Distributed Markov Chains

Ratul Saha
National University of Singapore

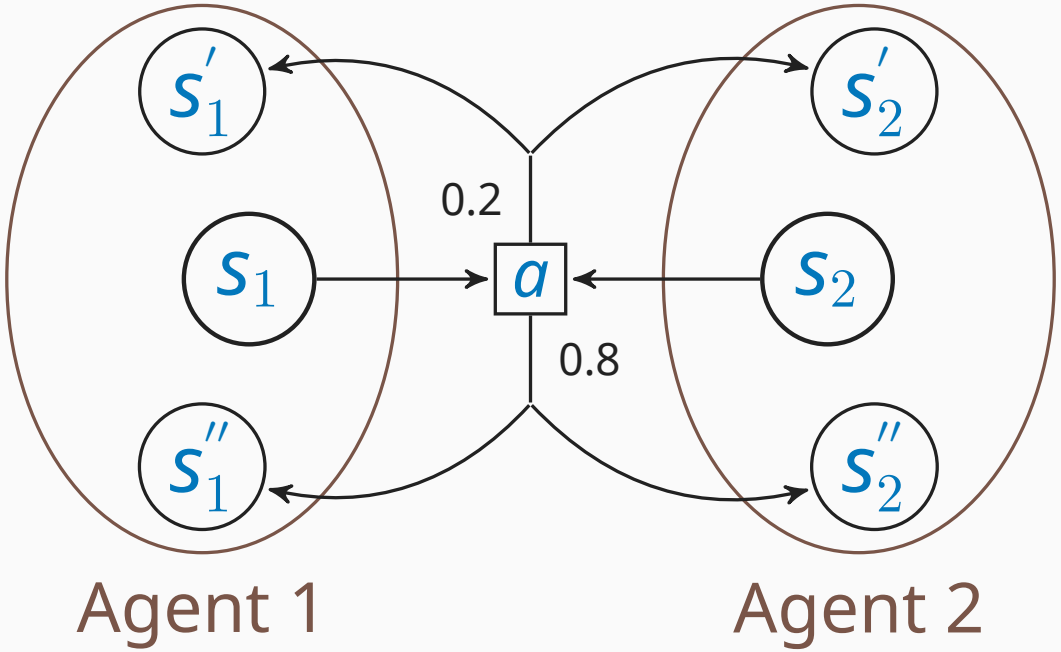
Joint work with:
Javier Esparza, Sumit K. Jha,
Madhavan Mukund, and P. S. Thiagarajan

Distributed Markov Chains (DMC)

- Network of communicating probabilistic transition systems
  - Synchronize on shared actions
  - Followed by joint probabilistic move
- Key restriction: no two enabled synchronizations will involve the same agent
  - Enforced syntactically (will explain soon)
- Efficient model checking using an interleaved semantics

DMC: Synchronization

→ Joint probabilistic move after the synchronization action

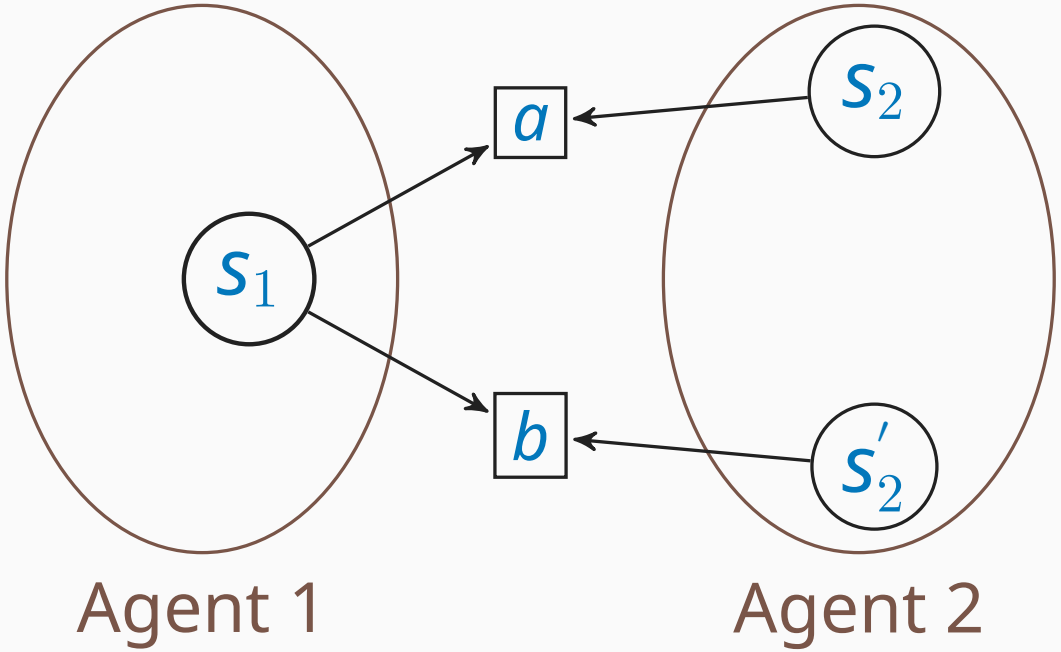


DMC: Key Restriction

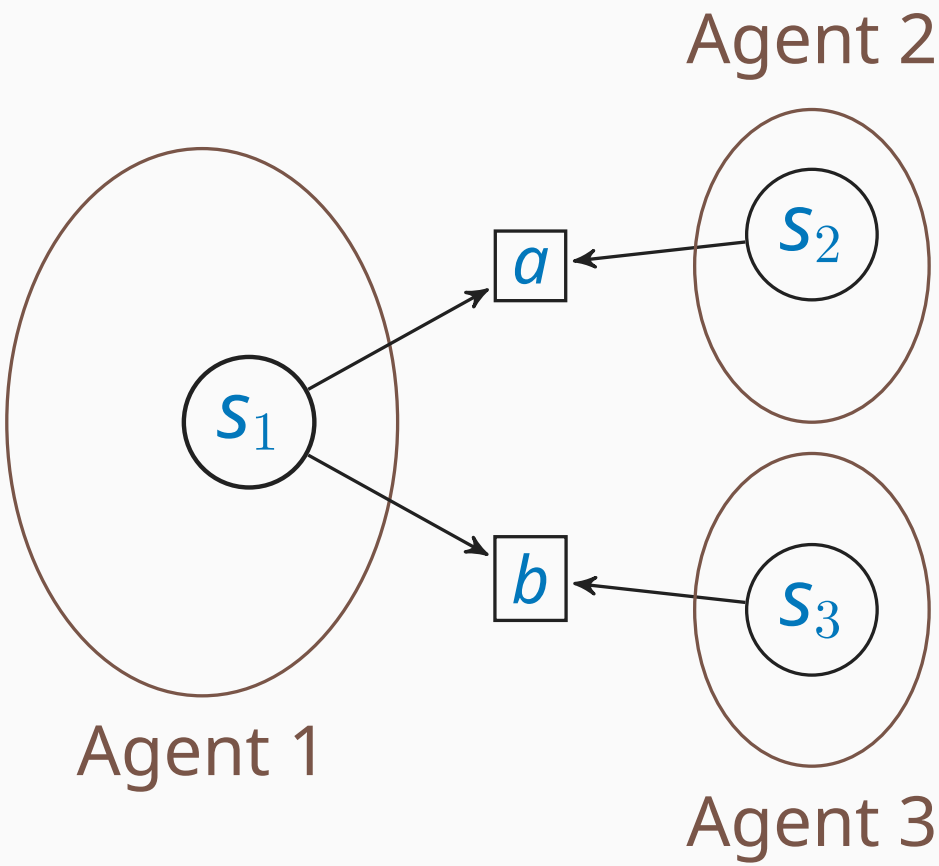
→ Any two simultaneously enabled actions involve disjoint sets of agents

→ Syntactically, local state uniquely determines its communicating partners

DMC: This is allowed

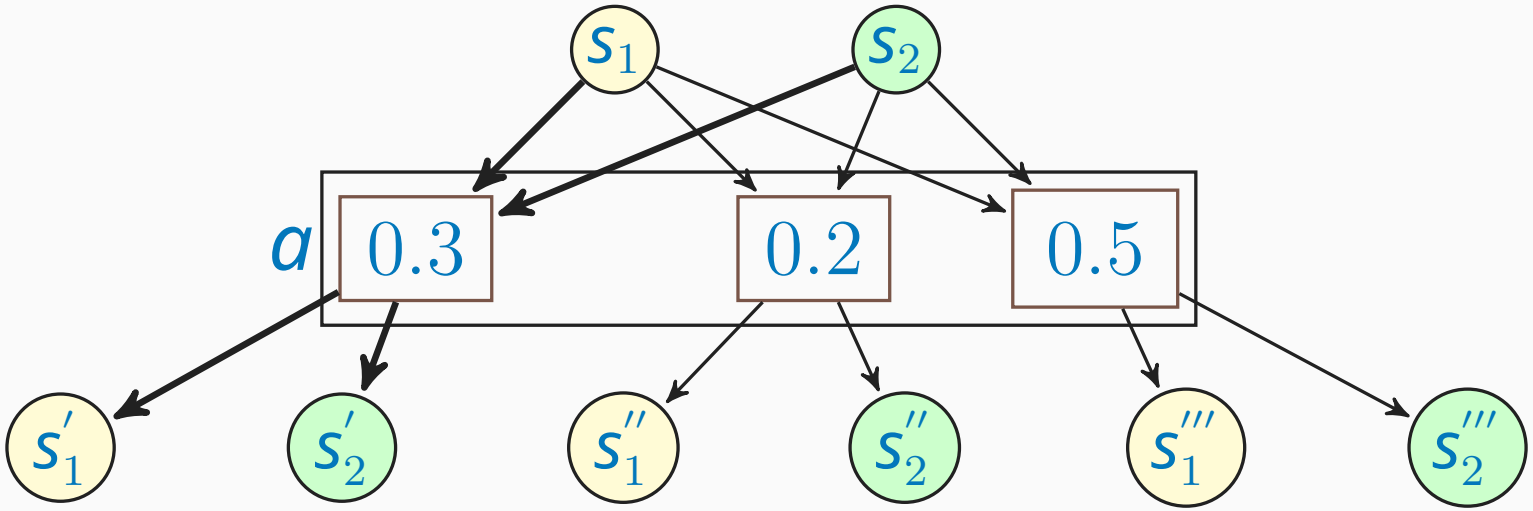


DMC: This is not allowed!



DMC: Events

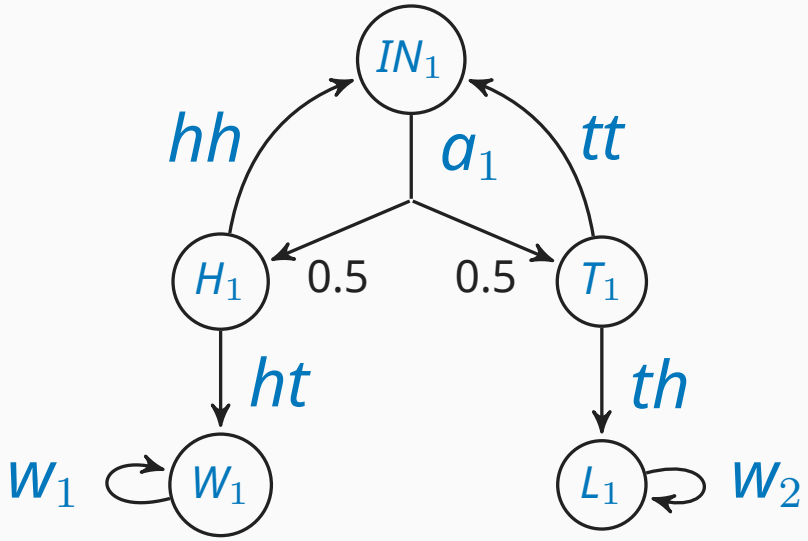
→ Event: One synchronization is executed at a time, followed by a probabilistic move by the participating agents



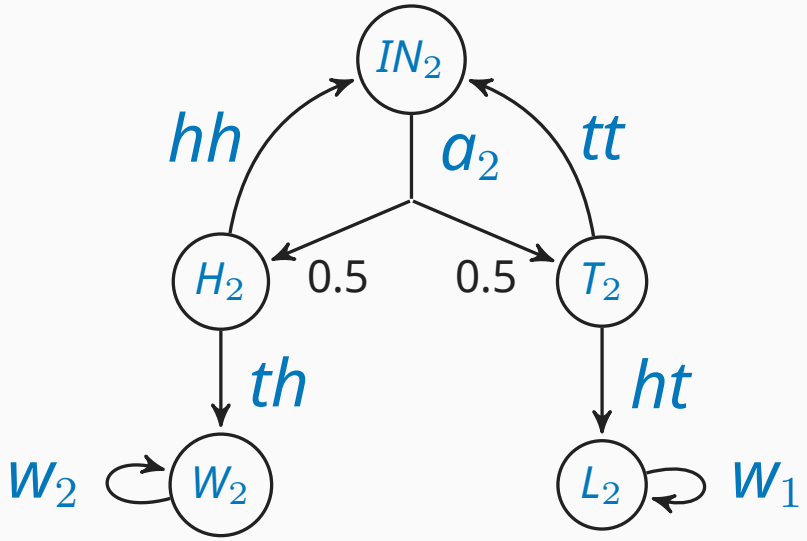
$e = ((s_1, s_2), a, (s_1', s_2'))$ is an event, $p_e = 0.3$

DMC: Coin Toss Example

- → Two players. Each toss a fair coin
- → Outcomes are the same: they toss again
- → Outcomes are different: who tosses Heads wins



Agent 1



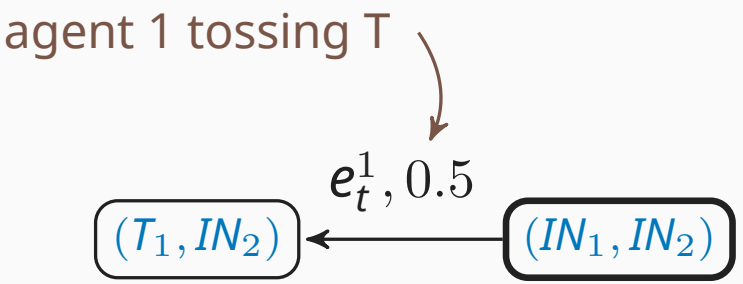
Agent 2

Global Transition System

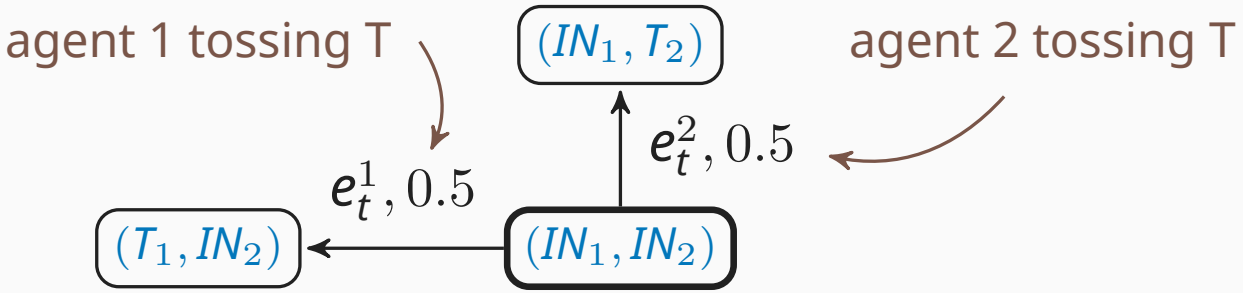
→ Associate a global transition system based on event occurrences

→ This is interleaved semantics

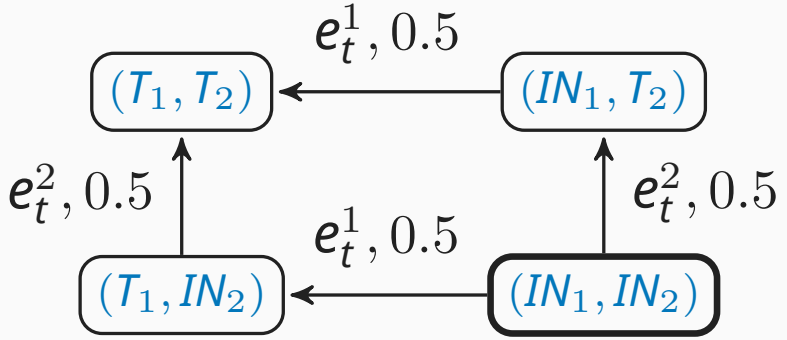
Global Transition System: Coin Toss



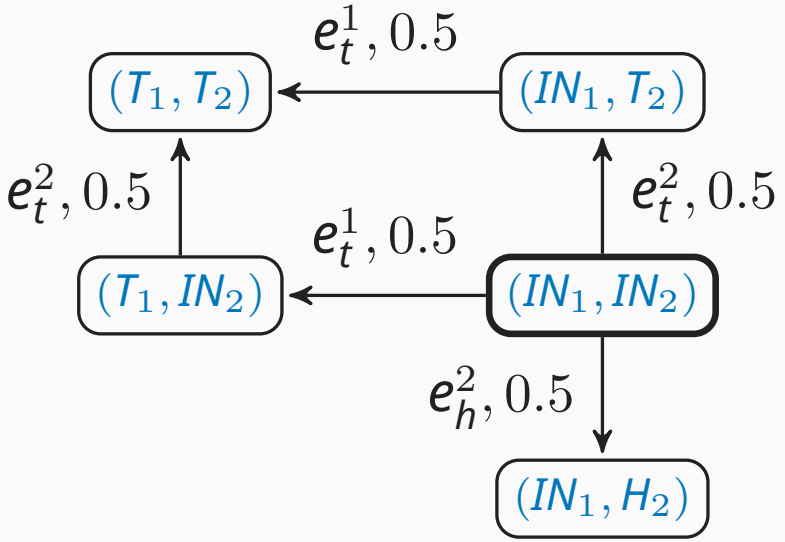
Global Transition System: Coin Toss



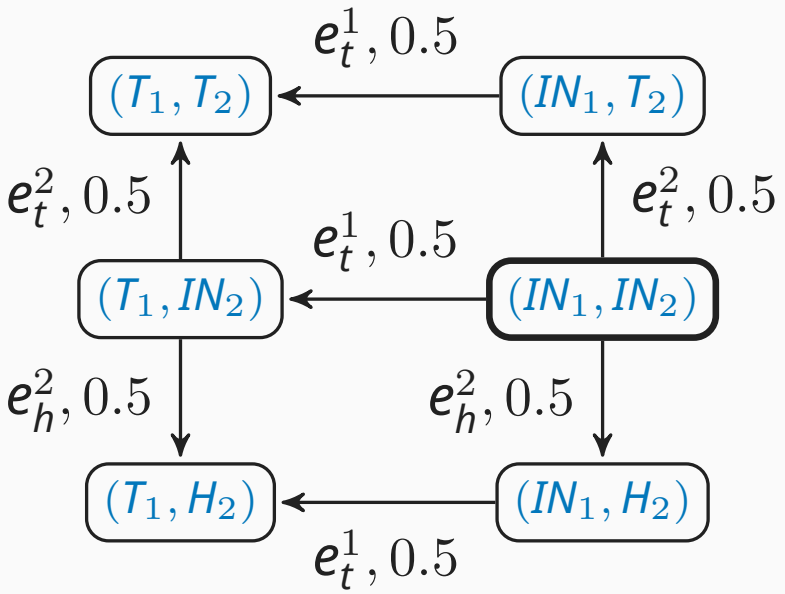
Global Transition System: Coin Toss



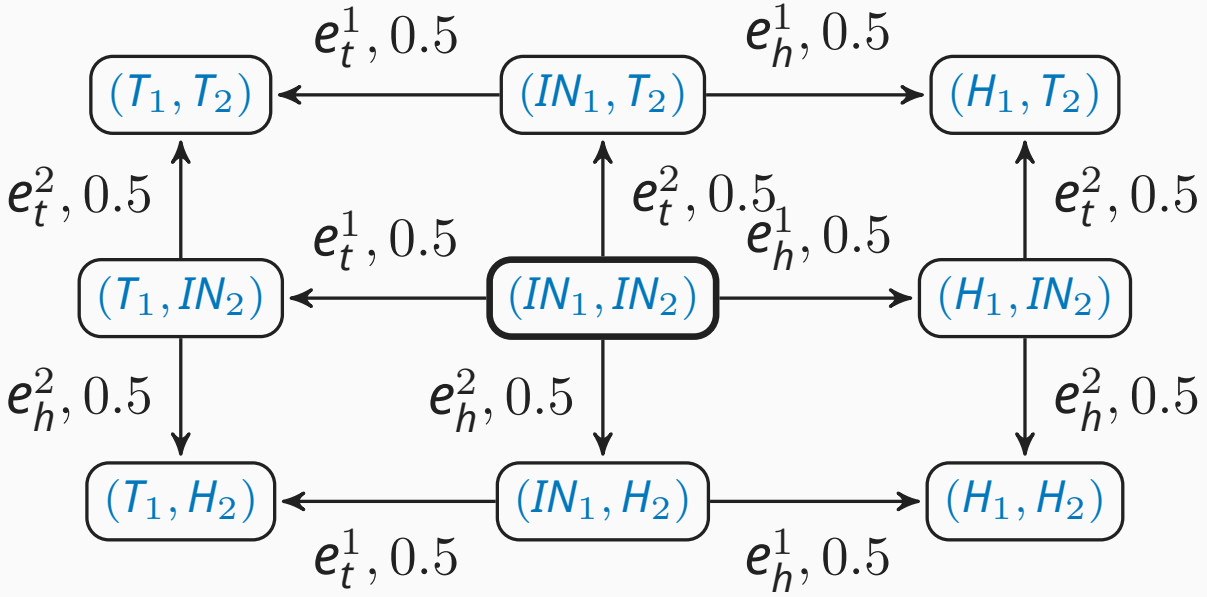
Global Transition System: Coin Toss



Global Transition System: Coin Toss

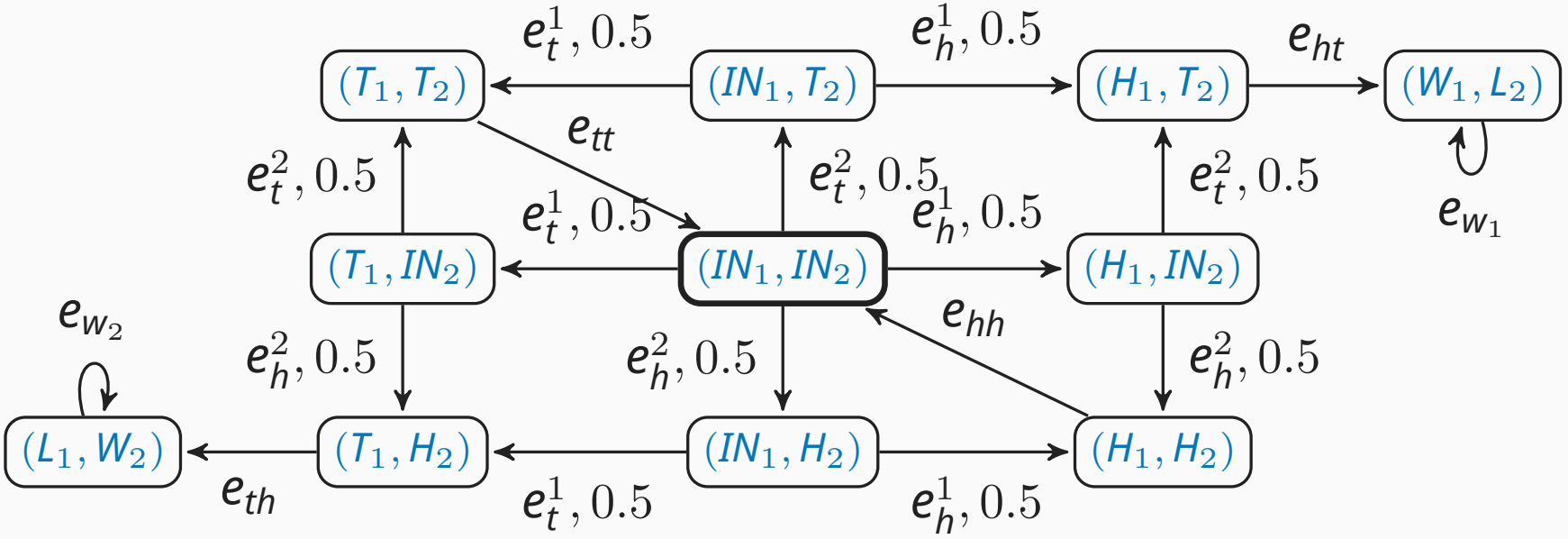


Global Transition System: Coin Toss



(both agents tossed)

Global Transition System: Coin Toss



(full global transition system)

(unmarked events have probability 1)

The Trajectory Space

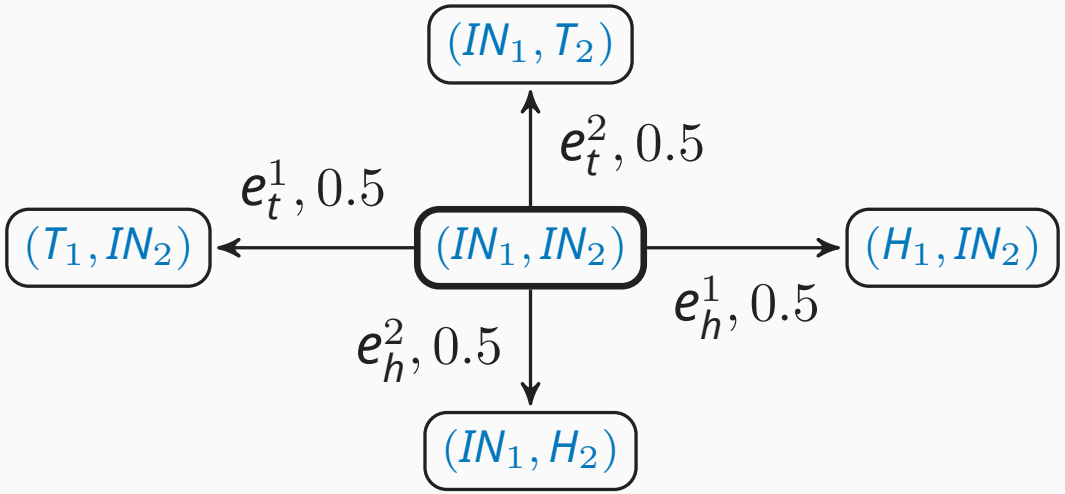
→ We refer to paths in *TS* as trajectories

→ We wish to reason about the behavior of the system using the interleaved semantics

Problem: It is *hard* to define a probability measure over the set of maximal trajectories

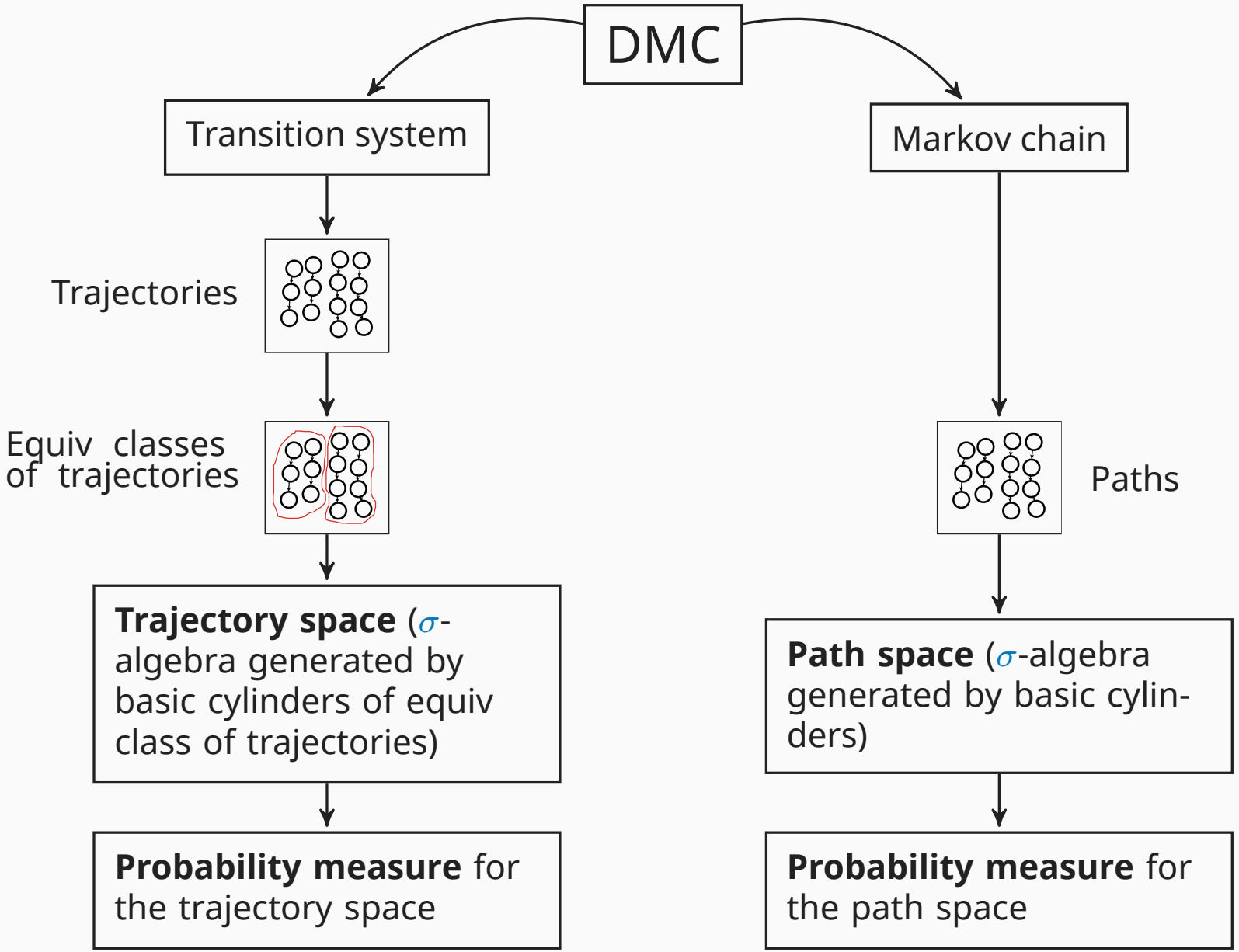
The Trajectory Space

Due to mix of concurrency and stochasticity, *TS* is not a Markov chain in general

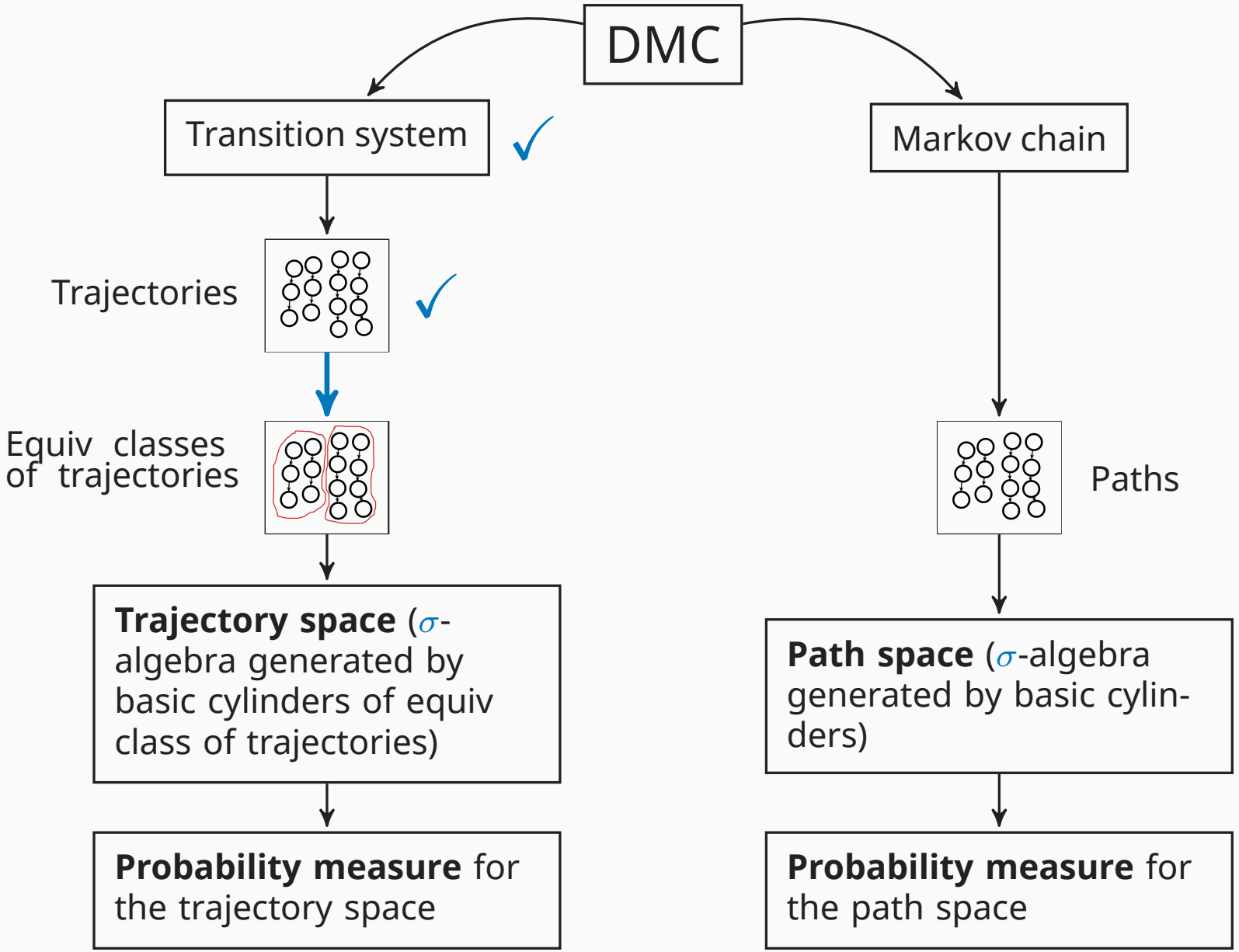


Here, the sum of the probabilities of the transitions from the state (IN_1, IN_2) is 2

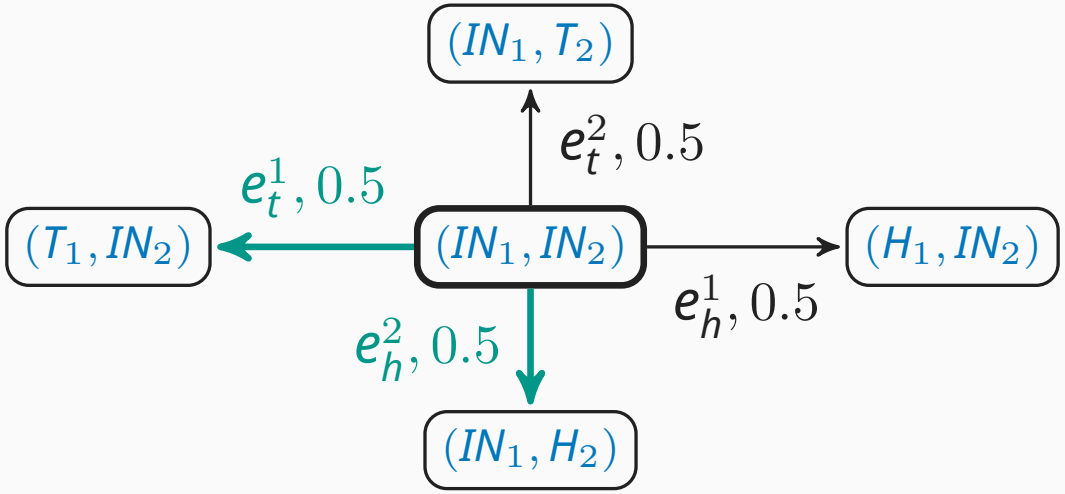
The Solution



Equivalence Classes of Trajectories

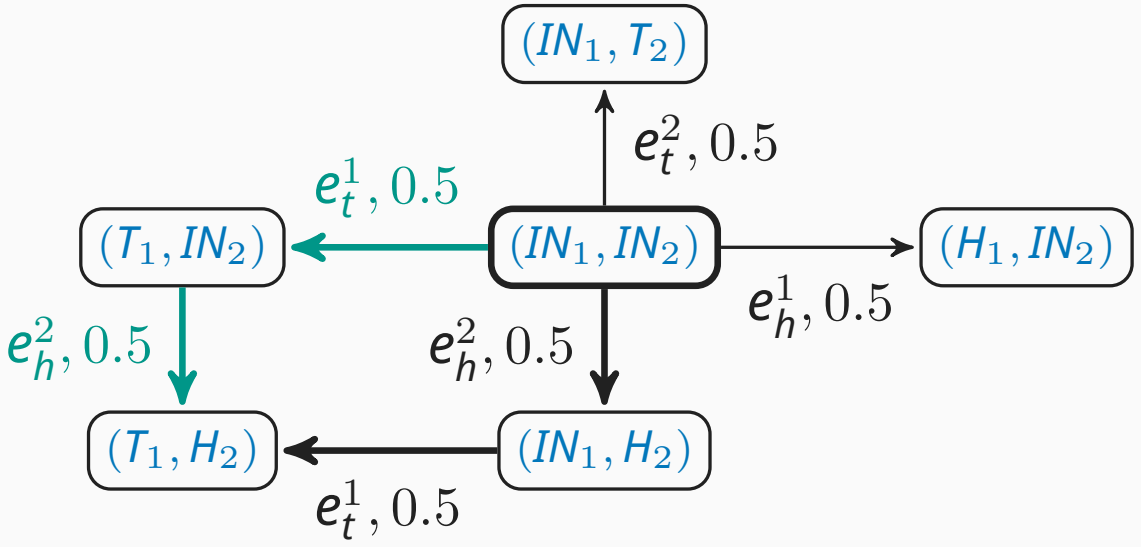


Independence over Events



→ $e_t^1 \; I \; e_h^2$ — agent 1 tossing tail and agent 2 tossing head are independent

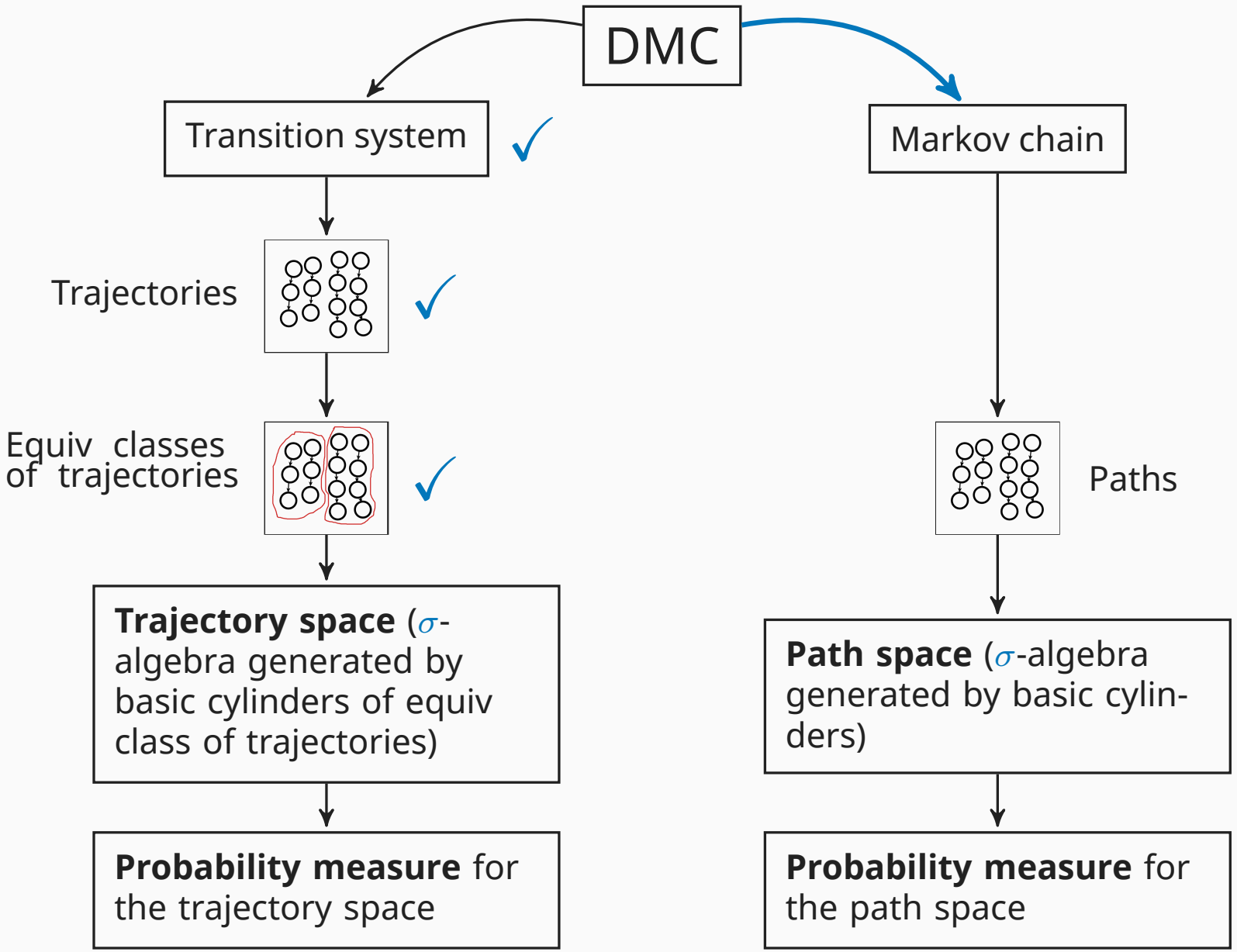
Equivalence over Event Sequences



→ $[e_t^1 e_h^2] = \{e_t^1 e_h^2, e_h^2 e_t^1\}$ — equivalence class over event sequences

→ Lifts to equivalence over trajectories

Markov Chain Semantics



Markov Chain Semantics

→ A step at a global state is a maximal set of independent enabled events

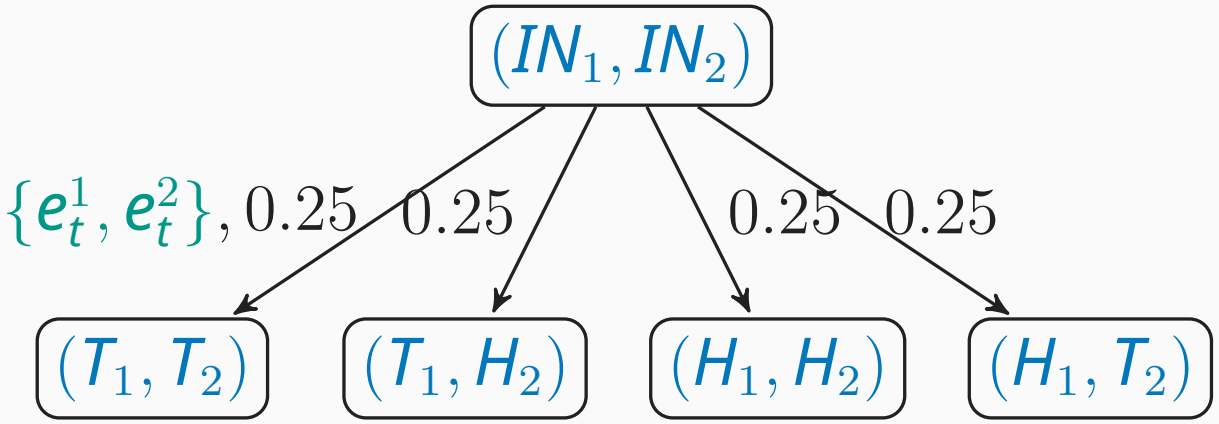
→ The transition relation using steps induces a Markov chain

Markov Chain Semantics

→ $\{e_t^1, e_t^2\}$ is a maximal step at (IN_1, IN_2)

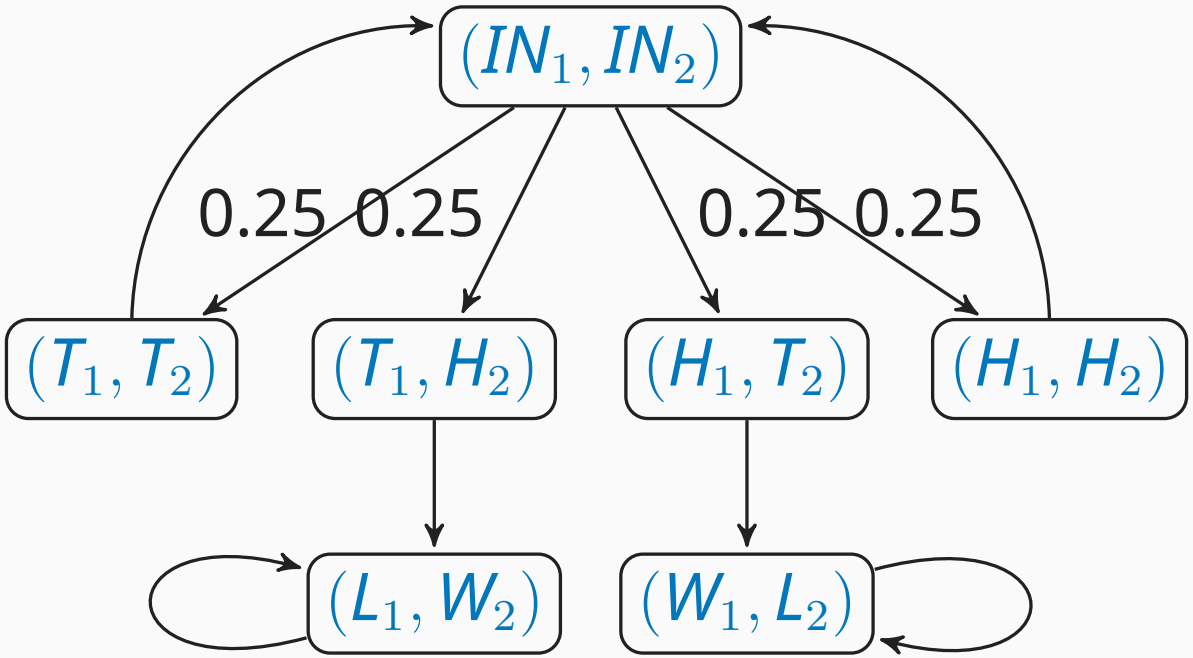
→ The probability of a step is the product of probabilities associated with the events in the step

Markov Chain Semantics



The maximal steps at (IN_1, IN_2) are
$\{e_h^1, e_h^2\}$, $\{e_h^1, e_t^2\}$, $\{e_t^1, e_h^2\}$, $\{e_t^1, e_t^2\}$

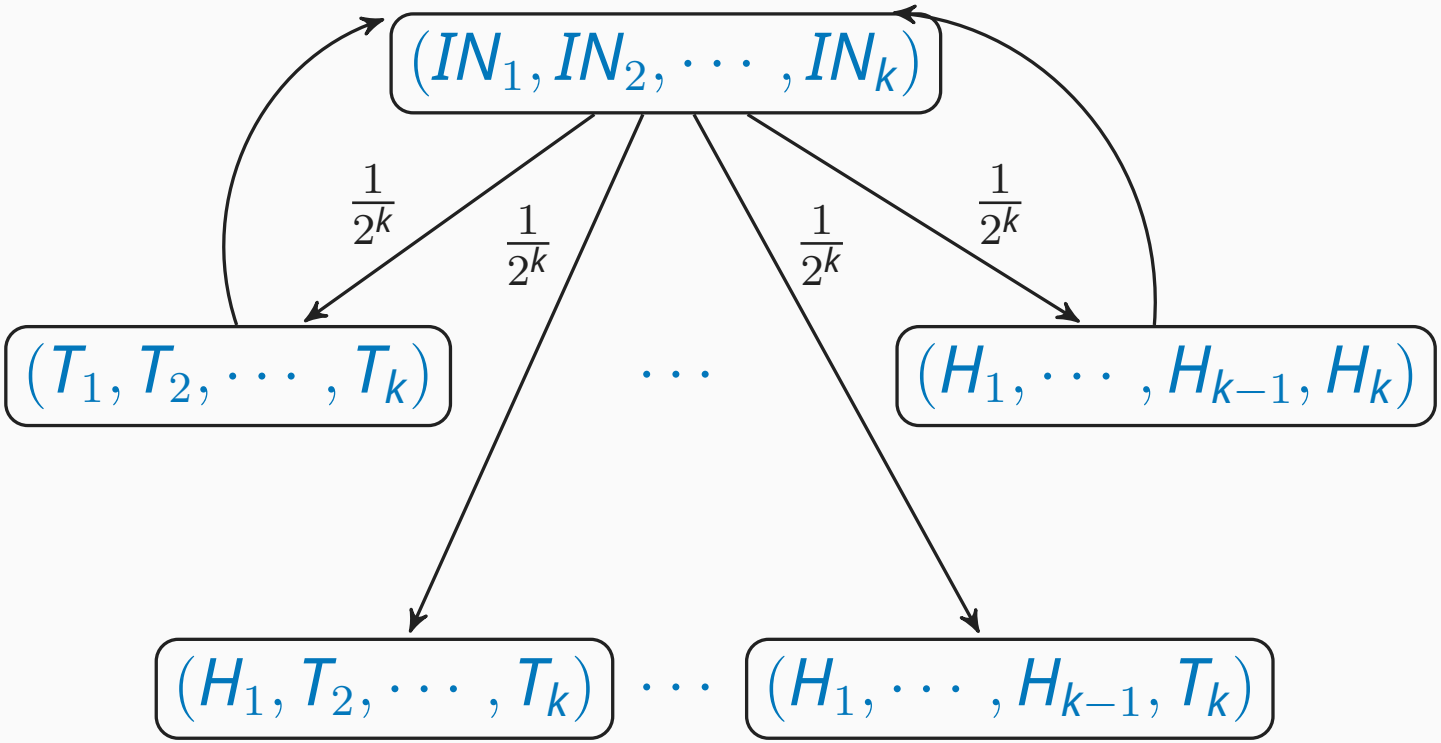
Coin Toss: Global Markov Chain



(The unmarked transitions have probability 1)

Coin Toss: Global Markov Chain

What if there were k players?



k parallel probabilistic moves generate 2^k global transitions

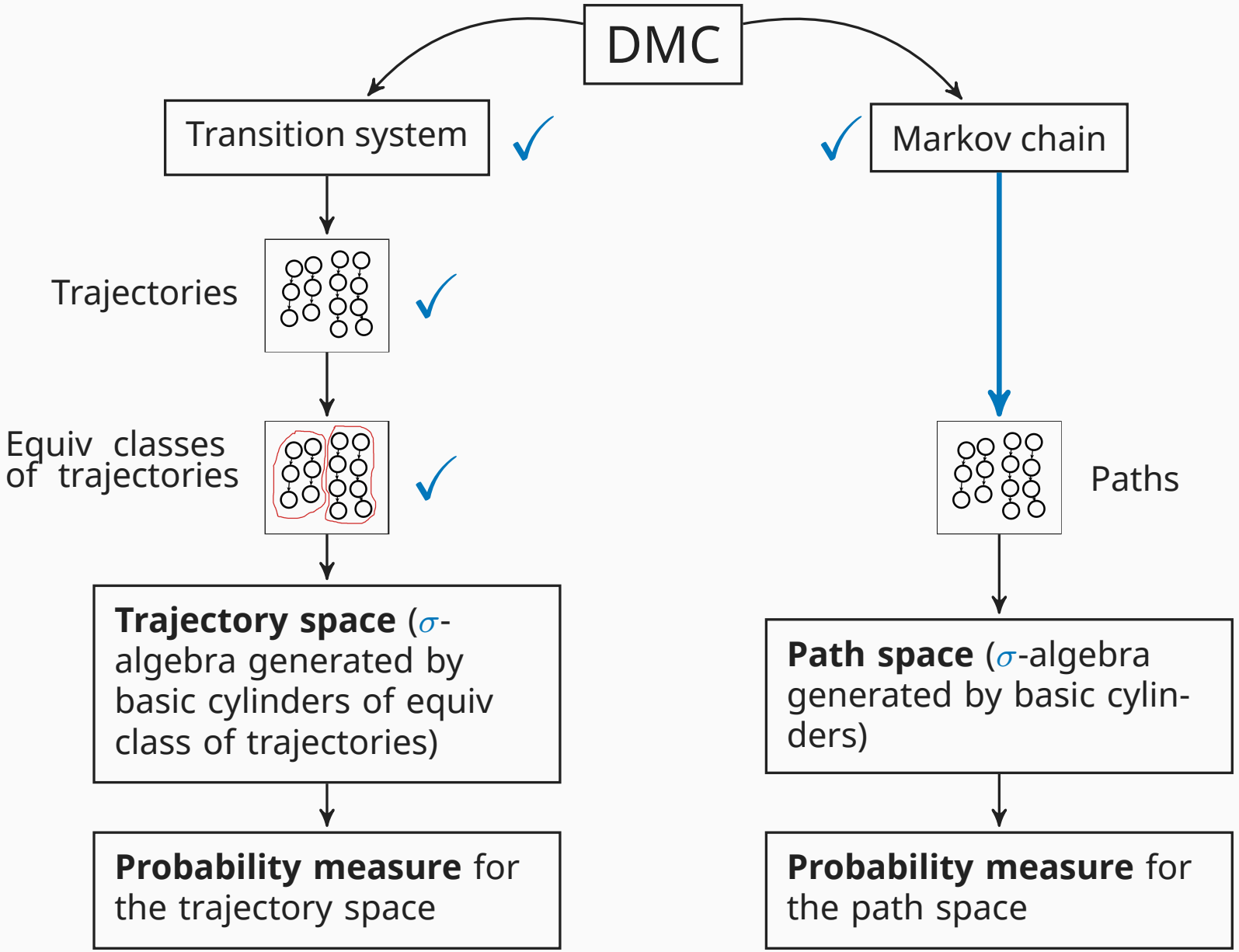
Markov Chain Semantics

The number of transitions out of a global state can be (in number of agents)

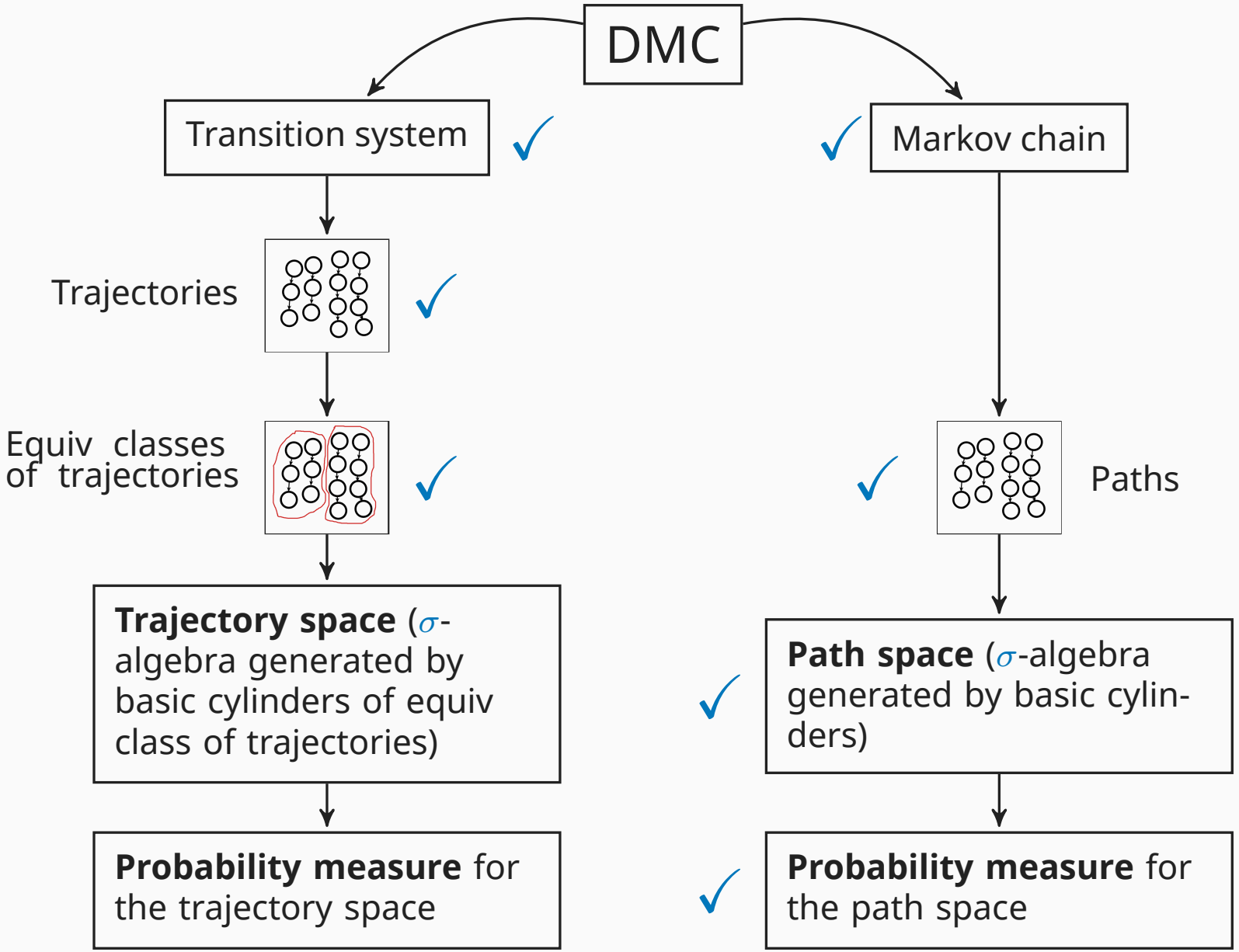
exponential
in Markov chain semantics

polynomial
in interleaved semantics

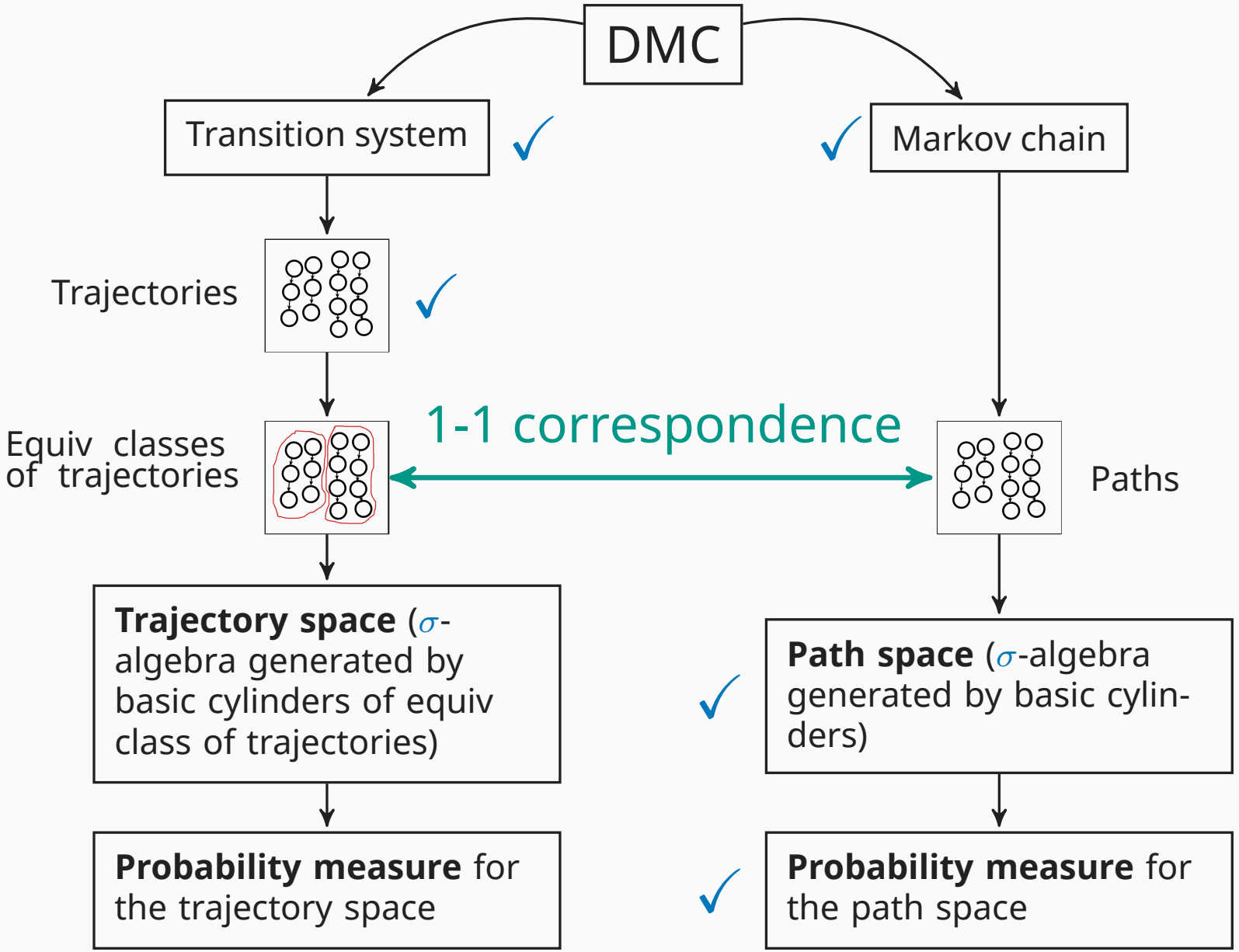
Markov Chain Semantics



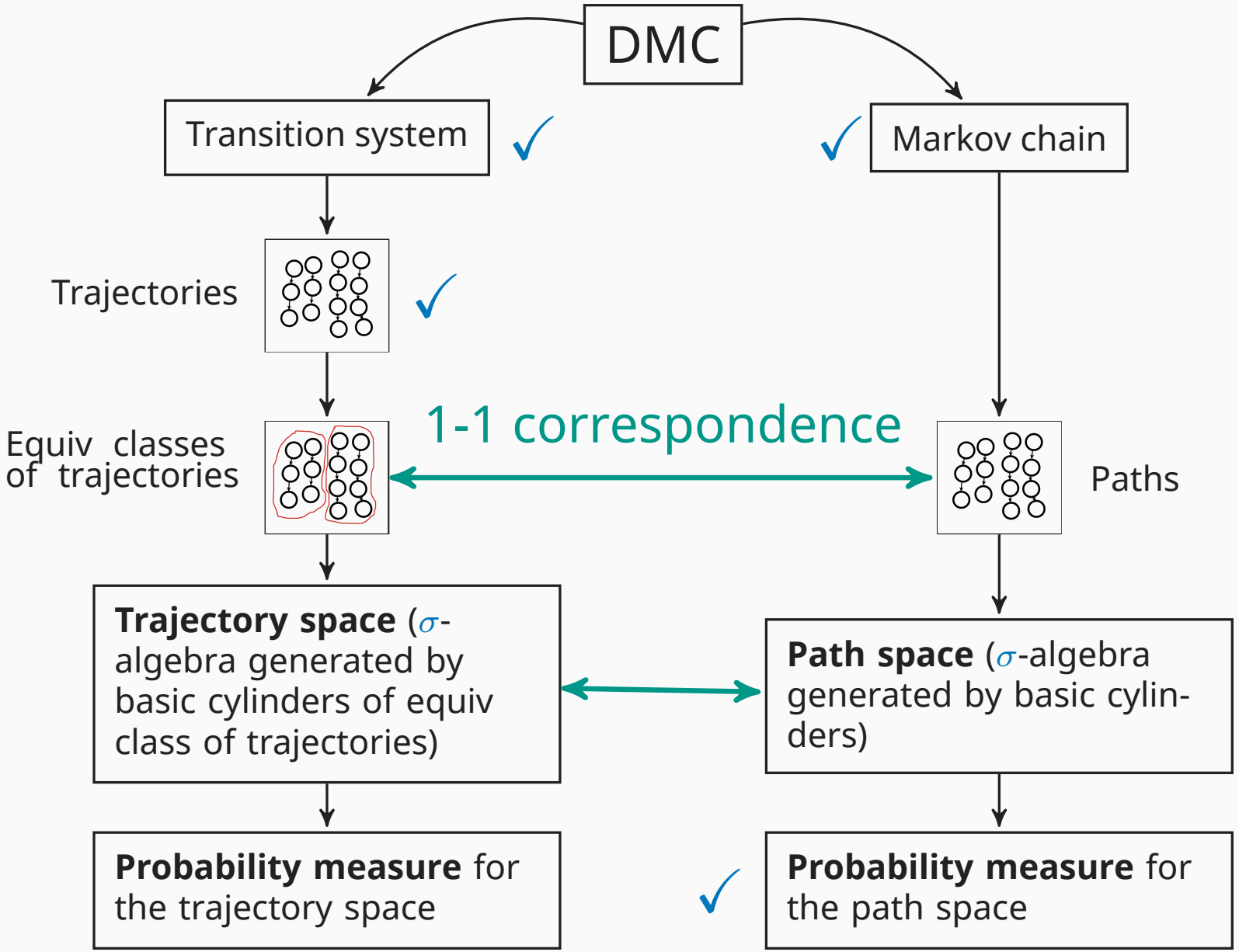
Markov Chain Semantics



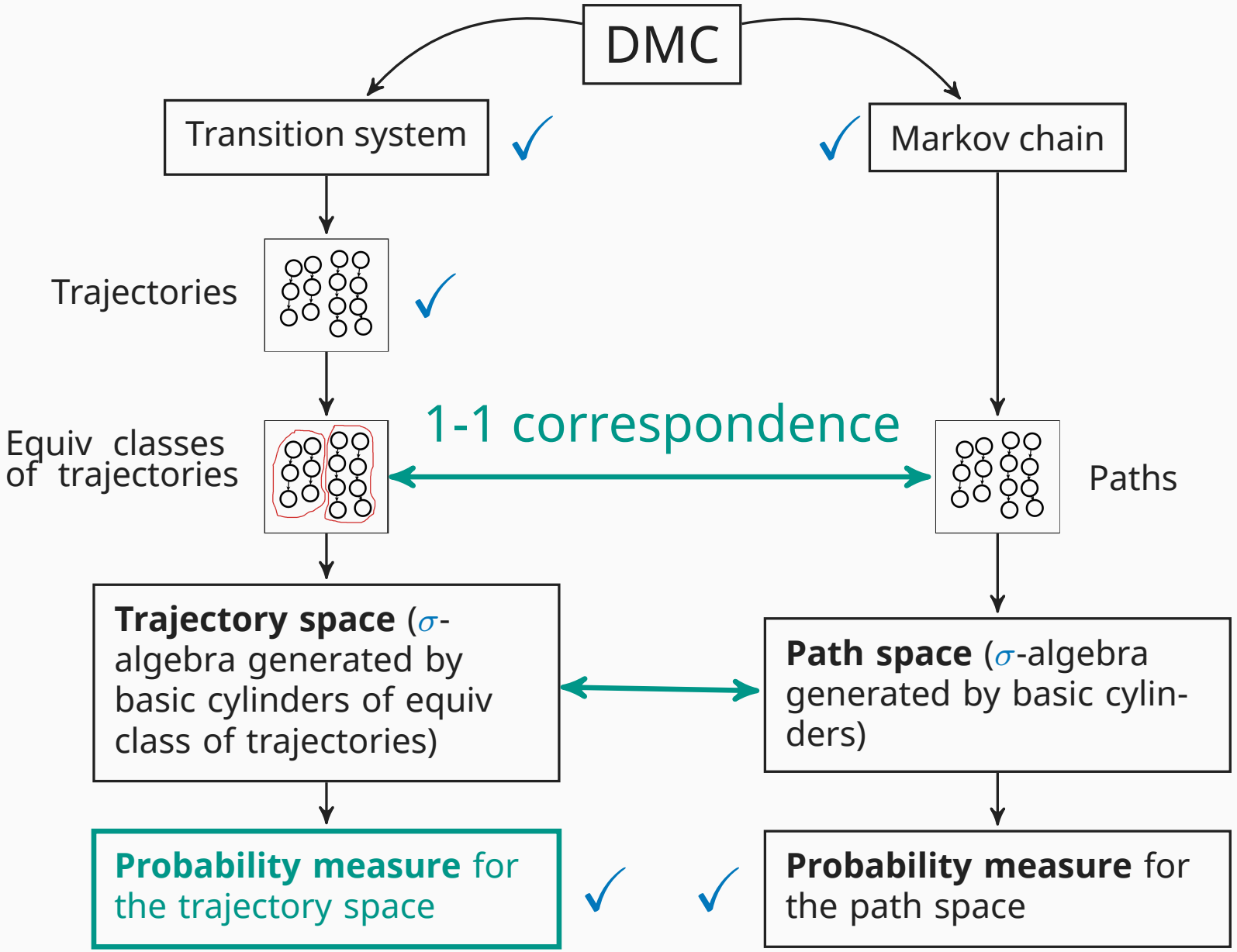
Defining the Probability Measure



Defining the Probability Measure



Defining the Probability Measure



$PBLTL^{\otimes}$: The Specification Logic

Local Bounded LTL ($BLTL_i$)

- → Time bounded *LTL*
- → Each agent i has a local set of atomic propositions AP_i
- → Formula of type i: $ap \in AP_i \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \mathsf{U}_i^k \varphi_2$
  - → $\varphi_1 \mathsf{U}_i^k \varphi_2$ — Until holds within k (local) moves of agent i

Product Bounded LTL ($BLTL^{\otimes}$)

- → Boolean combinations of $\{BLTL_i\}$ formulas

$PBLTL^{\otimes}$: The Specification Logic

Probabilistic Product Bounded LTL ($PBLTL^{\otimes}$)

$\rightarrow$ $Pr_{\geq\gamma}(\varphi)$, where φ is a $BLTL^{\otimes}$ formula

Example (coin toss):

$$Pr_{\geq 0.99}\left[\left(F^{7}(L_1) \wedge F^{7}(W_2)\right) \vee \left(F^{7}(W_1) \wedge F^{7}(L_2)\right)\right]$$

with probability at least 0.99, the coin toss game terminates within 7 rounds

(the local states serve as the atomic propositions)

Statistical Model Checking

- Given a DMC and $PBLTL^{\otimes}$ formula $Pr_{\geq\gamma}(\varphi)$
- Explicit computation is impractical for very large systems
- Instead, estimate through sampling
  - Draw sample trajectories from the interleaved semantics
  - Use statistical estimation: returns "γ in (predefined) interval (a, b)" with high confidence

Experimental Results

→ Modeled a number of PRISM benchmarks including:

(i) Distributed leader election protocol [Itai and Rodeh]
(ii) A randomized solution to the dining Philosophers problem [Pnueli and Zuck]

→ Compared simulation time with a statistical model checker — PLASMA

Distributed Leader Election

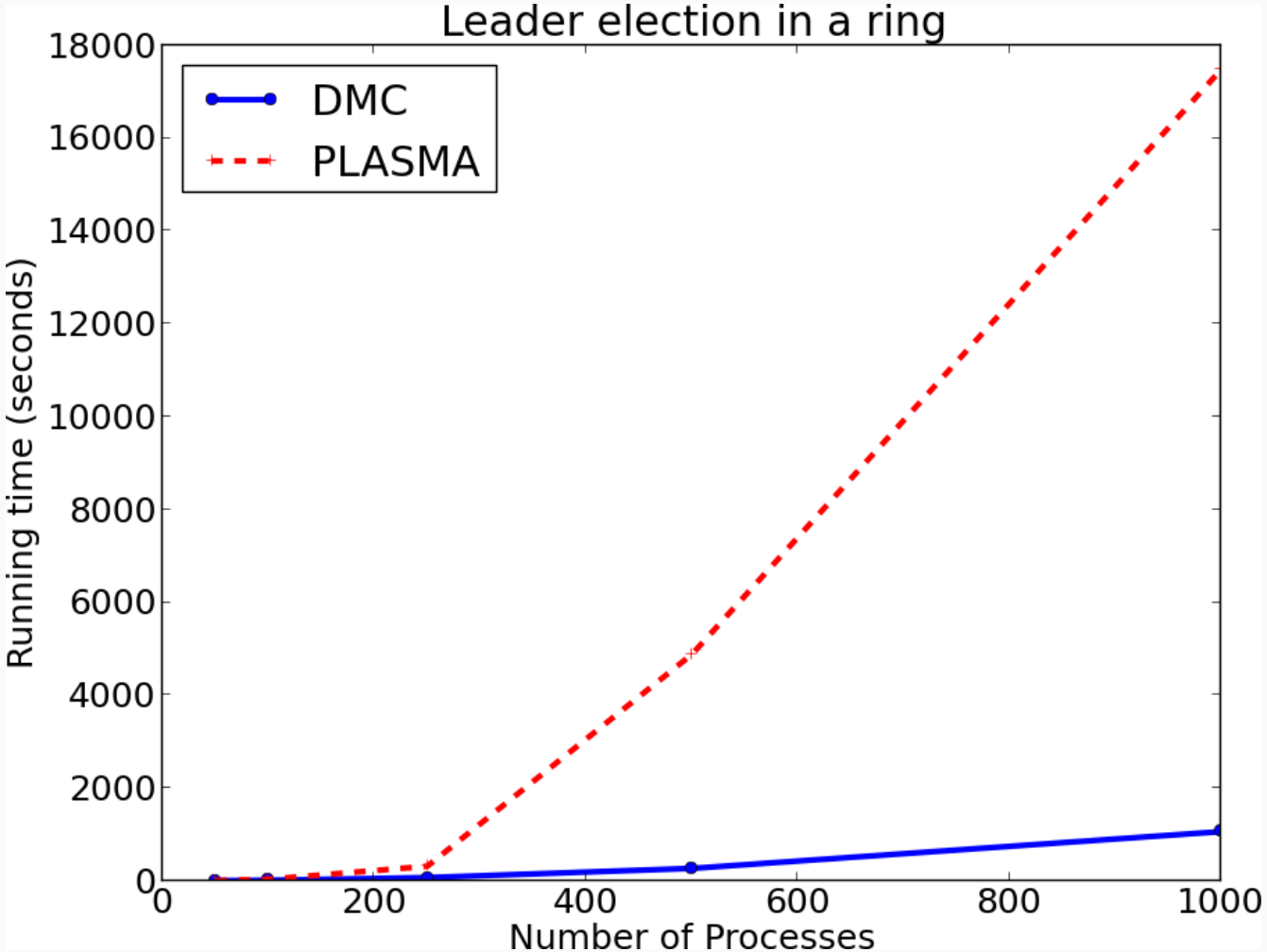
- → Verify: with probability 1, a leader is elected eventually
- → Since specification logic is $BLTL^{\otimes}$ and the verification procedure is SMC, we instead verify:

  In a ring of N nodes, with high probability (p), a leader will be elected within B rounds

  (Type I and II error = 0.01, indifference region = 0.01, for various choices of N and B)

Distributed Leader Election: Comparison with PLASMA

$p = 0.99$, N up to 1000 (no parallelization in DMC)



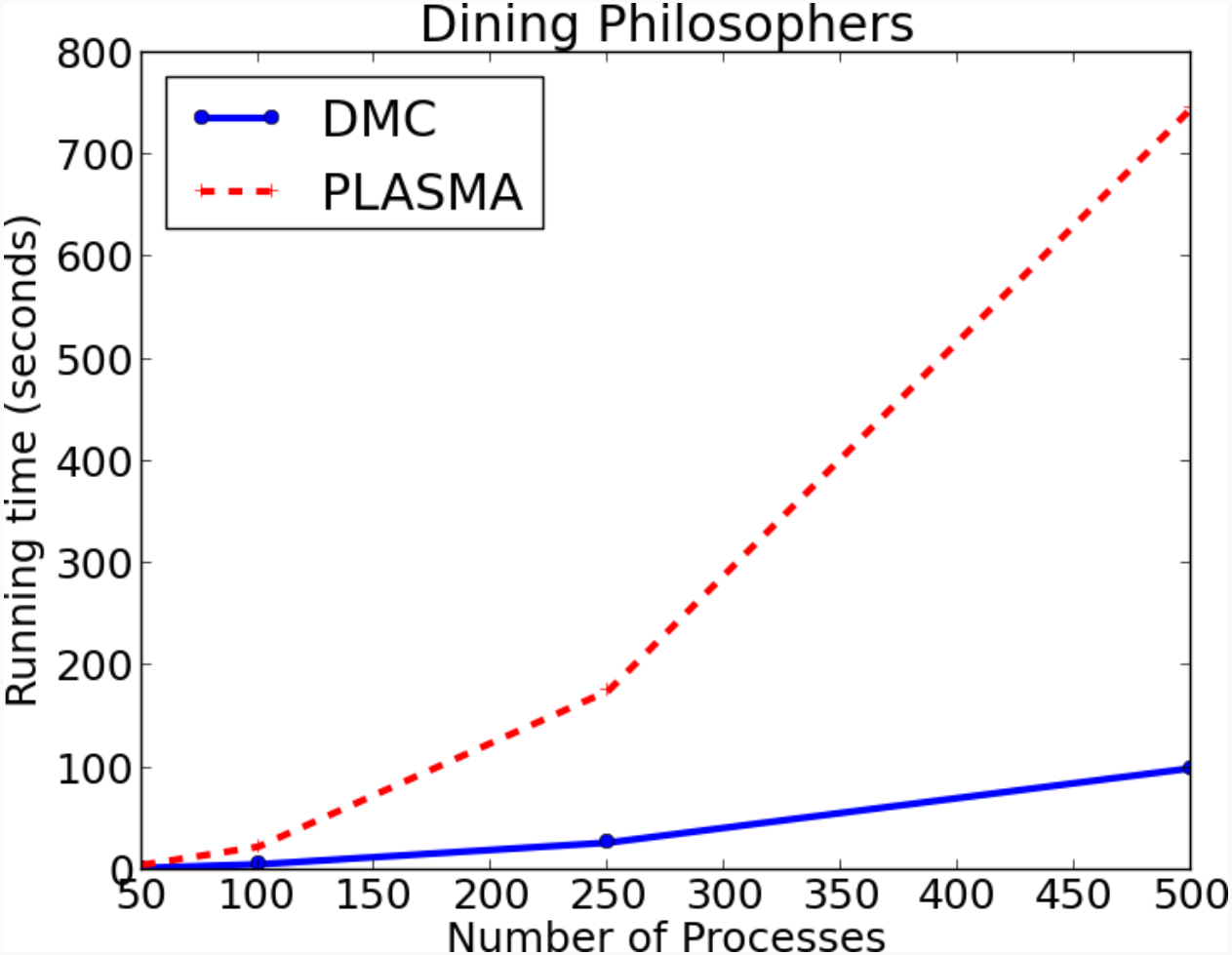
Dining Philosophers Problem

- To start, a philosopher probabilistically chooses the order in which it will try the forks
- Forks between philosophers are also agents in DMC formalism
- We use deterministic round robin protocol to simulate shared variable
- The property we verify:

  With high probability (p), every philosopher eats within B rounds

Dining Philosophers Problem: Comparison with PLASMA

$p = 0.95$, N up to 500 (no parallelization in DMC)



Future Work

- → Other case studies from PRISM benchmark
- → Extend our current sequential SMC procedure to a parallel implementation
- → Study DMCs with observable and non-observable components
- → Other application domains (eg. distributed decision making in robotics)
- → Exact probabilistic verification for PCTL and other probabilistic temporal logics

THANK YOU!