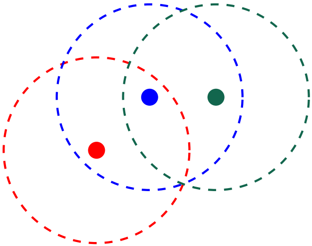
Adding Probabilities to Networks with Selective Broadcast

Arnaud Sangnier

Joint Work with:
Nathalie Bertrand and Paulin Fournier

Parameterized Verification - 5th september 2015

Ad Hoc Networks



Main characteristics of Ad Hoc Networks

- Nodes can be mobile
- Topology is not known a priori
- Messages are broadcasted to the neighbours
- Problems linked to communication (collision, loss of messages, etc.)

Parameterized model for ad hoc network

Networks with a parametric number of processes executing the same ’program’

A bit of history:

- Parametric model with broadcast [CONCUR’10]
- Qualitative reachability problem [CONCUR’10,FOSSACS’10]
- Model with faulty behavior [FORTE’12]
- Analysis quantitative reachability queries [FSTTCS’12]
- Adding identifiers and data to the model [RP’13]
- **Introducing local probabilistic choices in the model** [FOSSACS’14]

In this talk

Characteristics of the model

- Each node executes a finite state process
- Communication through broadcast of messages
- Messages sent to neighbors
- Neighbors can change non deterministically at any moment
- Number of entities not fixed a priori (**parameter**)
- Internal probabilistic choice for changing state

Verification problem

- Probabilistic version of control state reachability
  - Qualitative question (probability 0 or 1)
  - Minimize or maximize the probability

Difficulties:
Infinite state system + non-determinism + probabilities

Outline

1. **Probabilistic Reconfigurable Broadcast Network (PRBN)**
2. **Parity Reconfigurable Broadcast Networks (Parity RBN)**
3. **Playing with probabilities in PRBN**
4. **Conclusion and future works**

Outline

1. **Probabilistic Reconfigurable Broadcast Network (PRBN)**
2. Parity Reconfigurable Broadcast Networks (Parity RBN)
3. Playing with probabilities in PRBN
4. Conclusion and future works

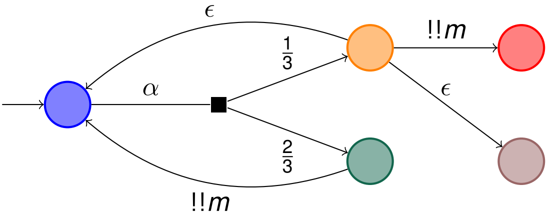
Model

A probabilistic process $P = \langle Q, \Sigma, R, q_0 \rangle$

Finite state system whose transitions are labeled with:

1. broadcast of messages - $!!m$
2. reception of messages - $??m$
3. internal action - ε
4. **Probabilistic actions** - α

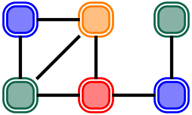
where m belongs to the finite alphabet Σ



PRBN: Configurations

A configuration is a graph $\gamma = \langle V, E, L \rangle$

- V : finite set of vertices
- $E : V \times V$: finite set of edges
- $L : V \to Q$: labeling function



- **Initial configurations**: **all** vertices are labeled with initial state

Remarks:

- The size of the considered graphs is not bounded
- Infinite number of configurations

$\Rightarrow$ **PRBN are infinite state systems**

PRBN: Semantics

Markov decision process $\mathcal{M}_P = \langle C, \Rightarrow, C_0 \rangle$ Induced by P

- C: (infinite) set of configurations
- $\Rightarrow$: $C \times C \cup C \times \mathit{Dist}(C)$: Transition relation
- C_0: (infinite) set of initial configurations

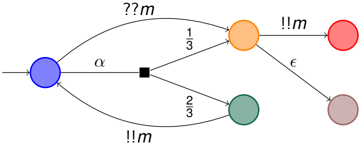
The transition relation is decomposed in three phases:

1. Choose of a process
2. Execution of an action
3. Choose of a new topology

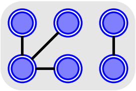
Properties

- The number of processes does not change
- At each step the topology can evolve non-deterministically

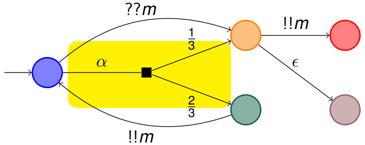
With fixed N and a Scheduler: Finite Markov chain



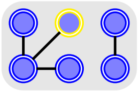
A scheduler π resolve the non-determinism
It chooses the process, the action, and the new topology.



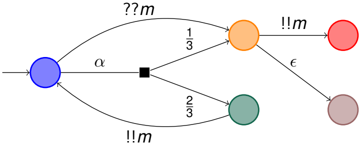
With fixed N and a Scheduler: Finite Markov chain



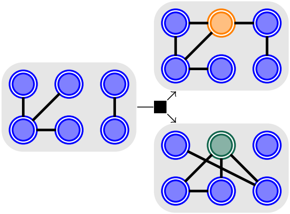
A scheduler π resolve the non-determinism
It chooses the process, the action, and the new topology.



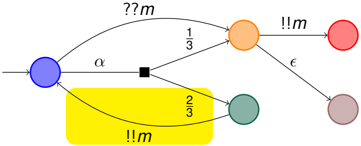
With fixed N and a Scheduler: Finite Markov chain



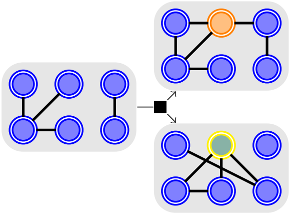
A scheduler π resolve the non-determinism
It chooses the process, the action, and the new topology.



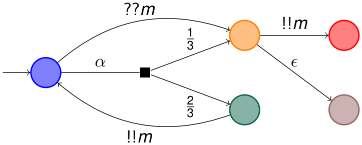
With fixed N and a Scheduler: Finite Markov chain



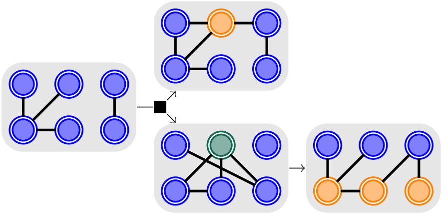
A scheduler π resolve the non-determinism
It chooses the process, the action, and the new topology.



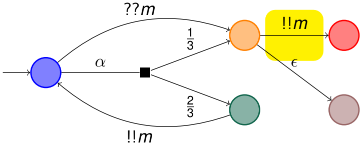
With fixed N and a Scheduler: Finite Markov chain



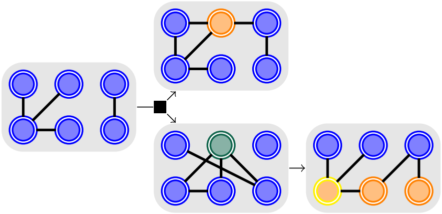
A scheduler π resolve the non-determinism
It chooses the process, the action, and the new topology.



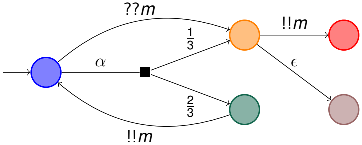
With fixed N and a Scheduler: Finite Markov chain



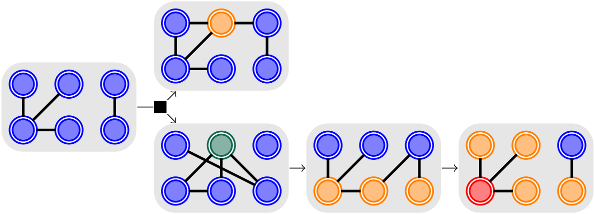
A scheduler π resolve the non-determinism
It chooses the process, the action, and the new topology.



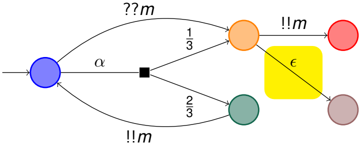
With fixed N and a Scheduler: Finite Markov chain



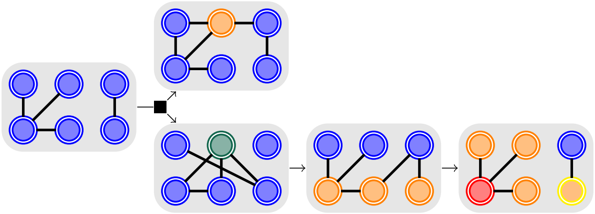
A scheduler π resolve the non-determinism
It chooses the process, the action, and the new topology.



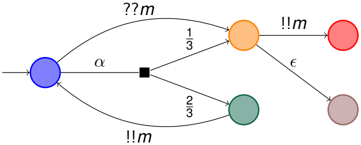
With fixed N and a Scheduler: Finite Markov chain



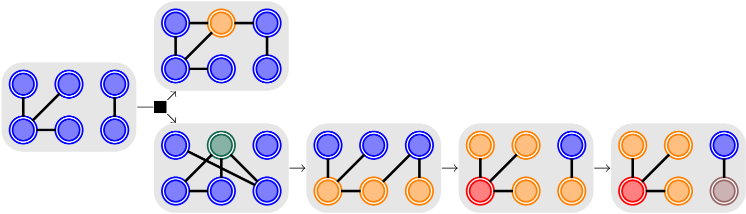
A scheduler π resolve the non-determinism
It chooses the process, the action, and the new topology.



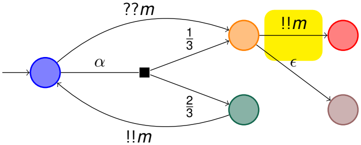
With fixed N and a Scheduler: Finite Markov chain



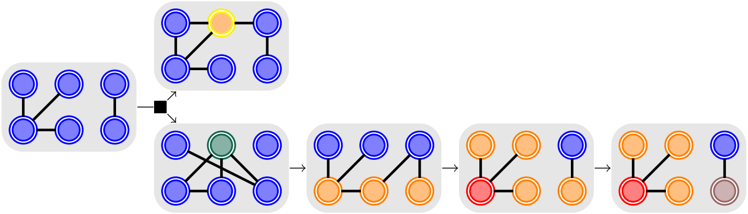
A scheduler π resolve the non-determinism
It chooses the process, the action, and the new topology.



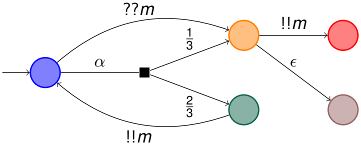
With fixed N and a Scheduler: Finite Markov chain



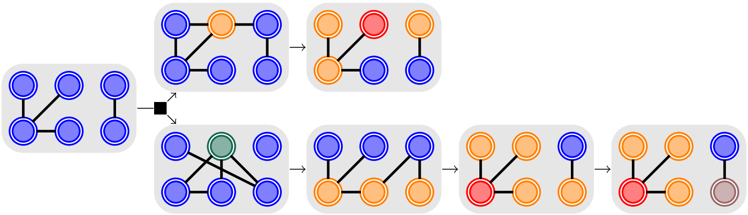
A scheduler π resolve the non-determinism
It chooses the process, the action, and the new topology.



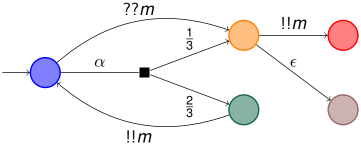
With fixed N and a Scheduler: Finite Markov chain



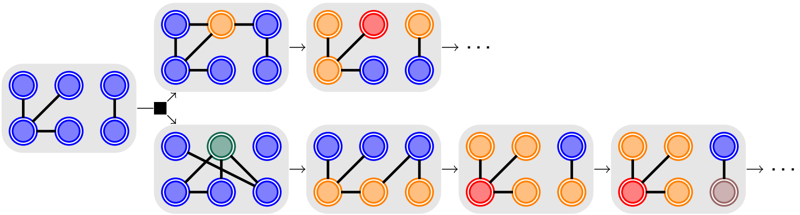
A scheduler π resolve the non-determinism
It chooses the process, the action, and the new topology.



With fixed N and a Scheduler: Finite Markov chain



A scheduler π resolve the non-determinism
It chooses the process, the action, and the new topology.



Studied Problems

$\text{REACH}_{opt}^{\sim b}$ $opt \in \{\min, \max\}, \sim \in \{>, <, \leq, \geq, =\}, b \in \{0, 1\}$

Input: A process and a control state $q_F \in Q$;

Output: Does there exists N such that:

$$\underset{\pi}{opt} \{\mathbb{P}(\mathcal{M}_P, N, \pi, \Diamond q_F)\} \sim b$$

$\Rightarrow$ N denotes here the number of initial process **Example of interesting questions:**

- Is a target state reachable almost surely, for some number of processes? $\text{REACH}_{\max}^{=1}$
- Is a target avoidable almost surely, for all numbers of processes and all decisions? $\text{REACH}_{min}^{>0}$

Trivial cases

$\textsc{Reach}^{=0}_{\max}$

- Consider the case with a single node
- If we have $\max_{\pi} \{\mathbb{P}(\mathcal{M}_P, 1, \pi, \Diamond q_F)\} > 0$
- Then for all $N > 1$, $\max_{\pi} \{\mathbb{P}(\mathcal{M}_P, 1, \pi, \Diamond q_F)\} > 0$
- The reason being you can always execute a process on its own with no communication
- So we have $\textsc{Reach}^{=0}_{\max}$ iff $\max_{\pi} \{\mathbb{P}(\mathcal{M}_P, 1, \pi, \Diamond q_F)\} = 0$

The same reasoning holds for $\textsc{Reach}^{<1}_{\max}$, $\textsc{Reach}^{=1}_{\min}$, $\textsc{Reach}^{>0}_{\min}$

Proposition

$\textsc{Reach}^{=0}_{\max}$, $\textsc{Reach}^{<1}_{\max}$, $\textsc{Reach}^{=1}_{\min}$ and $\textsc{Reach}^{>0}_{\min}$ are in PTIME.

This because with a single node, we have a finite state MDP and we use result on qualitative reachability in finite MDPs

Other cases

$\textsc{Reach}_{\max}^{>0}$

- Equivalent to parameterized control state reachability
- **Is in PTIME** [Delzanno et al., FSTTCS'12]
- It is even possible to compute the set of reachable states in polynomial time
- And to have an execution which reaches a configuration with a high number of nodes in each of these reachable states

What about $\textsc{Reach}_{\min}^{=0}$, $\textsc{Reach}_{\min}^{<1}$ and $\textsc{Reach}_{\max}^{=1}$?

- This problem are more difficult
- No previously known techniques can be applied
- Dealing with branching in infinite states systems with probability and non determinism is a difficult task

A word on Probabilities in Infinite State Systems

- RBN are **Well-Structured Transitions Systems**
  - *What you can do with small configurations can be achieved with bigger configurations*
- However the algorithms for reachability are not based on classical algorithms for WSTS
  - The model allows much more efficient algorithms
- There exists a general work on WSTS equipped with probabilities but **without non determinism**
  - Decisive Markov Chains [Abdulla et al. 2007]
- Some infinite state systems have been extended with probabilities and non-determinism
  - Non-deterministic and Probabilistic Lossy Channel System [Baier et al. 2007]
  - Recursive Markov Decision Process [Etessami et al. 2015]
- Adapting the used techniques seems difficult

A way to solve problems in finite state MDPs

- Solving qualitative reachability in finite MDP is easy
- It can be achieved in polynomial time
- One way to solve it although is to consider a μ-calculus formula [Chatterjee et al. 2007]
- There is a strong connection between μ-calculus and parity game
- Qualitative reachability problem can be transformed into solving a parity game

This is the path we will follow

Main difficulties:

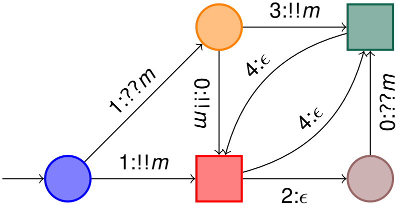
1. Define a game with broadcast networks
2. Find methods to solve this game

Outline

Parity Protocol

A protocol $P = \langle Q, Q_1, Q_2, \Sigma, R, q_0, \texttt{safe} \rangle$

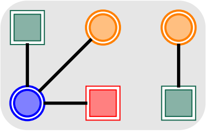
- Q_1 states belonging to Player 1: 
- Q_2 states belonging to Player 2: 
- Transitions are labelled with:
  1. broadcast of messages: $!!m$ (*resp.*)
  2. reception of messages: $??m$
  3. internal action: ε
     $\Rightarrow$ **mandatory for transitions leaving player 2's states**
  4. parities, i.e. colors in $\mathbb{N}$
  5. $\texttt{safe} \subseteq R$ set of safe transitions



Parity RBN: Configurations

A configuration is a graph $\gamma = \langle V, E, L \rangle$

- V: finite set of vertices (also called processes)
- $E : V \times V$: finite set of edges (also called topology)
- $L : V \to Q_1 \cup Q_2$: labels (current state of processes)
- Plus a single active node



- **Initial configurations**: **all** vertices are labeled with initial state

Remarks:

- The size of the considered graphs is not bounded
- Infinite number of configurations

$\Rightarrow$ **Parity RBN are infinite state systems**

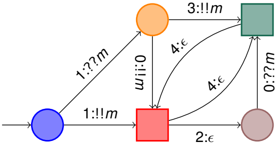
Characteristic of the induced game

- Each configuration is a vertex of the game
- The number of processes does not change
- Topology is controlled by Player 1 and can evolve at each step
- The number of configurations is unbounded
- Configurations of Player 2 are configurations where a node with label in Q_2 is active
- Colors are on the transitions (the colors of broadcast is considered)
- Safe transitions involved only safe actions

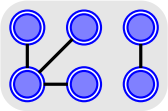
Strategies for Player 1 (σ) and Player 2 (τ) are asymmetric

- They are asymmetric
- Player 1 chooses the new topology and the active process v
- Player 1 or Player 2 chooses the action depending on $L(v) \in Q_1$ or $L(v) \in Q_2$

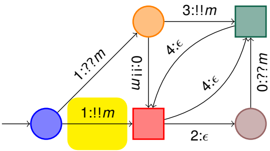
Example



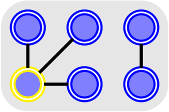
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



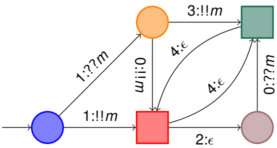
Example



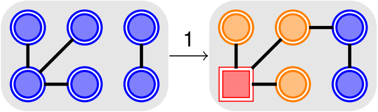
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



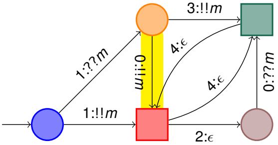
Example



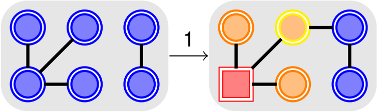
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



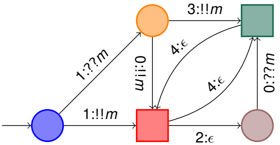
Example



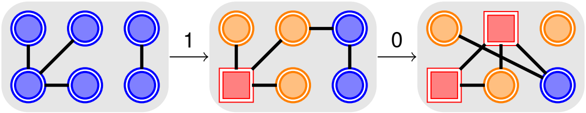
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



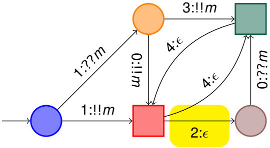
Example



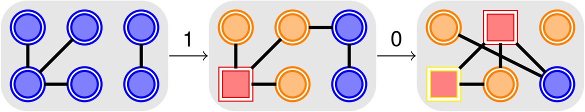
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



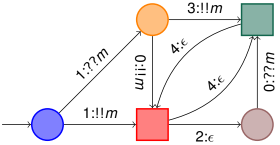
Example



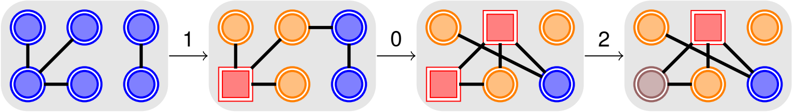
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



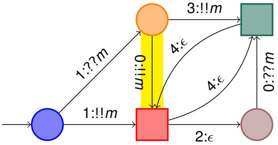
Example



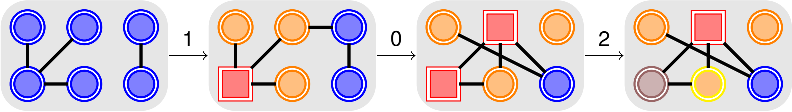
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



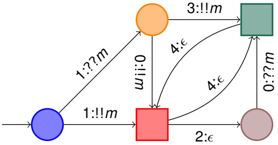
Example



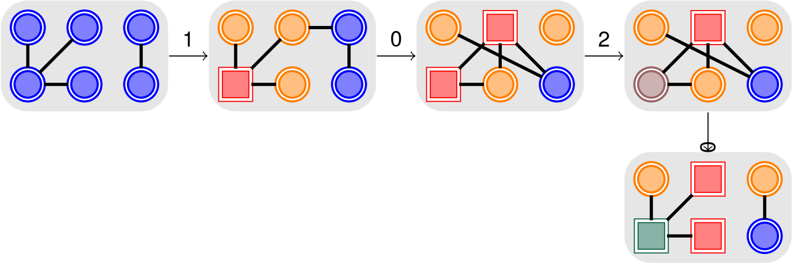
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



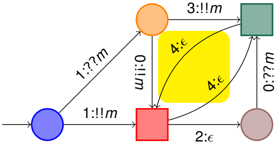
Example



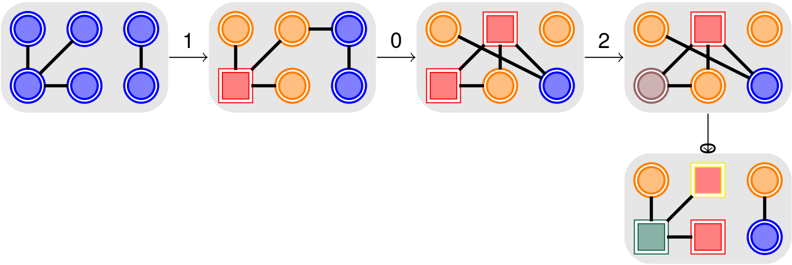
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



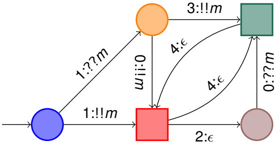
Example



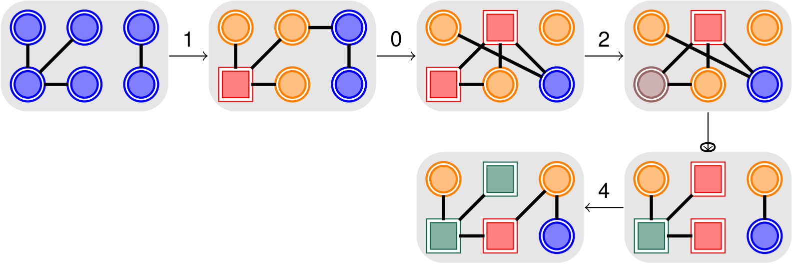
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



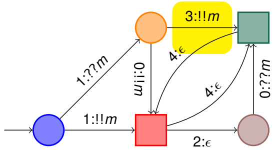
Example



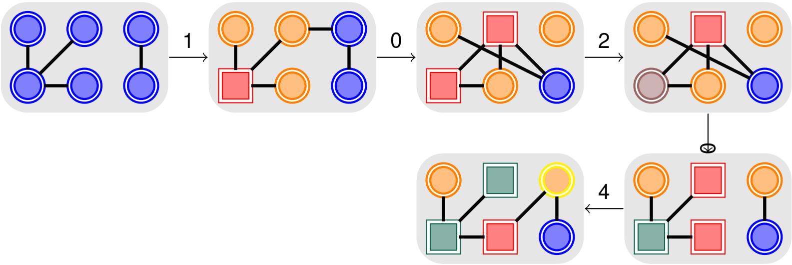
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



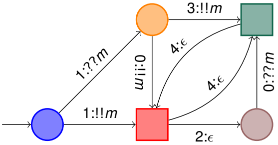
Example



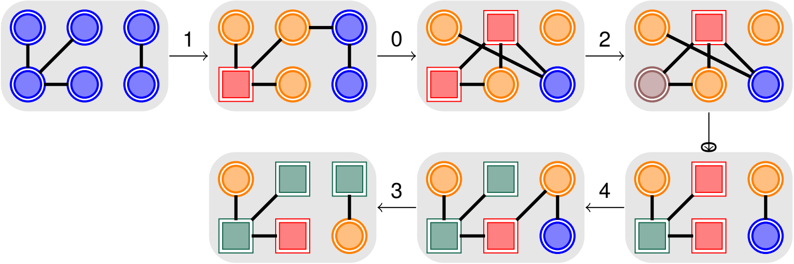
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



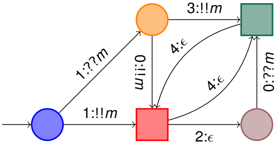
Example



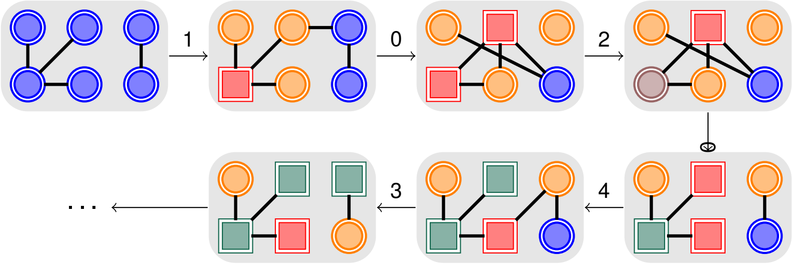
The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



Example



The strategies σ and τ define the play $\rho(\sigma, \tau, 6, P)$:



Studied Problem

Winning condition *Win*

The set of infinite plays ρ such that:

1. The maximal color repeated infinitely often is even
2. No unsafe transition is taken

Game problem for parity RBN

Input: A parity protocol P
Question: Does there exists N and a strategy σ for Player 1 such that for all strategies τ for Player 2:

$$\rho(\sigma, \tau, N, P) \in \mathit{Win}$$

How to solve games for parity RBN

The proof to solve such games respects the following steps :

1. Shows that it is enough to consider local strategies for Player 2
2. Show that in a Broadcast Reconfigurable Protocol (BRP), one can decide the existence of an infinite cycle

Local strategies

- Strategies that only depend on the control state labeling the active node
- There is a finite number of local strategies
- Once a local strategy is fixed, we obtain a normal BRP

Why is it enough to focus on local strategies

Proposition

If there exists a configuration and a strategy for Player 1 against any local strategy of Player 2, then there exists a configuration and a strategy for Player 1 winning against any strategy of Player 2.

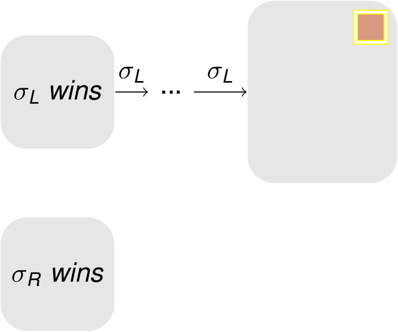
- The proof is by induction of the number of states of Player 2
- At the induction step, it isolates a state of Player 2  with two active edges L and R
- By induction if edge R is deleted, Player 1 has a winning strategy σ_L
- By induction if edge L is deleted, Player 1 has a winning strategy σ_R

Build a strategy using σ_L and σ_R

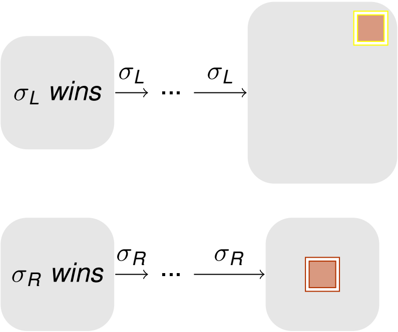
σ_L *wins*

σ_R *wins*

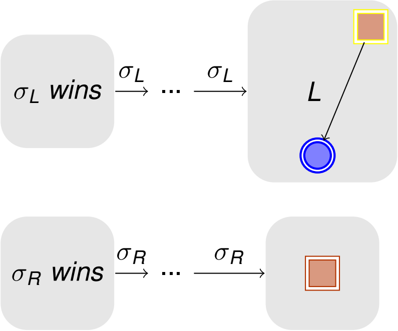
Build a strategy using σ_L and σ_R



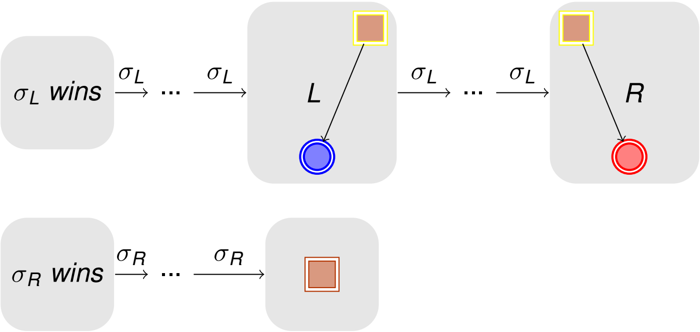
Build a strategy using σ_L and σ_R



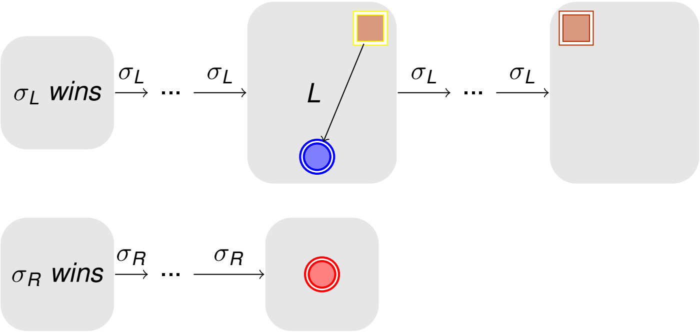
Build a strategy using σ_L and σ_R



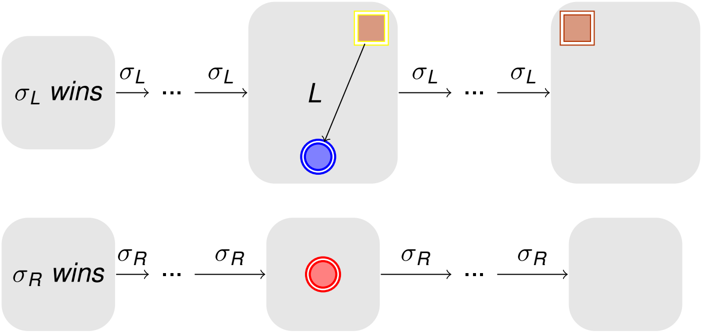
Build a strategy using σ_L and σ_R



Build a strategy using σ_L and σ_R



Build a strategy using σ_L and σ_R



Finding infinite path in a RBN

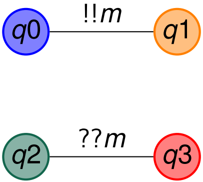
- Once a local strategy fixed for Player 2, we obtain a Reconfigurable Broadcast Network (RBN)
- We can compute its set of reachable sets
- We know there exists reachable configuration, with 'a lot' of nodes as desired in the reachable states
- We will find infinite path in such a RBN using Vector Addition System with States (VASS)

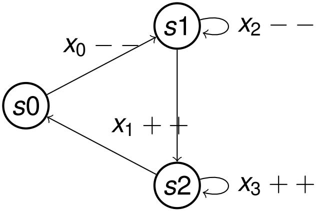
Theorem [Kosaraju & Sullivan 1988]

Detecting positive cycles in VASS can be done in PTIME.

Encoding RBN in a VASS

- Compute the reachable states
- Encode each transitions with reachable states into the VASS
- Check for positive cycles sin the VASS





Complexity of game problem for parity RBN

Theorem

The game problem for parity RBN is in CO-NP.

Idea of the proof:

- Guess a local strategy for Player 2
- Check if it is winning for any configurations against any strategies of Player 1
  - Basically, if the VASS has a positive cycle it is not winning
  - This can be done in PTIME
- If the local strategy is winning then the answer to the game problem is NO
- Furthermore we know that local strategies are enough for Player 2, hence if the answer to the game is NO there is a winning local strategy for Player 2

Outline

Solving $\text{REACH}^{=1}_{\text{max}}$

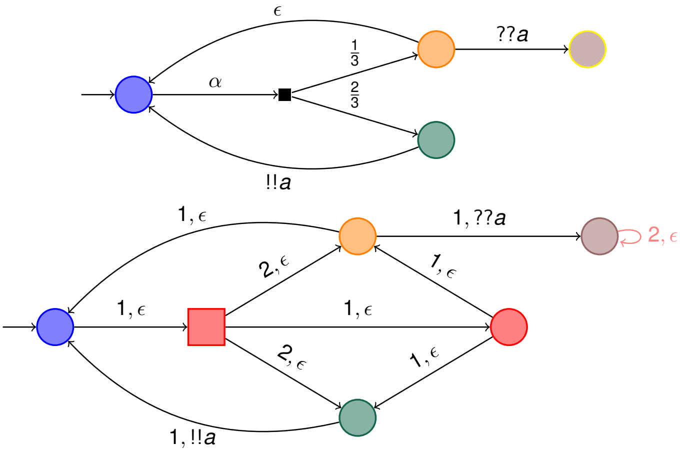
Meaning of $\text{REACH}^{=1}_{\text{max}}$: For some number of processes

- there exists a scheduler π that reaches target almost surely
- *i.e.* from any reachable configurations there is a path to target

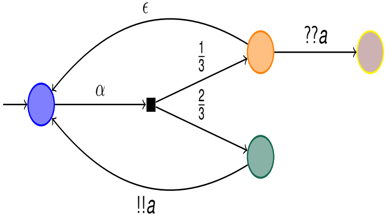
Idea of the reduction

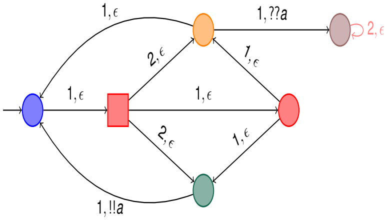
- Let probabilistic choices to player 2
- Force "fairness" for Player 2 by even parities
- Allow Player 2 to abandon choices with odd parity

Reduction $\text{Reach}_{max}^{=1}$



Result for $\text{REACH}^{=1}_{\max}$





Theorem

$\text{REACH}^{=1}_{\max}$ is co-NP-complete.

Proof idea

- Player 2 infinitely often chooses the outcome in $q_p, 2 \Leftrightarrow$ run of probability 0.
- Player 2, after some times, always let Player 1 chooses $\Leftrightarrow q_f$ reached for any reachable configuration.

Solving $\textsc{Reach}^{<1}_{\min}$

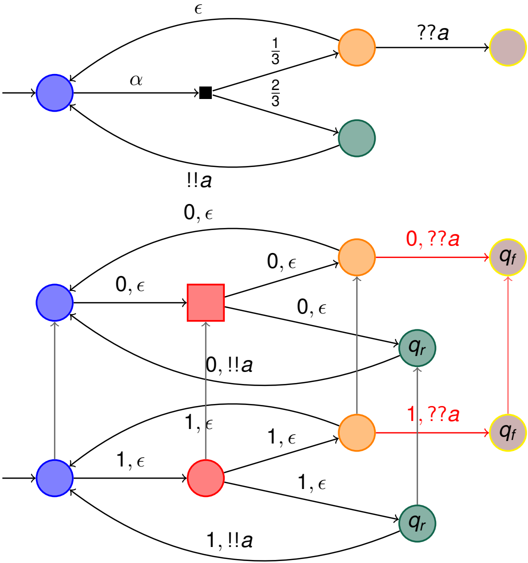
Meaning of $\textsc{Reach}^{<1}_{\min}$: For some number of processes

- there exists a scheduler π that reaches a configuration.
- from this configuration, Player 1 can ensure not to reach target

Idea of the reduction

- Let Player 1 chooses a finite path
- Afterwards, play the game considering probabilistic states as states of Player 2

Reduction $\text{REACH}^{<1}_{min}$



Result for $\text{REACH}_{\text{min}}^{<1}1$

Theorem

$\text{REACH}_{\text{min}}^{<1}$ is CO-NP-complete.

Proof idea:

- Player 1 chooses a finite prefix in the bottom copy
- Then the game can loop infinitely in the top copy

$\Rightarrow$ after a finite prefix q_f is avoided almost surely.

Outline

Conclusion

Results

Problem	$\text{REACH}^{=0}_{\text{min}}$	$\text{REACH}^{<1}_{\text{min}}$	$\text{REACH}^{=1}_{\text{max}}$	others
Complexity	coNP-complete	coNP-complete	coNP-complete	PTIME

Conclusion

Results

Problem	$\text{REACH}^{=0}_{min}$	$\text{REACH}^{<1}_{min}$	$\text{REACH}^{=1}_{max}$	others
Complexity	coNP-complete	coNP-complete	coNP-complete	PTIME

Questions

- What about the quantitative version ?
- Can we do more complex properties than simple reachability ?
- We can solve some kind of parity games, is there a logic characterization corresponding to these games ?
- Does this technique work for networks with other communication primitives ?