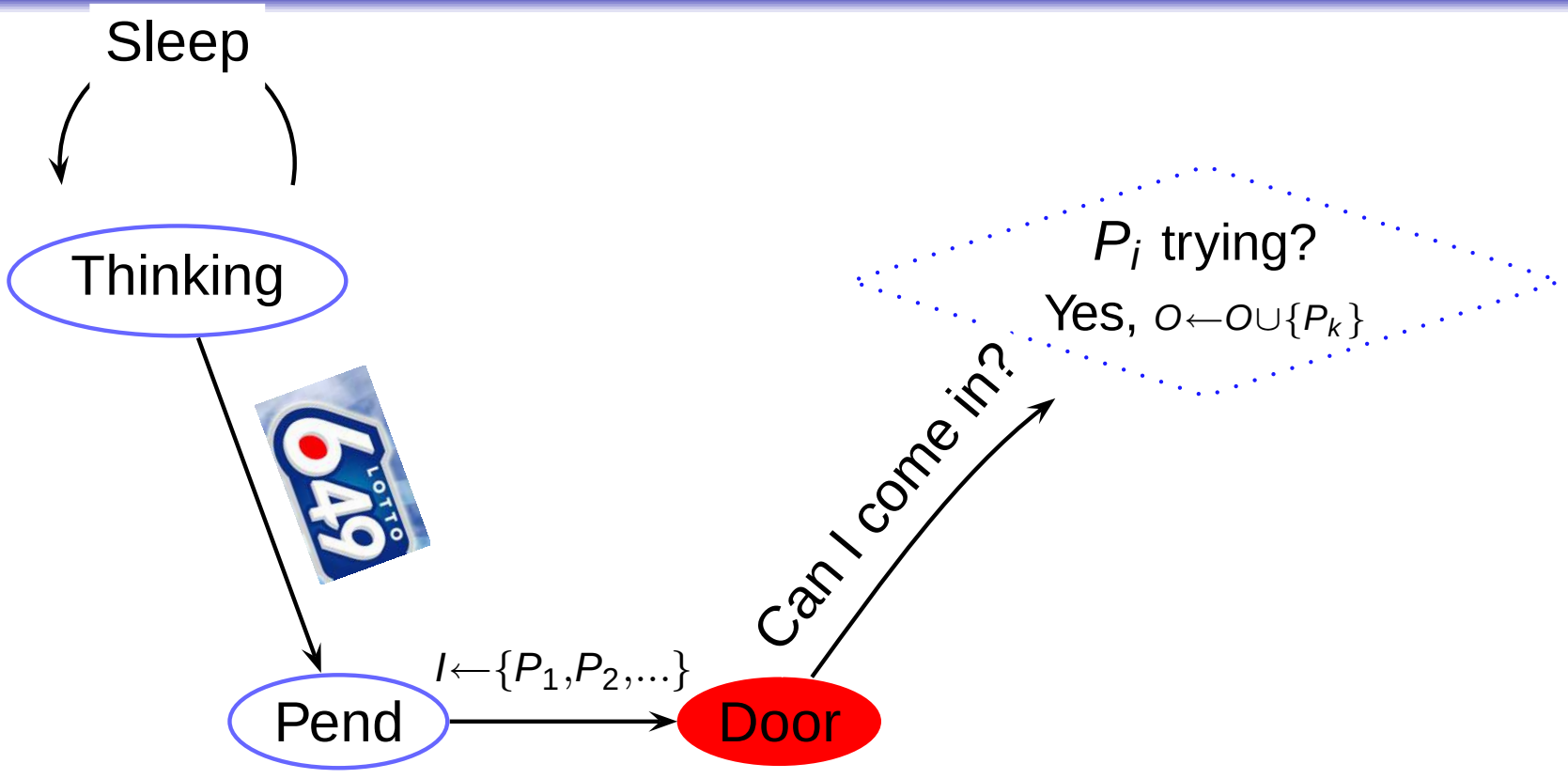
Control Flow-Message Passing:P_k

Control Flow-Message Passing:P_k

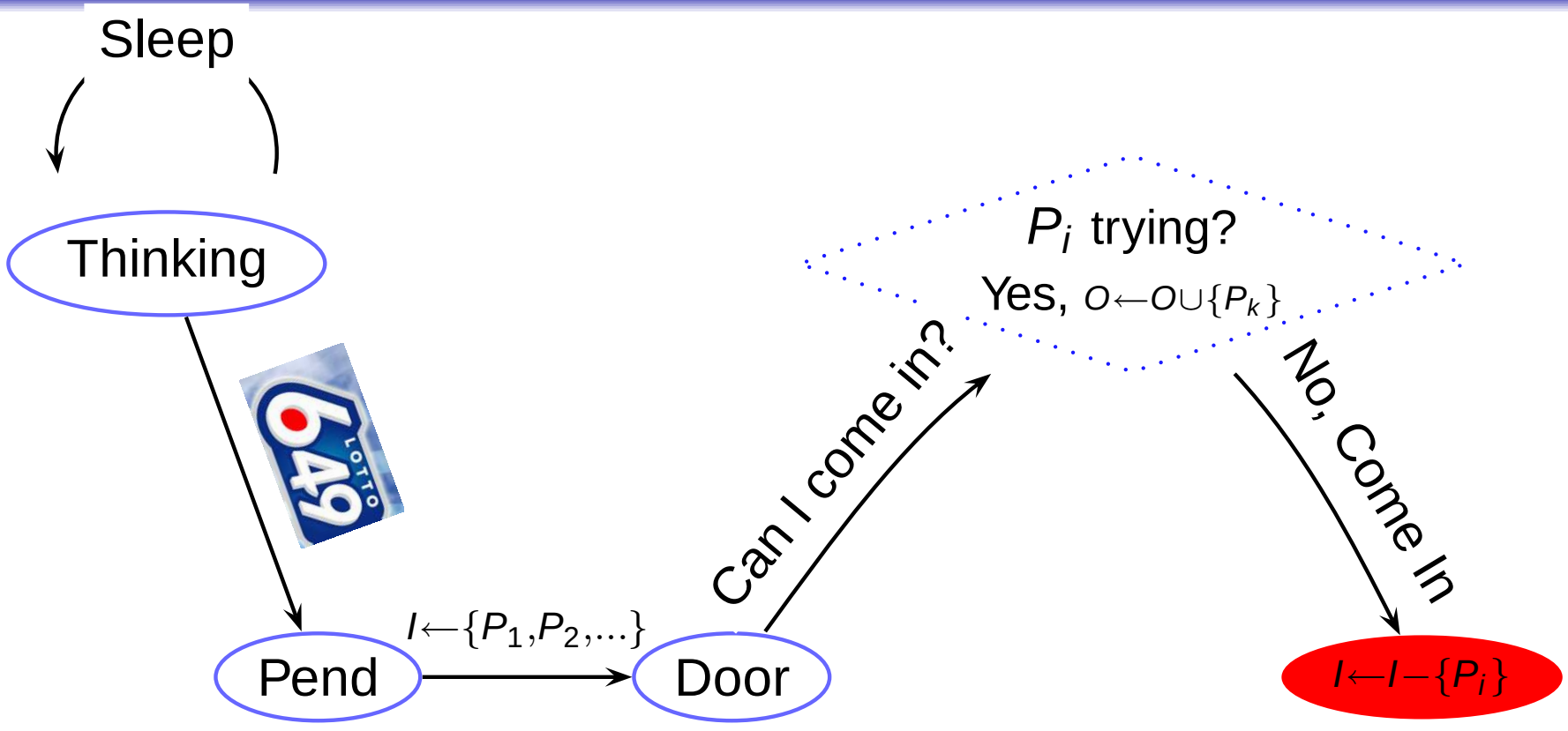
Control Flow-Message Passing:P_k



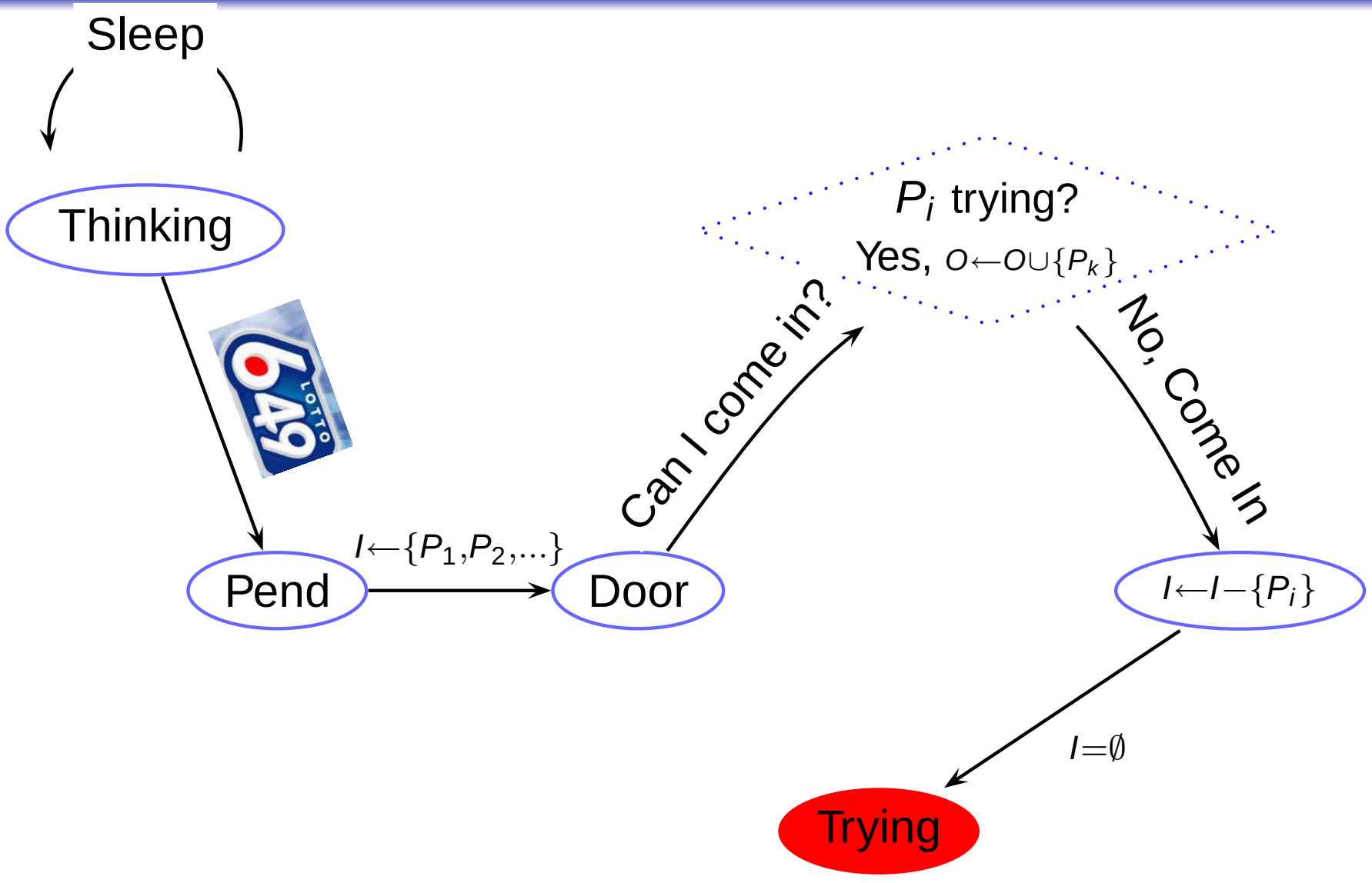
Control Flow-Message Passing:P_k



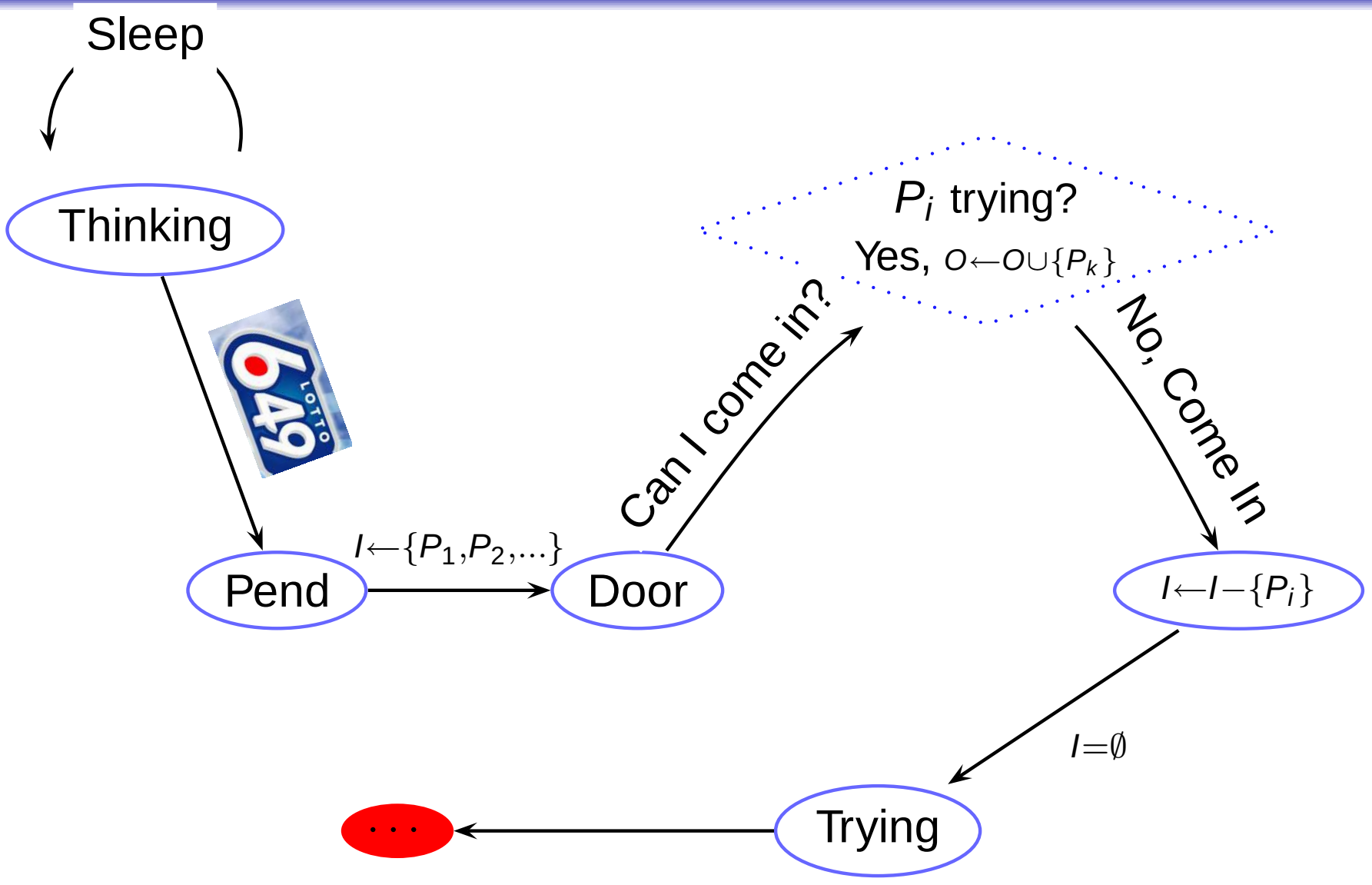
Control Flow-Message Passing:P_k



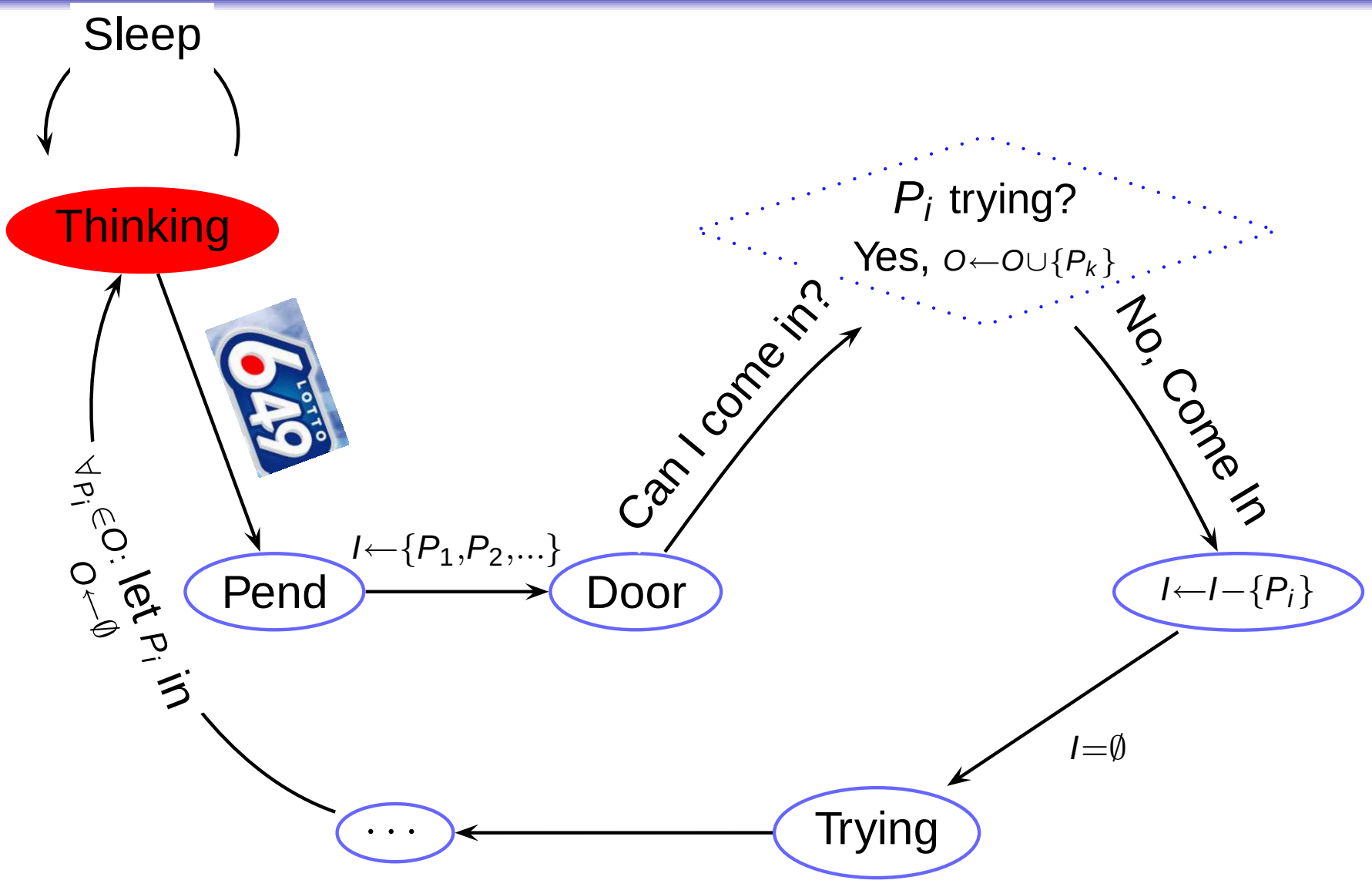
Control Flow-Message Passing:P_k



Control Flow-Message Passing:P_k



Control Flow-Message Passing:P_k



Result

- The authors claim with propability 1, this algorithm is deadlock free
- By carefully examining, Nancy Lynch proves after some philosphers feels hungry and try to grab forks, there is at least one philospher eats within time 13 and probability $\frac{1}{8}$.
- For the message passing algorithm the probability, that the philosopher keeps trying more than time c, is at most $1/e^c$

Extension

- Scheduling TDMA in wireless sensor networks
- Drinking philosophers