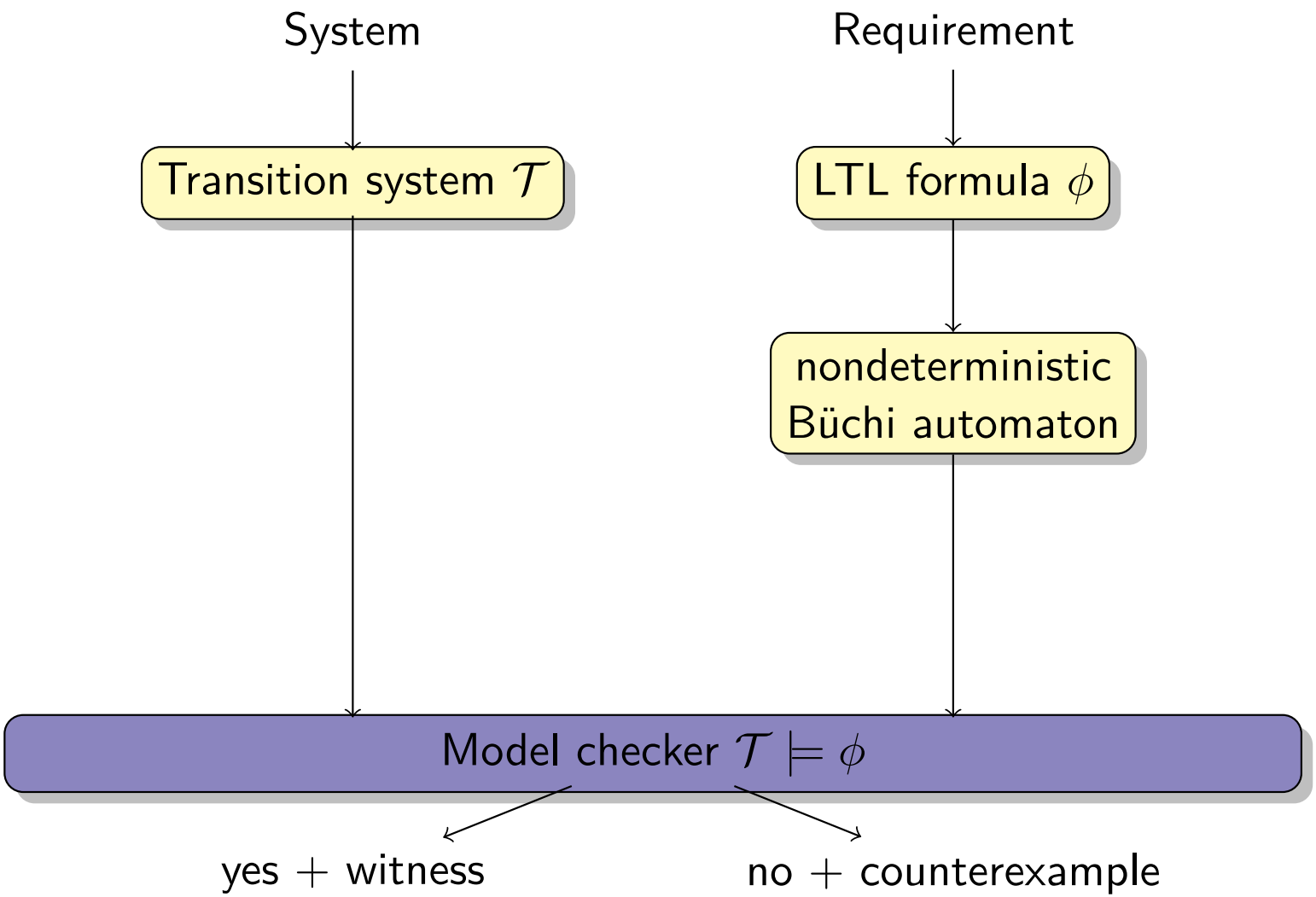
Are Good-for-games Automata Good for Probabilistic Model Checking?

Joachim Klein, **David Müller**, Christel Baier, Sascha Klüppelholz

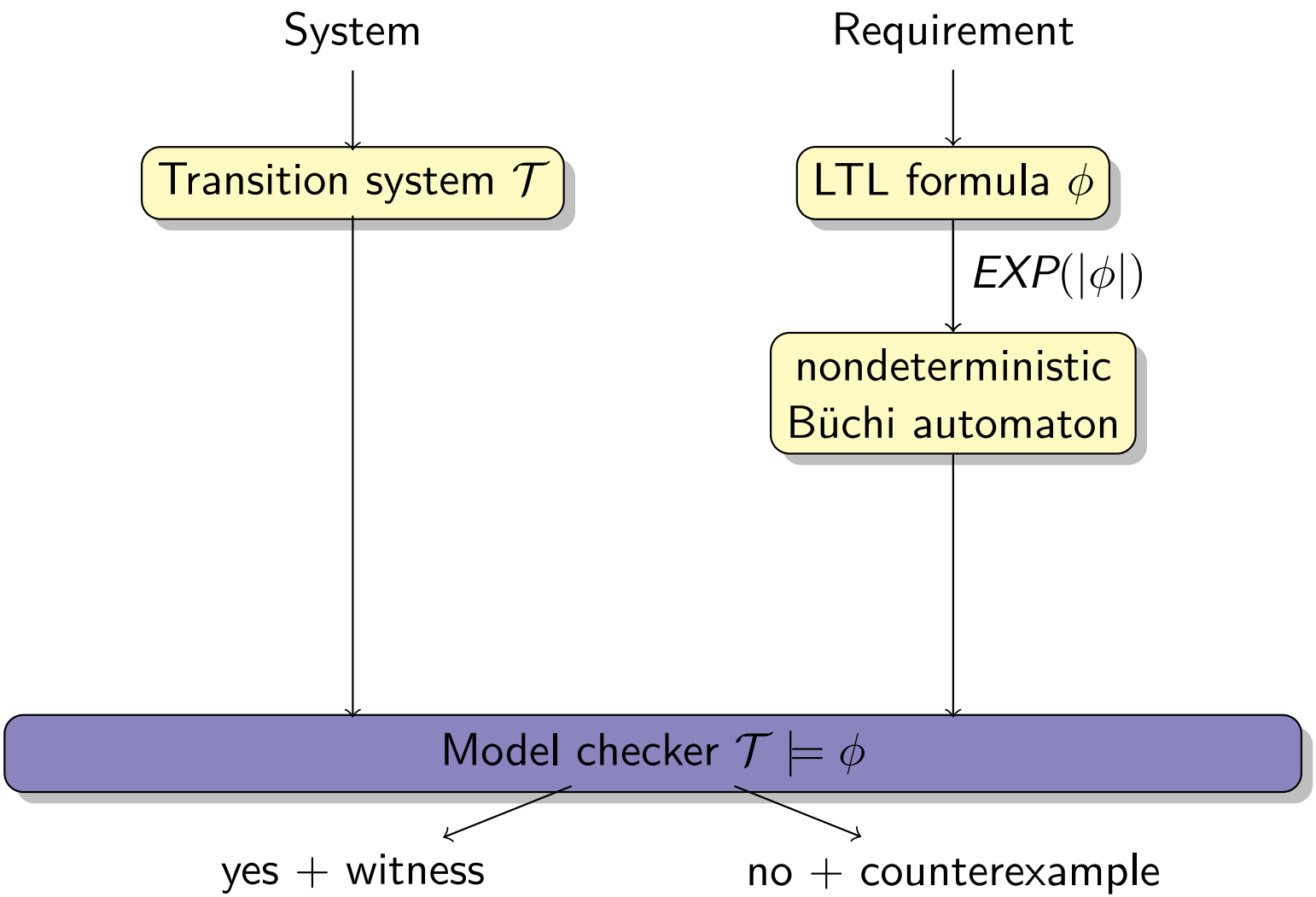
Institute for Theoretical Computer Science
Technische Universität Dresden

March 15, 2014

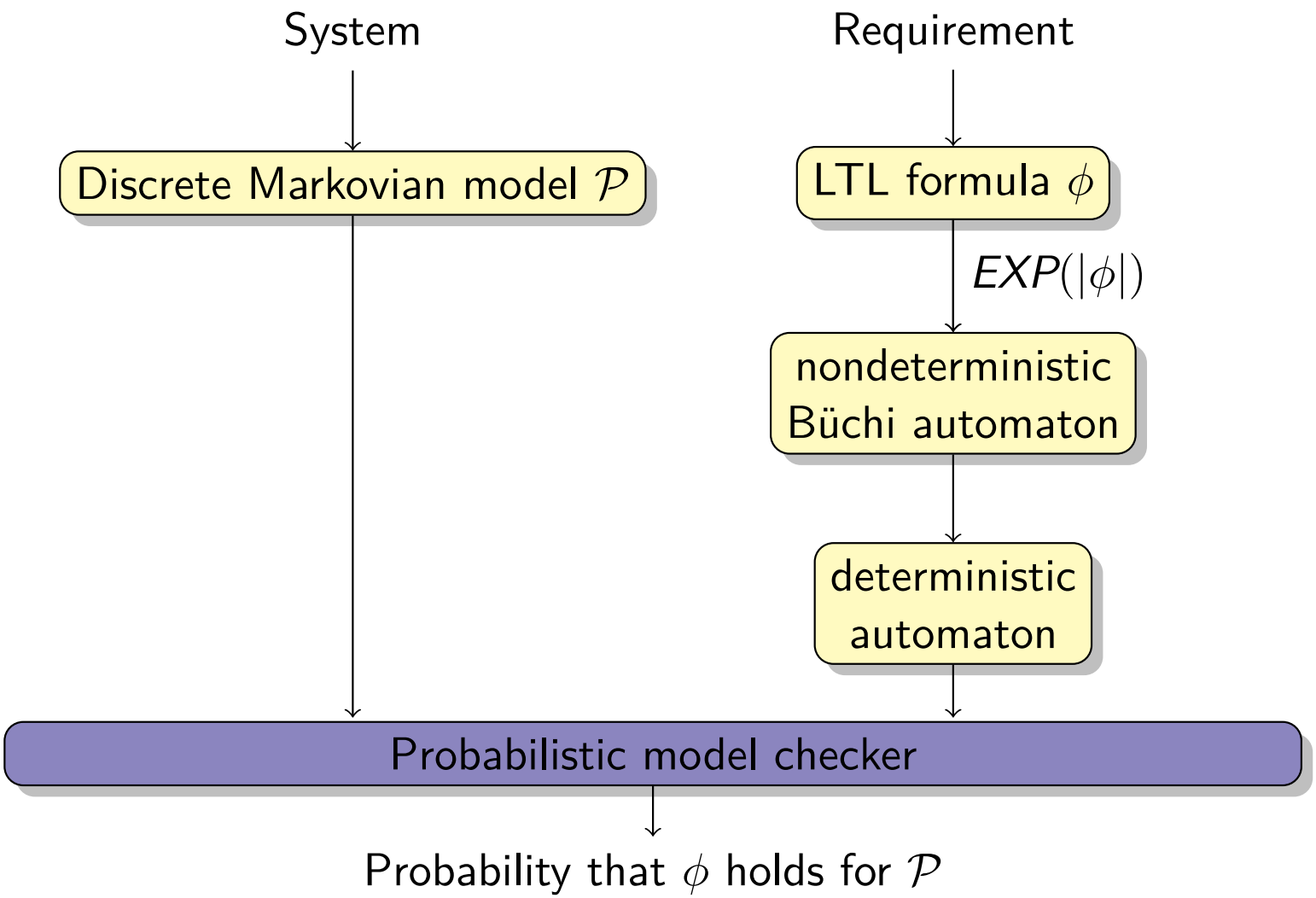
LTL model checking



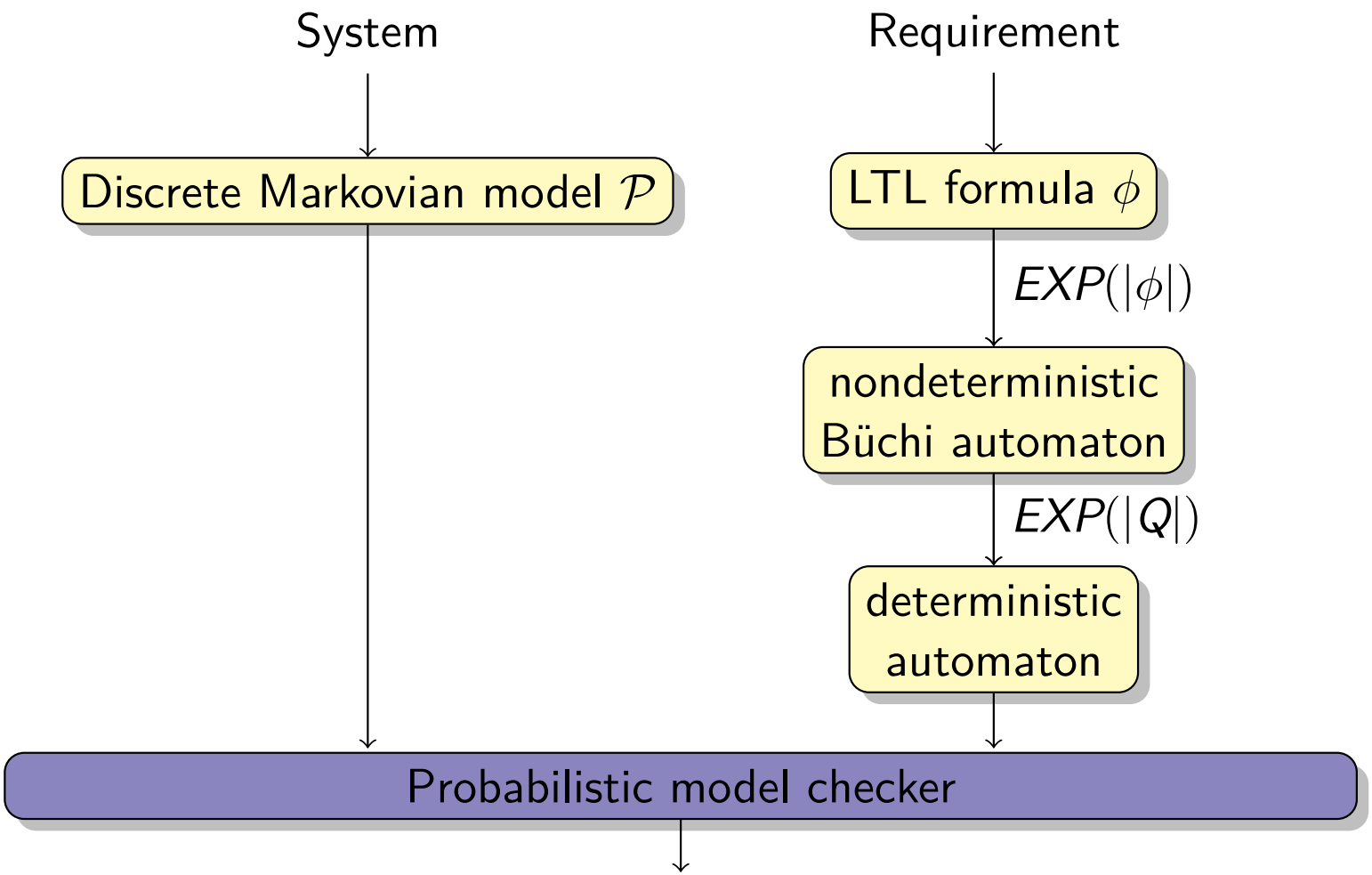
LTL model checking



Probabilistic model checking

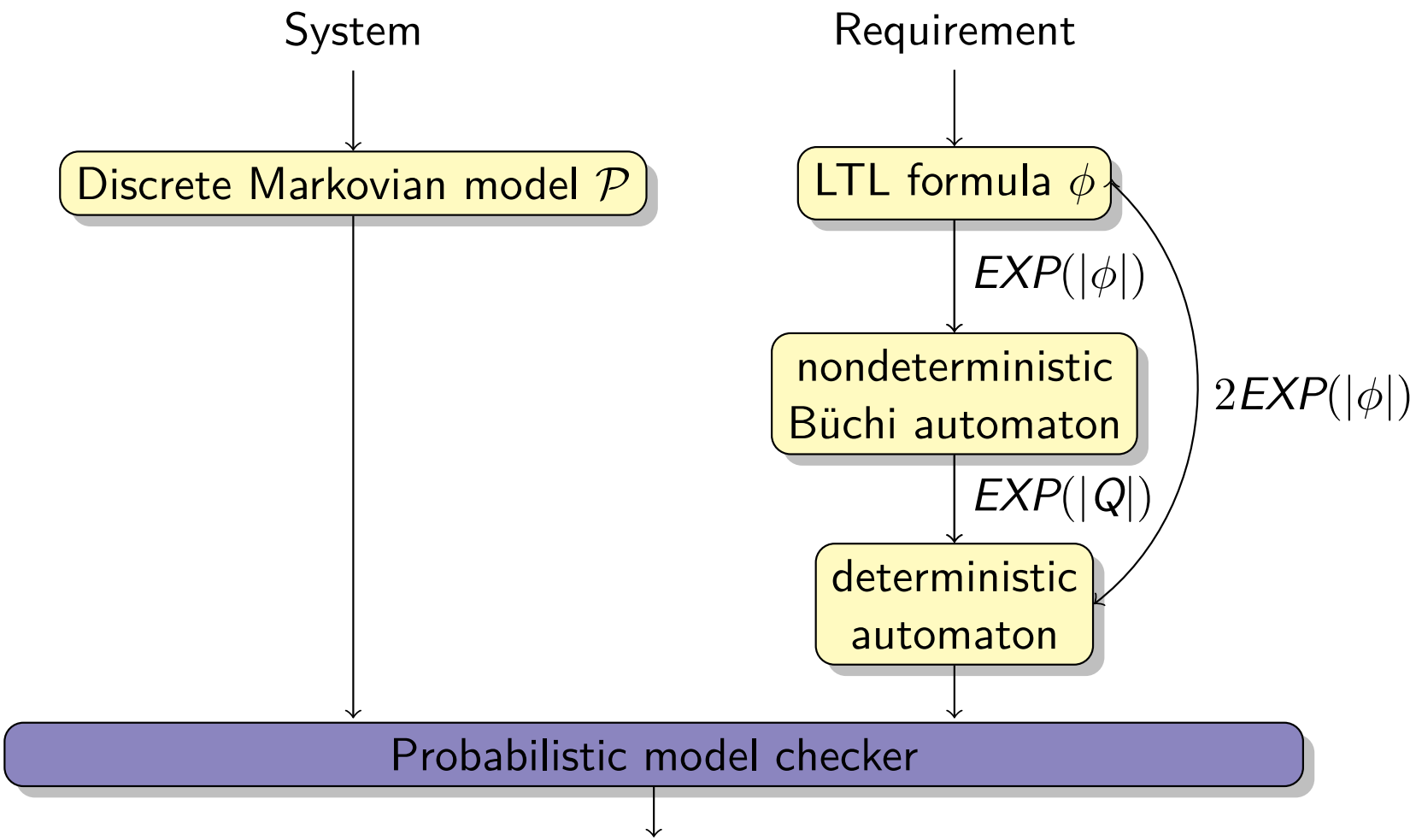


Probabilistic model checking



Lower bound NBA $\rightarrow$ det. automaton by Michel, 1988

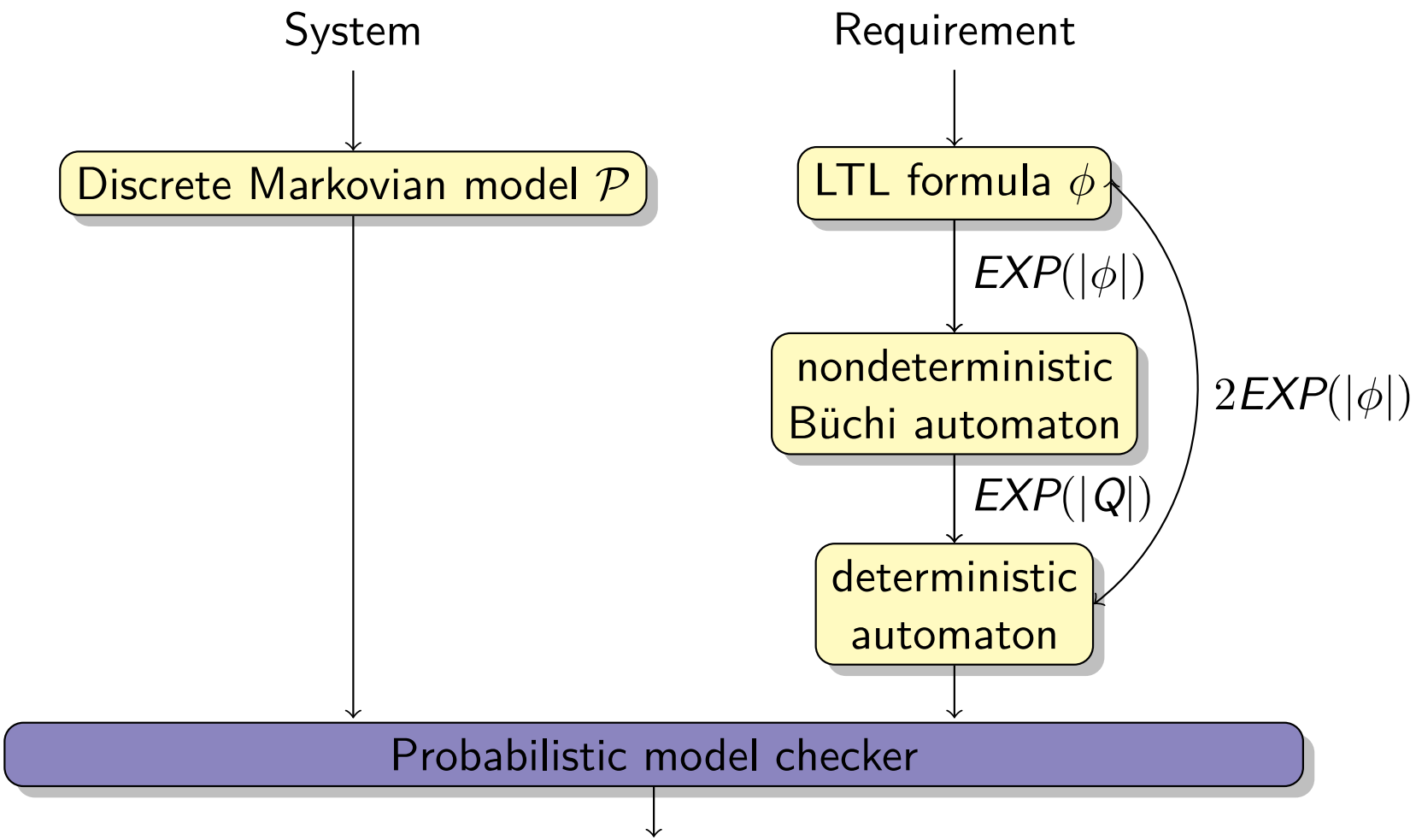
Probabilistic model checking



Lower bound NBA $\rightarrow$ det. automaton by Michel, 1988

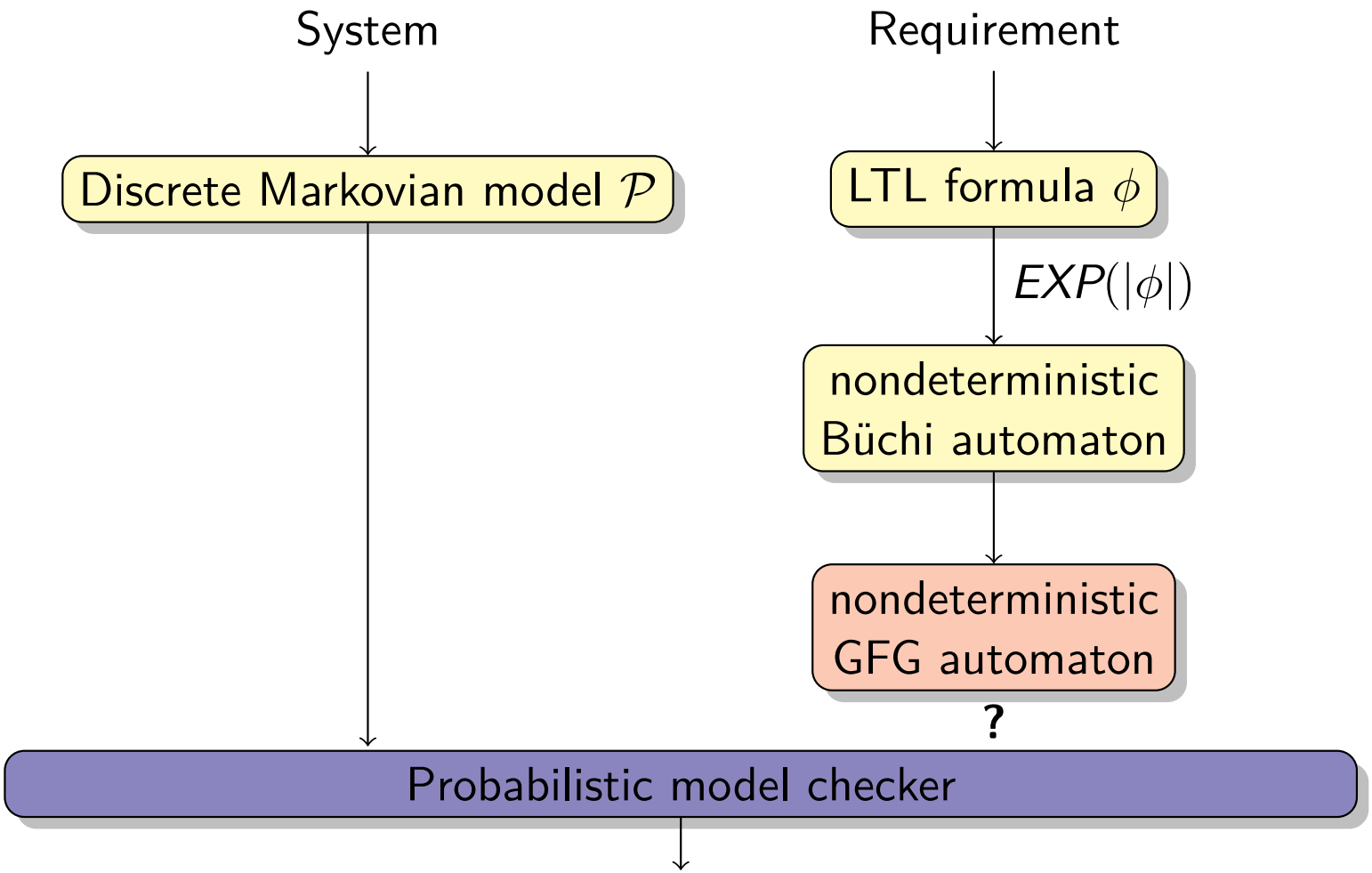
Lower bound LTL $\rightarrow$ det. automaton by Kupferman, Rosenberg, 2011

Probabilistic model checking



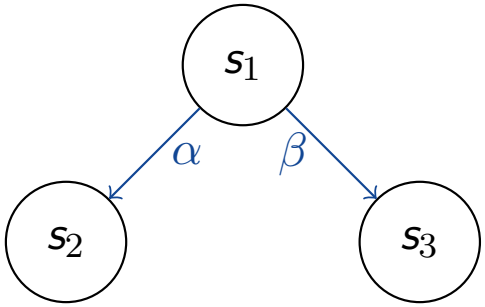
For MDP $\mathcal{P}$ and LTL ϕ $Pr^{max}_{\mathcal{P}}(\phi)$ is 2EXPTIME-complete by Courcoubetis, Yannakakis, 1995

Probabilistic model checking

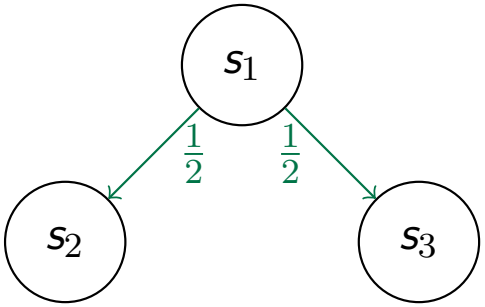


For MDP $\mathcal{P}$ and LTL ϕ $Pr^{max}_{\mathcal{P}}(\phi)$ is 2EXPTIME-complete by Courcoubetis, Yannakakis, 1995

MDP: Probabilism and nondeterminism

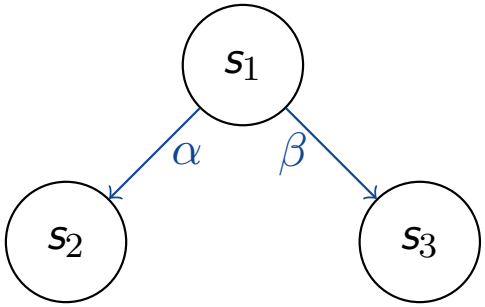


Transition system:
Nondeterminism

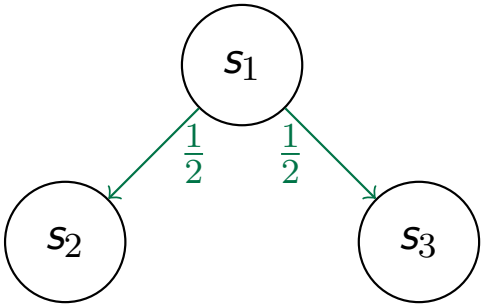


Markov chain:
Probabilism

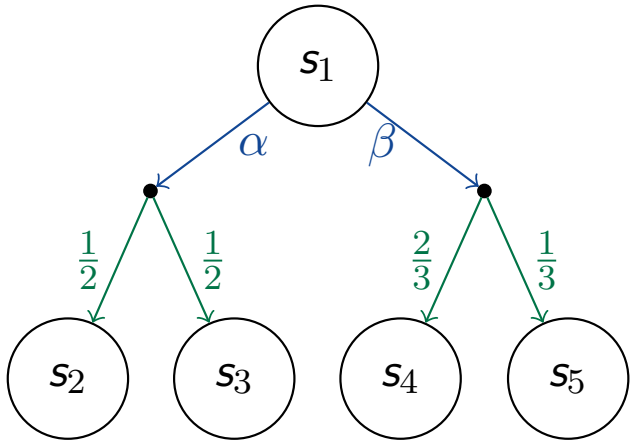
MDP: Probabilism and nondeterminism



Transition system: Nondeterminism

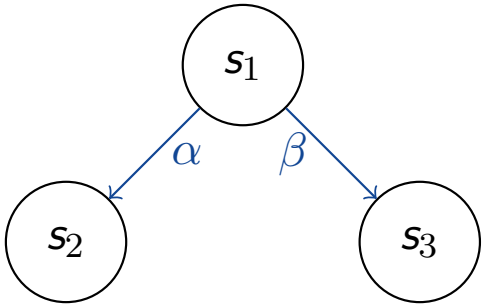


Markov chain: Probabilism

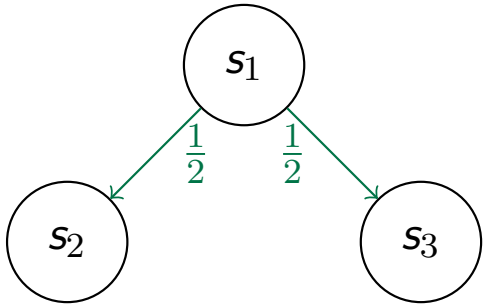


Markov decision process: Nondeterminism and probabilism

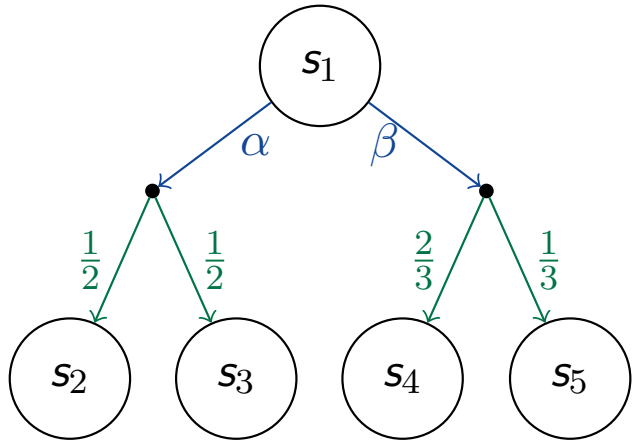
MDP: Probabilism and nondeterminism



Transition system: Nondeterminism



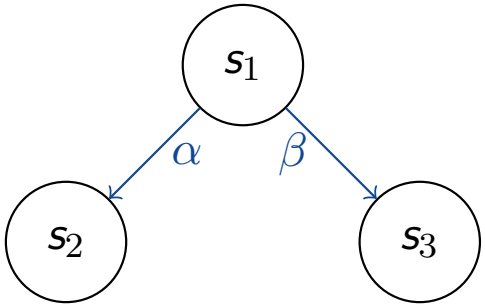
Markov chain: Probabilism



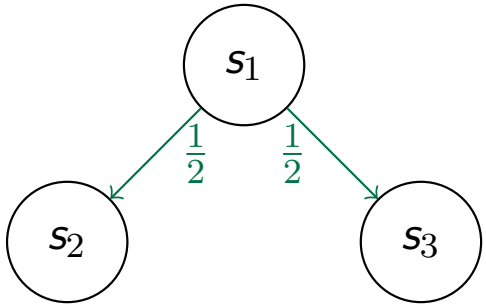
Markov decision process: Nondeterminism and probabilism

Scheduler $\mathfrak{S} : S^+ \to Act$ chooses for given history of states next action

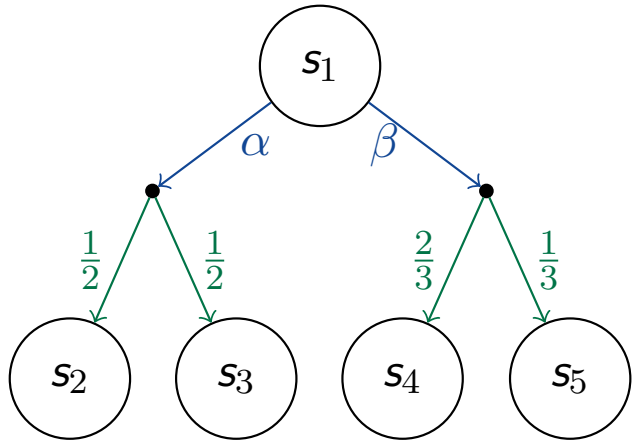
MDP: Probabilism and nondeterminism



Transition system: Nondeterminism



Markov chain: Probabilism

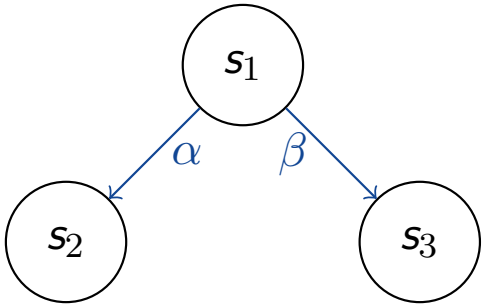


Markov decision process: Nondeterminism and probabilism

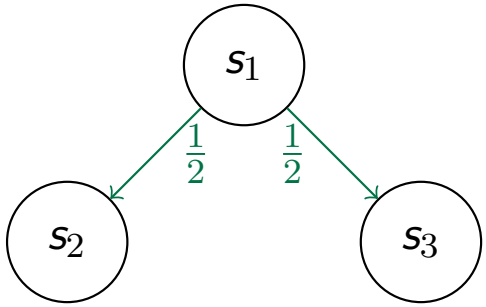
Scheduler $\mathfrak{S} : S^+ \to Act$ chooses for given history of states next action

$Pr_{\mathcal{P}}(\Diamond s_4)$?

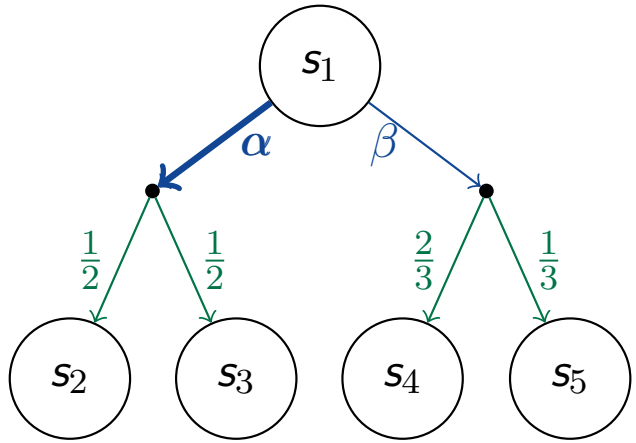
MDP: Probabilism and nondeterminism



Transition system: Nondeterminism



Markov chain: Probabilism

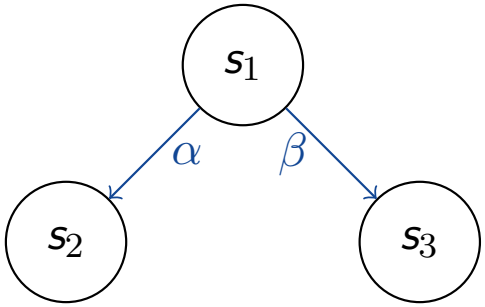


Markov decision process: Nondeterminism and probabilism

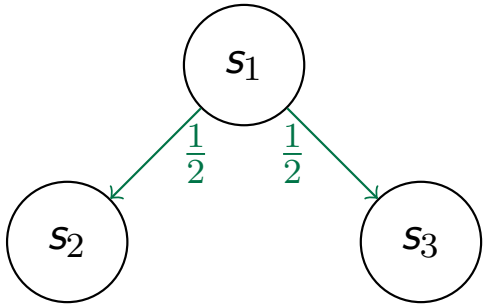
Scheduler $\mathfrak{S} : S^+ \to Act$ chooses for given history of states next action

$Pr_{\mathcal{P}}(\Diamond s_4)$? $\longrightarrow$ $Pr^{min}_{\mathcal{P}}(\Diamond s_4) = 0$

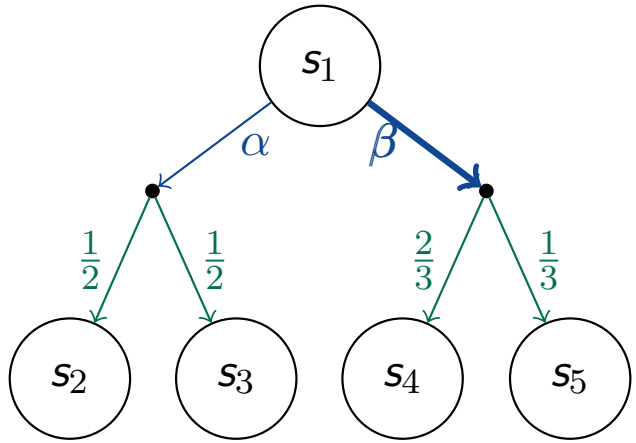
MDP: Probabilism and nondeterminism



Transition system: Nondeterminism



Markov chain: Probabilism



Markov decision process: Nondeterminism and probabilism

Scheduler $\mathfrak{S} : S^+ \to Act$ chooses for given history of states next action

$$Pr_{\mathcal{P}}(\Diamond s_4)\ ? \begin{cases} \to Pr^{max}_{\mathcal{P}}(\Diamond s_4) = \frac{2}{3} \\ \to Pr^{min}_{\mathcal{P}}(\Diamond s_4) = 0 \end{cases}$$

Good-for-games automata

- Property of nondeterministic automata

Good-for-games automata

- Property of nondeterministic automata
- defined in Henzinger, Piterman, 2006 “Solving games without determinization”

Good-for-games automata

- Property of nondeterministic automata
- defined in Henzinger, Piterman, 2006 “Solving games without determinization”
- Used to avoid determinisation for LTL synthesis

Good-for-games automata

deterministic automata

$w \in \mathcal{L}(\mathcal{A}) \Leftrightarrow$ *unique run is an accepting run*

Nondeterministic automata

$w \in \mathcal{L}(\mathcal{A}) \Leftrightarrow$ *exists an accepting run*

Good-for-games automata

deterministic automata
$w \in \mathcal{L}(\mathcal{A}) \Leftrightarrow$ *unique run is an accepting run*

Good-for-games
exists a strategy s.t.
for all $w \in \mathcal{L}(\mathcal{A})$ *strategy-induced run accepting*

Nondeterministic automata
$w \in \mathcal{L}(\mathcal{A}) \Leftrightarrow$ *exists an accepting run*

Good-for-games

- two player game on the automaton:

Good-for-games

- two player game on the automaton:
  - *Player* 1 chooses *symbols* constructing a word

Good-for-games

- two player game on the automaton:
  - *Player* 1 chooses *symbols* constructing a word
  - *Player* 0 chooses *next states* constructing a run

Good-for-games

- two player game on the automaton:
  - *Player* 1 chooses *symbols* constructing a word
  - *Player* 0 chooses *next states* constructing a run
- there exists a strategy $D : (Q \times \Sigma)^* \to Q$ (for player 0)

Good-for-games

- two player game on the automaton:
  - *Player* 1 chooses *symbols* constructing a word
  - *Player* 0 chooses *next states* constructing a run
- there exists a strategy $D : (Q \times \Sigma)^* \to Q$ (for player 0)
- every word $w \in \Sigma^\omega$ induces a D-run

Good-for-games

- two player game on the automaton:
  - *Player* 1 chooses *symbols* constructing a word
  - *Player* 0 chooses *next states* constructing a run
- there exists a strategy $D : (Q \times \Sigma)^* \to Q$ (for player 0)
- every word $w \in \Sigma^\omega$ induces a D-run

Definition

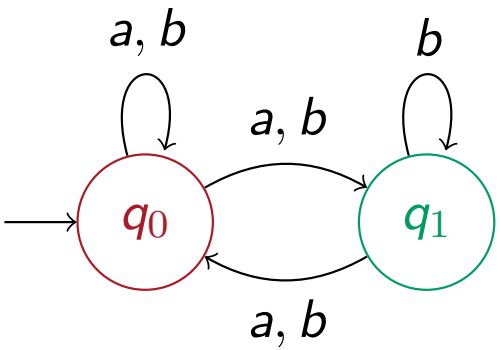
An ω-automaton is *good-for-games* if there exists a strategy, s.t. for every word $w \in \mathcal{L}(\mathcal{A})$ the induced run is accepting.

Good-for-games

- two player game on the automaton:
  - *Player* 1 chooses *symbols* constructing a word
  - *Player* 0 chooses *next states* constructing a run
- there exists a strategy $D : (Q \times \Sigma)^* \to Q$ (for player 0)
- every word $w \in \Sigma^\omega$ induces a D-run

Definition

An ω-automaton is *good-for-games* if there exists a strategy, s.t. for every word $w \in \mathcal{L}(\mathcal{A})$ the induced run is accepting.

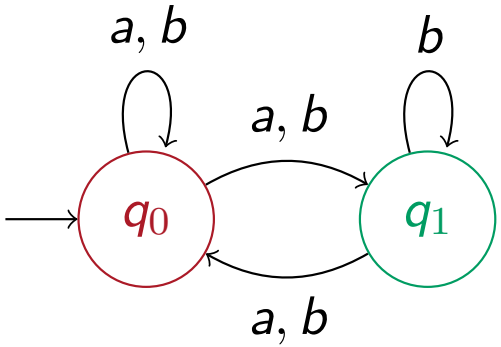


$$Acc = \Diamond\Box\neg q_0 \wedge \Box\Diamond q_1$$

Run π is accepting $\Leftrightarrow q_0 \notin \mathit{inf}(\pi) \wedge q_1 \in \mathit{inf}(\pi)$

$$\mathcal{L}(\mathcal{A}) = \mathcal{L}((a+b)^* b^\omega)$$

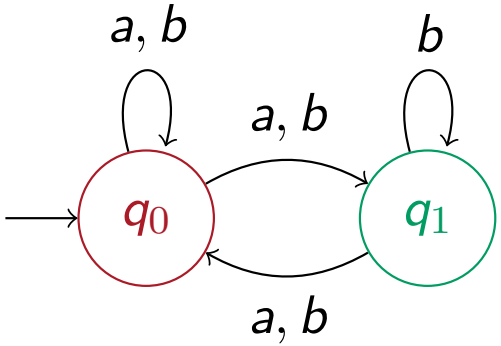
Strategy



GFG strategy

$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

Strategy



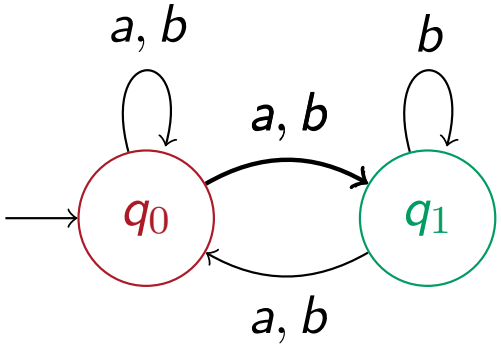
GFG strategy

$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

The word $abab^\omega$ induces the following play:

word	a
play	$q_0 a$

Strategy



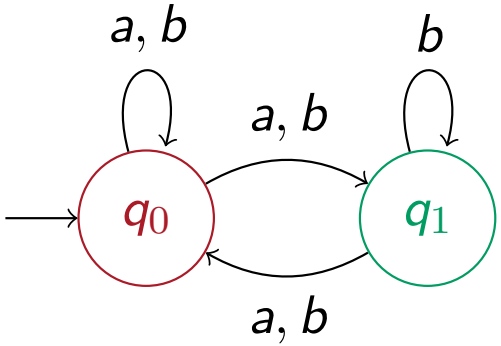
GFG strategy

$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

The word $abab^\omega$ induces the following play:

word	a		
play	$q_0 a$	q_1	

Strategy



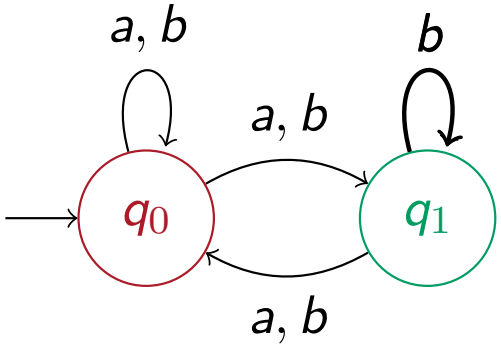
GFG strategy

$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

The word $abab^\omega$ induces the following play:

word	a	b
play	$q_0 a$	$q_1 b$

Strategy



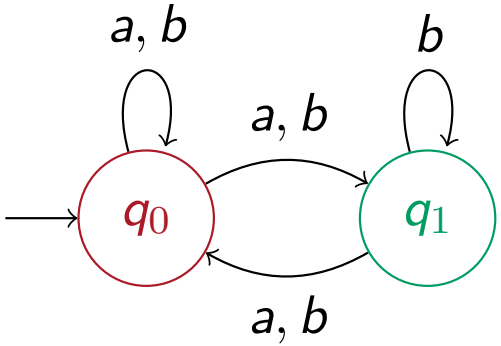
GFG strategy

$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

The word $abab^\omega$ induces the following play:

word	a	b	
play	$q_0 a$	$q_1 b$	q_1

Strategy



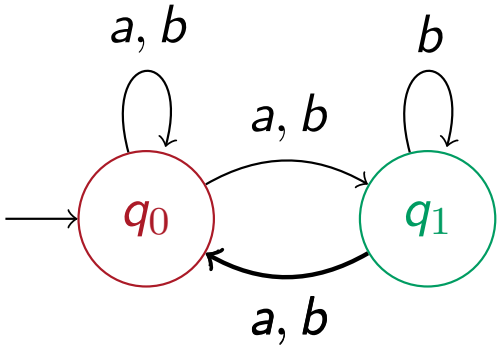
GFG strategy

$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

The word $abab^\omega$ induces the following play:

word	a	b	a
play	$q_0 a$	$q_1 b$	$q_1 a$

Strategy



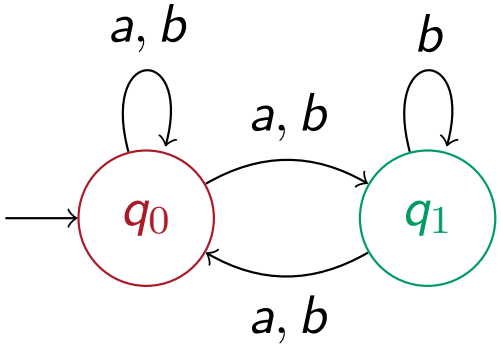
GFG strategy

$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

The word $abab^\omega$ induces the following play:

word	a	b	a	
play	$q_0 a$	$q_1 b$	$q_1 a$	q_0

Strategy



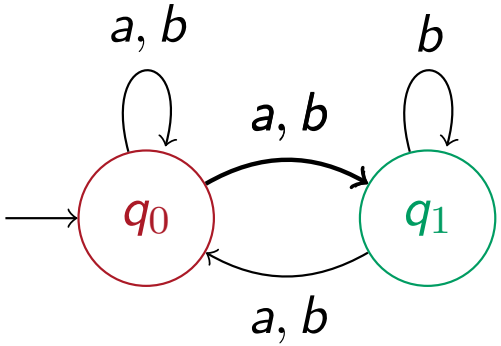
GFG strategy

$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

The word $abab^\omega$ induces the following play:

word	a	b	a	b
play	$q_0\, a$	$q_1\, b$	$q_1\, a$	$q_0\ b$

Strategy



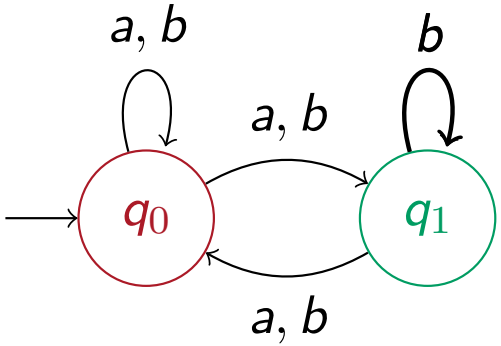
GFG strategy

$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

The word $abab^\omega$ induces the following play:

word	a	b	a	b		
play	$q_0\, a$	$q_1\, b$	$q_1\, a$	q_0	b	q_1

Strategy



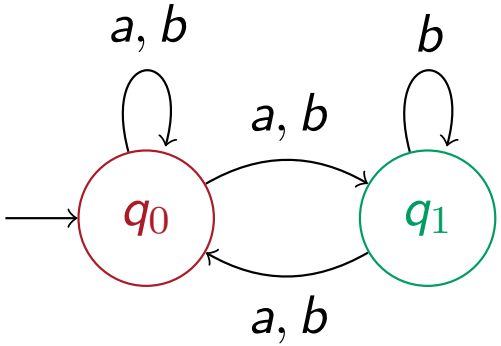
GFG strategy

$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

The word $abab^\omega$ induces the following play:

word	a	b	a	b^ω	
play	$q_0 a$	$q_1 b$	$q_1 a$	$q_0 (b$	$q_1)^\omega$

Strategy



GFG strategy

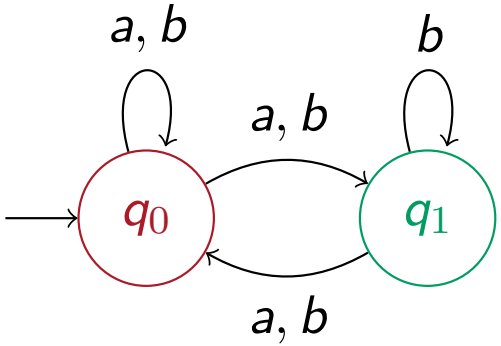
$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

The word $abab^\omega$ induces the following play:

word	a	b	a	b^ω	
play	$q_0 a$	$q_1 b$	$q_1 a$	$q_0 (b$	$q_1)^\omega$

- For an accepted word player 1 has to choose b from some point on

Strategy



GFG strategy

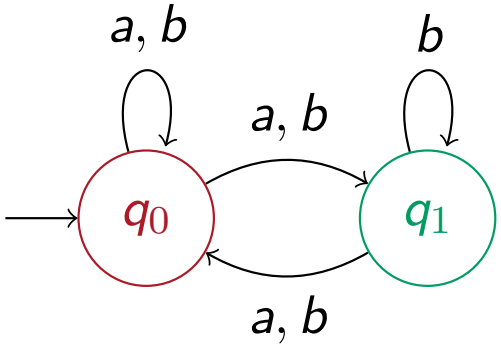
$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

The word $abab^\omega$ induces the following play:

word	a	b	a	b^ω	
play	$q_0 a$	$q_1 b$	$q_1 a$	$q_0 (b$	$q_1)^\omega$

- For an accepted word player 1 has to choose b from some point on
- Every induced D-run remains in q_1 from some point on

Strategy



GFG strategy

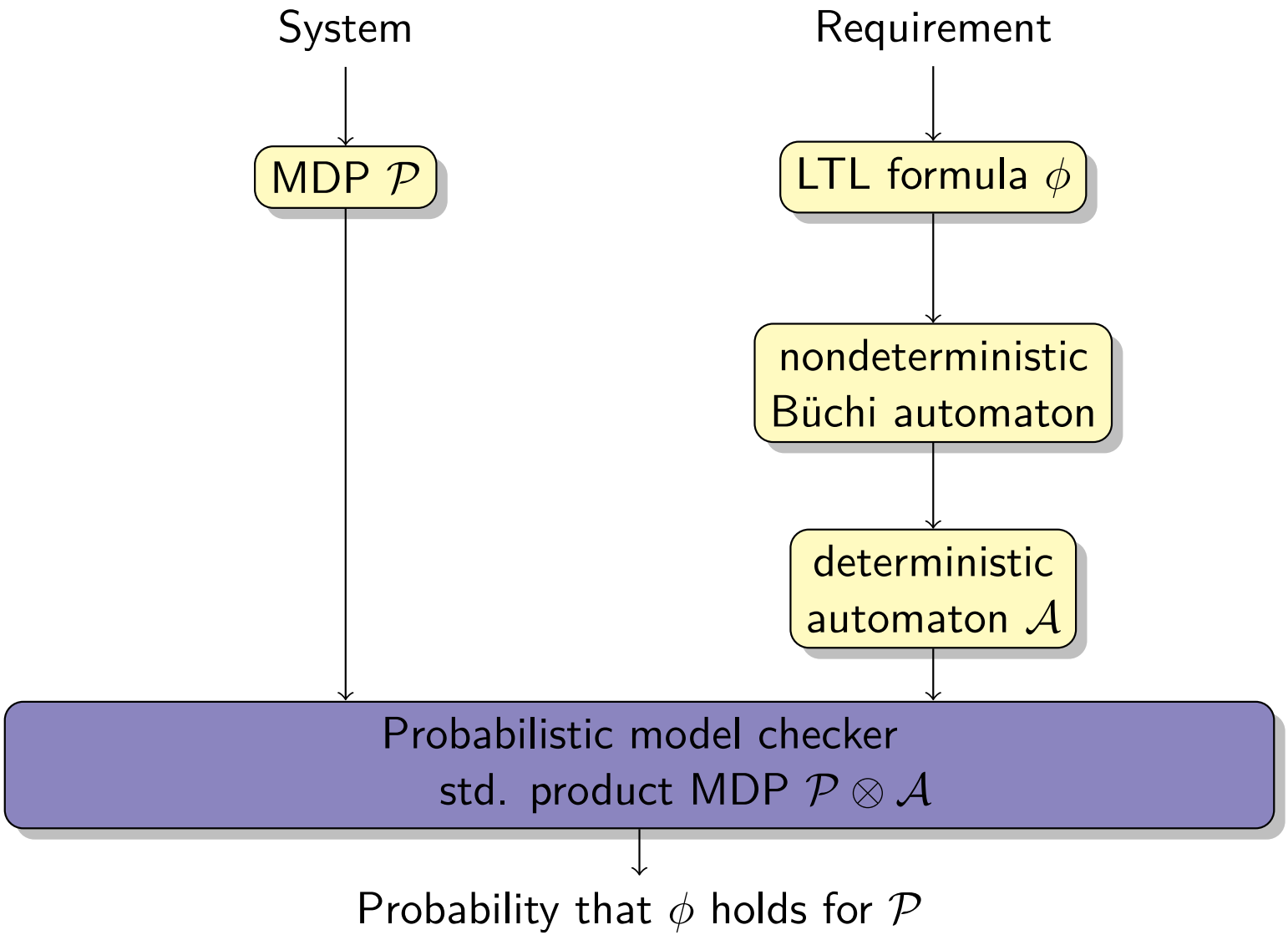
$$D(\pi) = \begin{cases} q_0 & \text{if } \pi \text{ ends with } q_1 a \\ q_1 & \text{otherwise} \end{cases}$$

The word $abab^\omega$ induces the following play:

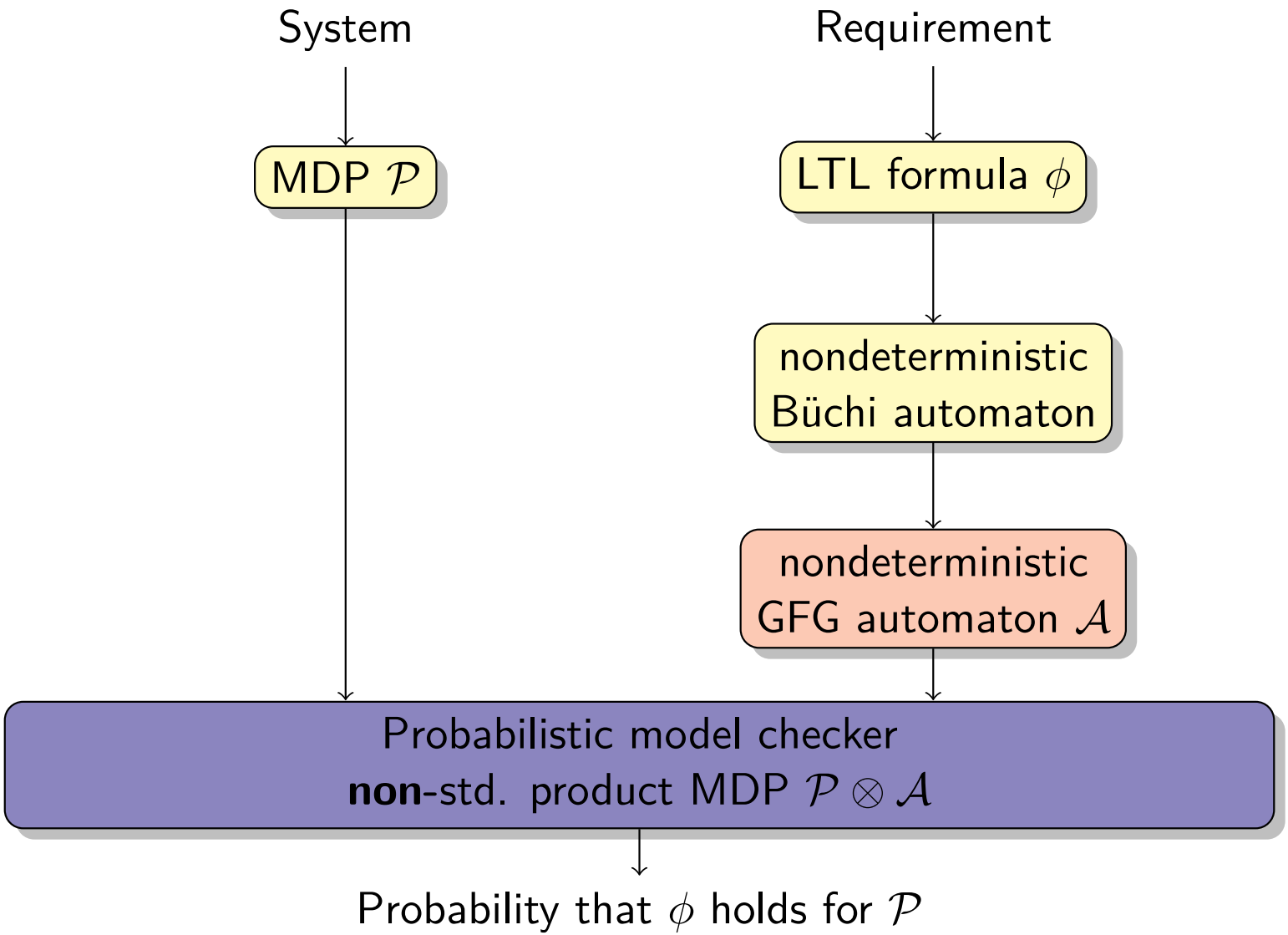
word	a	b	a	b^ω	
play	$q_0 a$	$q_1 b$	$q_1 a$	$q_0 (b$	$q_1)^\omega$

- For an accepted word player 1 has to choose b from some point on
- Every induced D-run remains in q_1 from some point on
- Every induced D-run is accepting

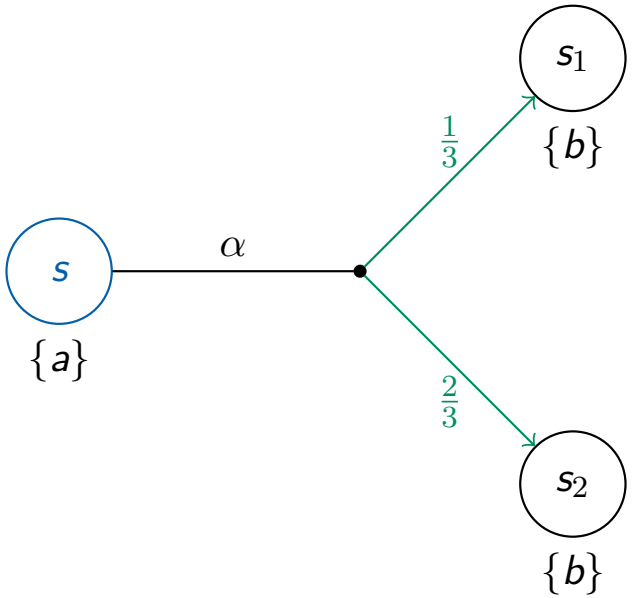
Probabilistic model checking

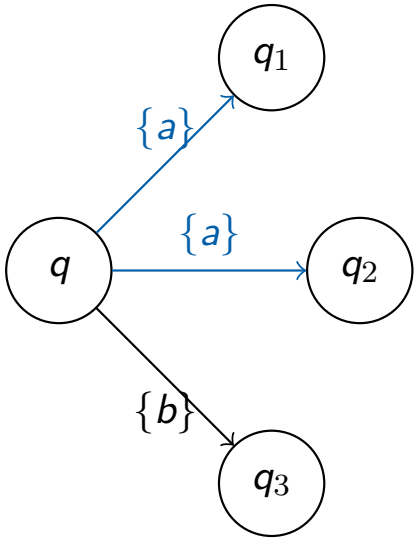


Probabilistic model checking

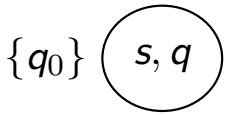


Product MDP: $\mathcal{P} \otimes \mathcal{A}$

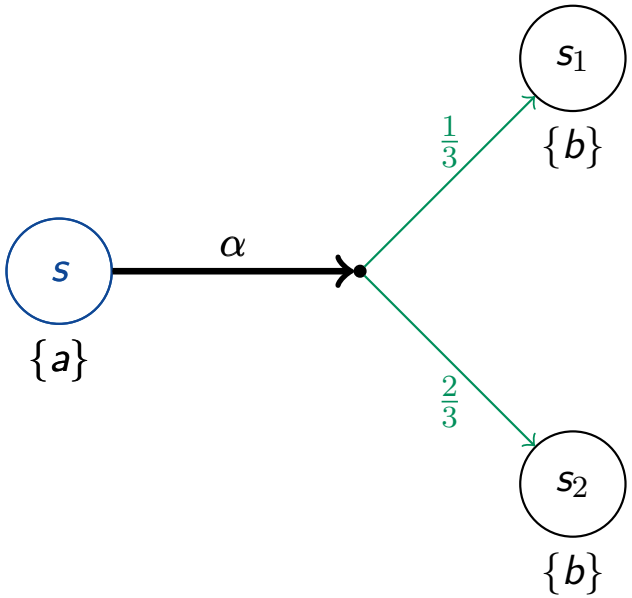


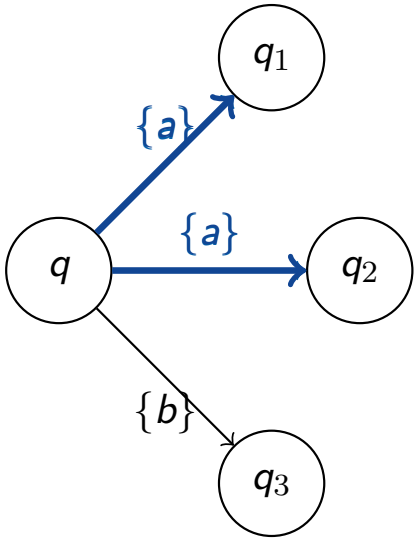


For state (s, q) in $\mathcal{P} \otimes \mathcal{A}$:



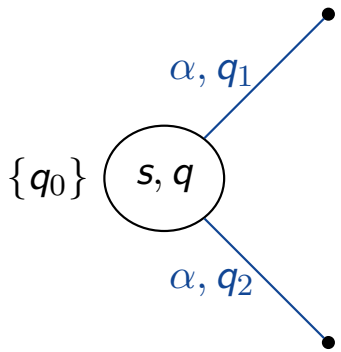
Product MDP: $\mathcal{P} \otimes \mathcal{A}$



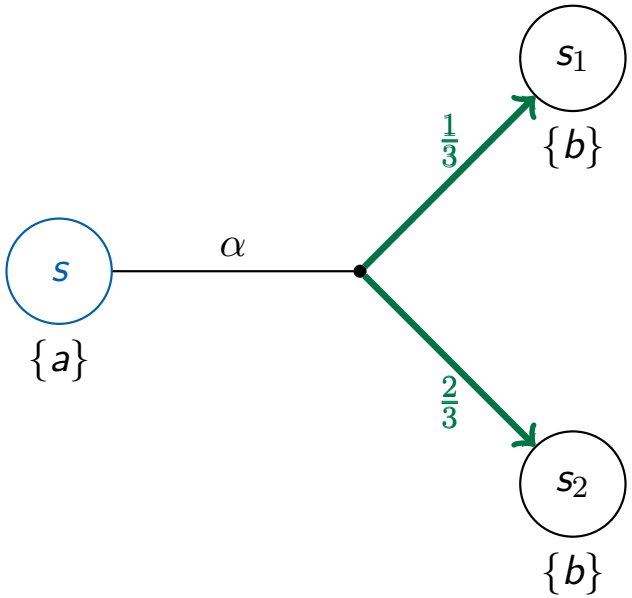


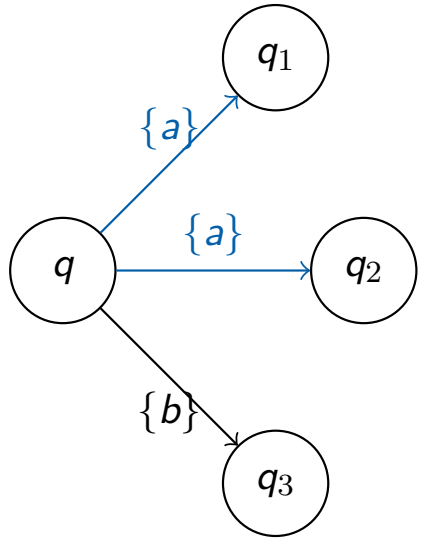
For state (s, q) in $\mathcal{P} \otimes \mathcal{A}$:

- enabled actions: (α, q') for $\alpha \in Act(s)$ and $q' \in \delta(q, L(s))$



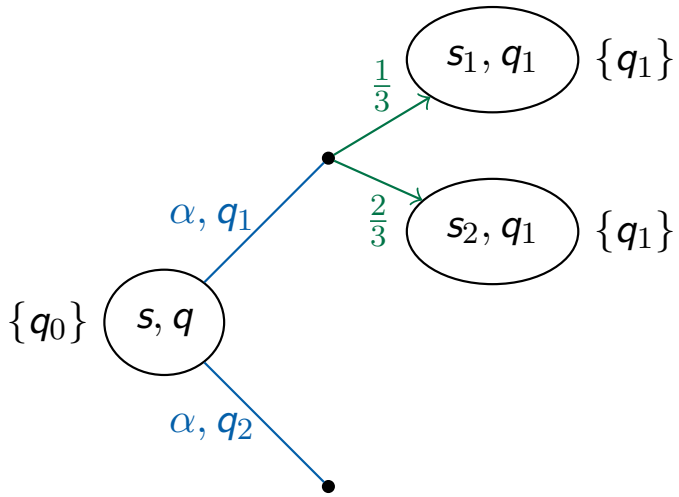
Product MDP: $\mathcal{P} \otimes \mathcal{A}$



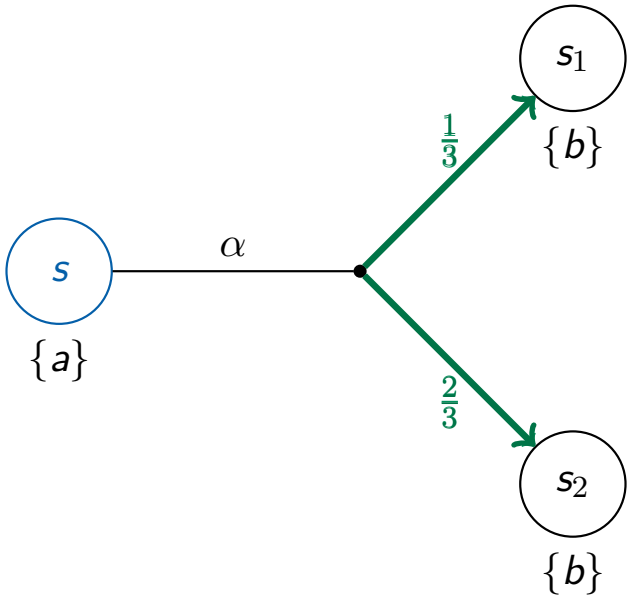


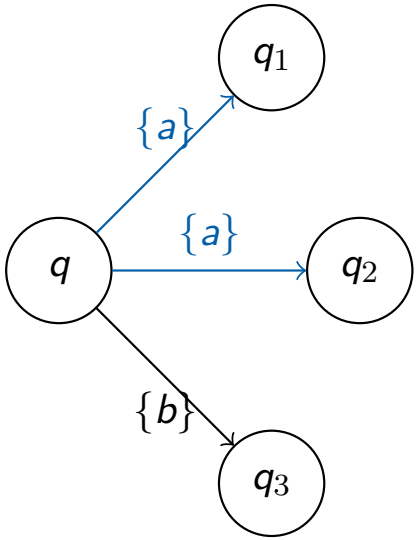
For state (s, q) in $\mathcal{P} \otimes \mathcal{A}$:

- enabled actions: (α, q') for $\alpha \in Act(s)$ and $q' \in \delta(q, L(s))$
- If action α, q' chosen, take over probability distributions of s in $\mathcal{P}$



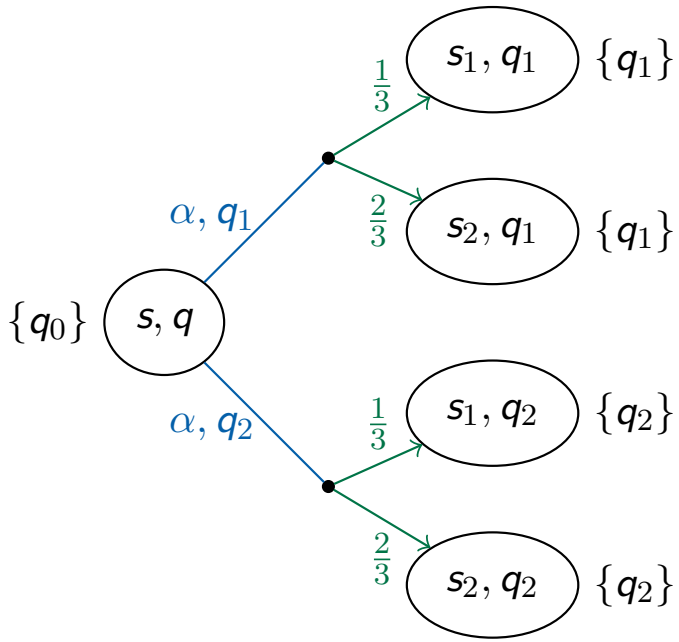
Product MDP: $\mathcal{P} \otimes \mathcal{A}$





For state (s, q) in $\mathcal{P} \otimes \mathcal{A}$:

- enabled actions: (α, q') for $\alpha \in Act(s)$ and $q' \in \delta(q, L(s))$
- If action α, q' chosen, take over probability distributions of s in $\mathcal{P}$



MDP analysis with GFG

Theorem

Let $\mathcal{P}$ be an MDP and $\mathcal{A}$ be an GFG ω-automaton. Then

$$Pr^{max}_{\mathcal{P}\otimes\mathcal{A}}(\langle s_0, q_0\rangle \models Acc) = Pr^{max}_{\mathcal{P}}(s_0 \models \mathcal{L}(\mathcal{A}))$$

MDP analysis with GFG

Theorem

Let $\mathcal{P}$ be an MDP and $\mathcal{A}$ be an GFG ω-automaton. Then

$$Pr^{max}_{\mathcal{P}\otimes\mathcal{A}}(\langle s_0, q_0\rangle \models Acc) = Pr^{max}_{\mathcal{P}}(s_0 \models \mathcal{L}(\mathcal{A}))$$

Proof idea of "$\geq$":

- scheduler in $\mathcal{P}$ maximising $\mathcal{L}(\mathcal{A})$ / GFG winning strategy $\Big\}$ scheduler in $\mathcal{P}\otimes\mathcal{A}$ maximising Acc

MDP analysis with GFG

Theorem

Let $\mathcal{P}$ be an MDP and $\mathcal{A}$ be an GFG ω-automaton. Then

$$Pr^{max}_{\mathcal{P}\otimes\mathcal{A}}(\langle s_0, q_0\rangle \models Acc) = Pr^{max}_{\mathcal{P}}(s_0 \models \mathcal{L}(\mathcal{A}))$$

Proof idea of "$\geq$":

- scheduler in $\mathcal{P}$ maximising $\mathcal{L}(\mathcal{A})$ / GFG winning strategy $\Big\}$ scheduler in $\mathcal{P}\otimes\mathcal{A}$ maximising Acc
- GFG winning strategy tries to maximise acceptance

MDP analysis with GFG

Theorem

Let $\mathcal{P}$ be an MDP and $\mathcal{A}$ be an GFG ω-automaton. Then

$$Pr^{max}_{\mathcal{P}\otimes\mathcal{A}}(\langle s_0, q_0\rangle \models Acc) = Pr^{max}_{\mathcal{P}}(s_0 \models \mathcal{L}(\mathcal{A}))$$

Proof idea of "$\geq$":

- scheduler in $\mathcal{P}$ maximising $\mathcal{L}(\mathcal{A})$ / GFG winning strategy } scheduler in $\mathcal{P}\otimes\mathcal{A}$ maximising Acc
- GFG winning strategy tries to maximise acceptance
- scheduler in $\mathcal{P}$ maximises $Pr^{\mathfrak{S}}_{\mathcal{P}}(\mathcal{L}(\mathcal{A}))$

MDP analysis with GFG

Theorem

Let $\mathcal{P}$ be an MDP and $\mathcal{A}$ be an GFG ω-automaton. Then

$$Pr^{max}_{\mathcal{P}\otimes\mathcal{A}}(\langle s_0, q_0\rangle \models Acc) = Pr^{max}_{\mathcal{P}}(s_0 \models \mathcal{L}(\mathcal{A}))$$

Proof idea of "$\geq$":

- scheduler in $\mathcal{P}$ maximising $\mathcal{L}(\mathcal{A})$ / GFG winning strategy $\Big\}$ scheduler in $\mathcal{P}\otimes\mathcal{A}$ maximising Acc
- GFG winning strategy tries to maximise acceptance
- scheduler in $\mathcal{P}$ maximises $Pr^{\mathfrak{S}}_{\mathcal{P}}(\mathcal{L}(\mathcal{A}))$
- scheduler in $\mathcal{P}\otimes\mathcal{A}$ maximises $Pr^{\mathfrak{S}}_{\mathcal{P}\otimes\mathcal{A}}(Acc)$

MDP analysis with GFG

Theorem

Let $\mathcal{P}$ be an MDP and $\mathcal{A}$ be an GFG ω-automaton. Then

$$Pr^{max}_{\mathcal{P}\otimes\mathcal{A}}(\langle s_0, q_0\rangle \models Acc) = Pr^{max}_{\mathcal{P}}(s_0 \models \mathcal{L}(\mathcal{A}))$$

Proof idea of "$\geq$":

- scheduler in $\mathcal{P}$ maximising $\mathcal{L}(\mathcal{A})$ / GFG winning strategy $\Big\}$ scheduler in $\mathcal{P}\otimes\mathcal{A}$ maximising Acc
- GFG winning strategy tries to maximise acceptance
- scheduler in $\mathcal{P}$ maximises $Pr^{\mathfrak{S}}_{\mathcal{P}}(\mathcal{L}(\mathcal{A}))$
- scheduler in $\mathcal{P}\otimes\mathcal{A}$ maximises $Pr^{\mathfrak{S}}_{\mathcal{P}\otimes\mathcal{A}}(Acc)$

Proof of "$\leq$" is rather technical

From LTL to GFG

- NBA to GFG nondet. parity automaton with at most $2^n \cdot n^{2n}$ states from Henzinger and Piterman, 2006

From LTL to GFG

- NBA to GFG nondet. parity automaton with at most $2^n \cdot n^{2n}$ states from Henzinger and Piterman, 2006
- HP-algorithm similar to Safra's determinisation

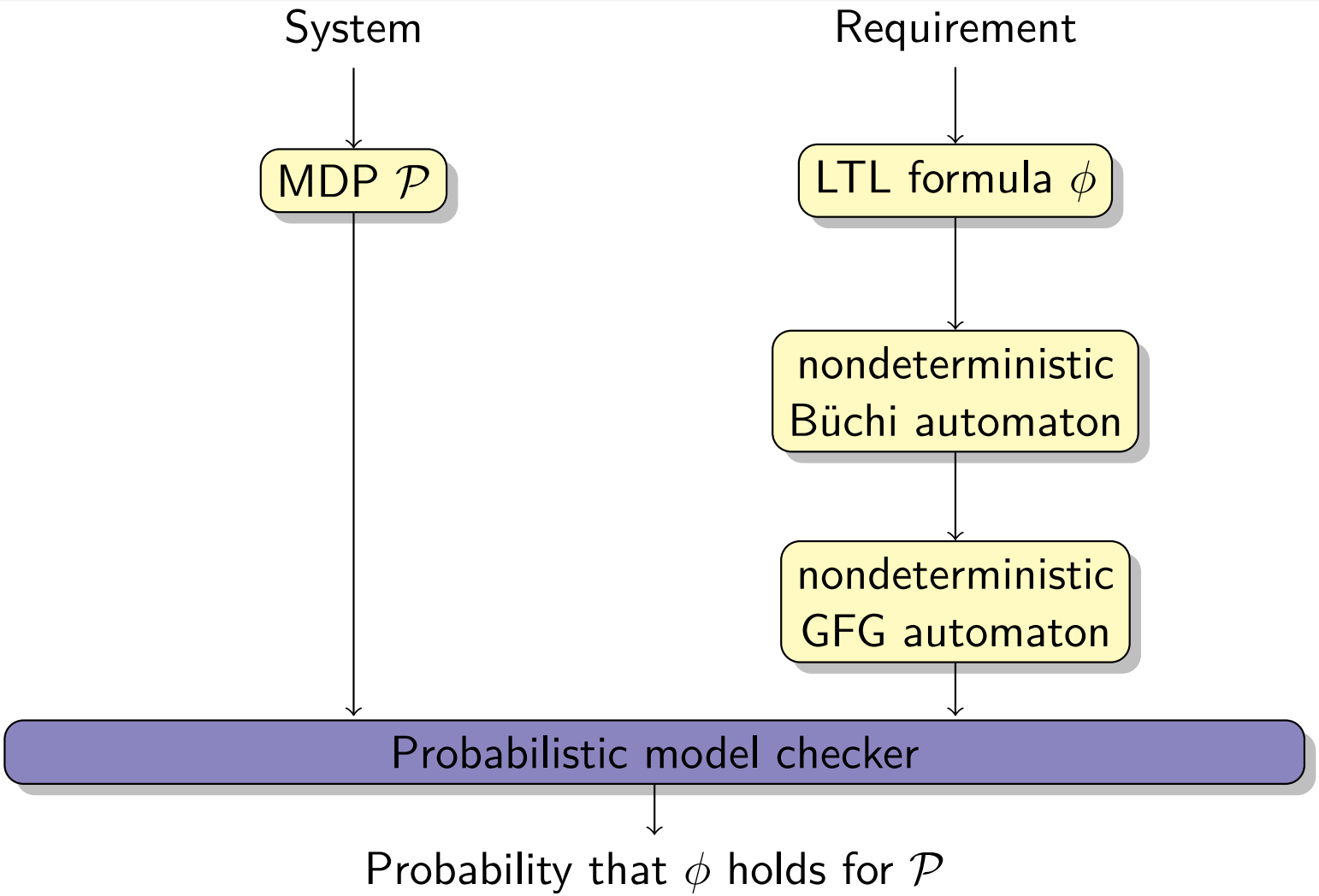
From LTL to GFG

- NBA to GFG nondet. parity automaton with at most $2^n \cdot n^{2n}$ states from Henzinger and Piterman, 2006
- HP-algorithm similar to Safra's determinisation
- parallel powerset constructions, but instead of a tree organization linear organization in tuples

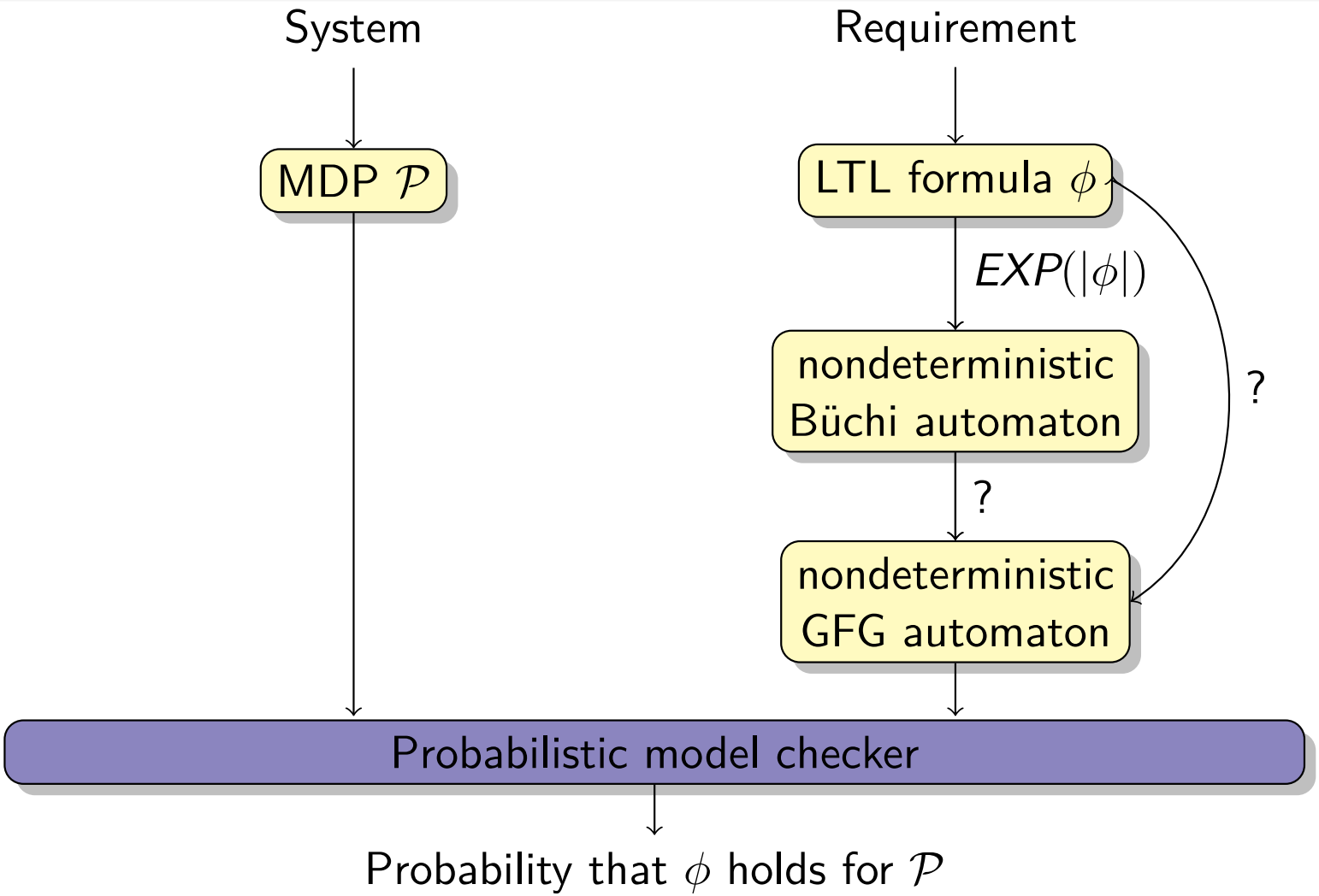
From LTL to GFG

- NBA to GFG nondet. parity automaton with at most $2^n \cdot n^{2n}$ states from Henzinger and Piterman, 2006
- HP-algorithm similar to Safra's determinisation
- parallel powerset constructions, but instead of a tree organization linear organization in tuples
- intended for symbolic implementation

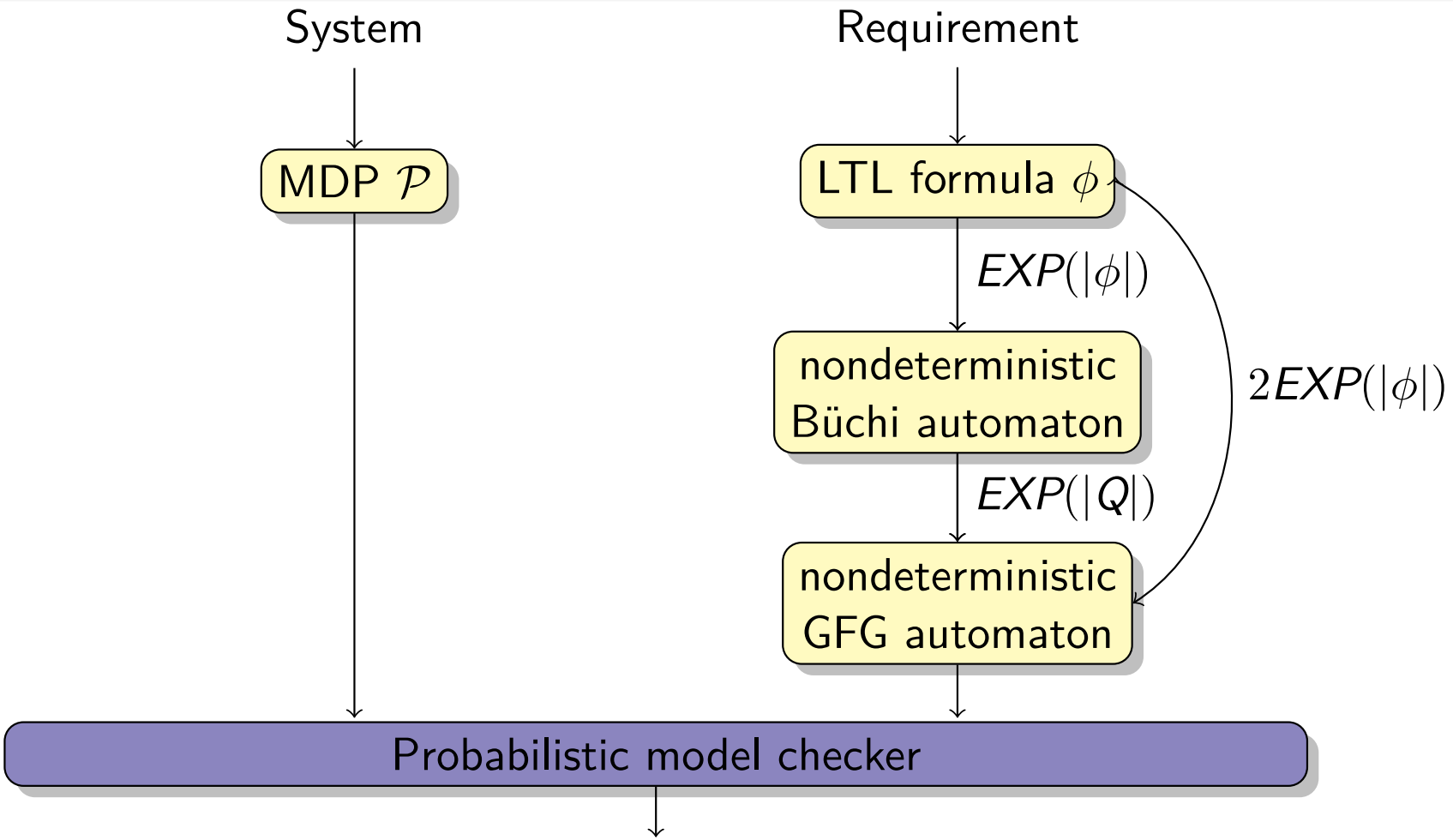
Probabilistic model checking



Probabilistic model checking

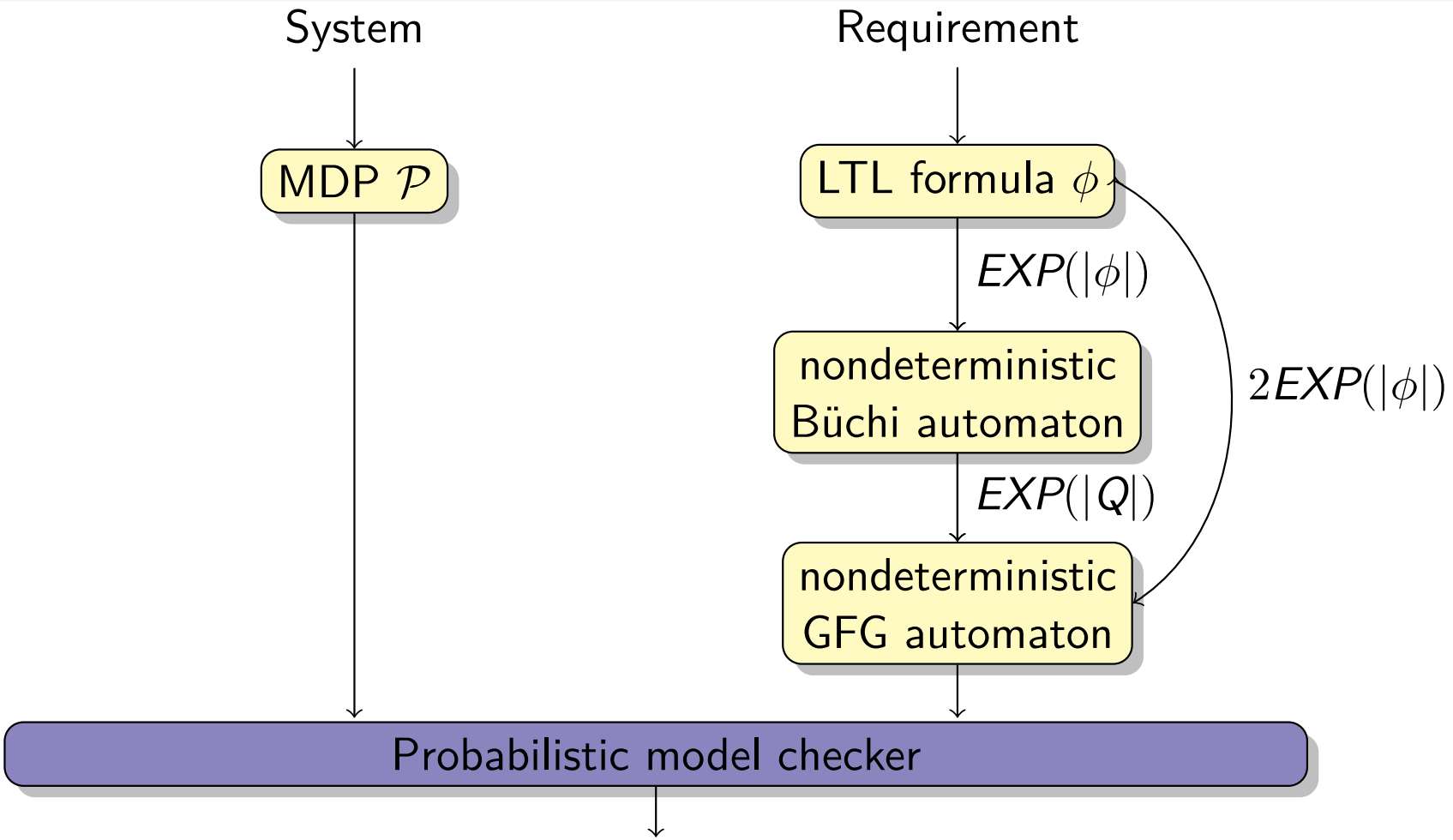


Probabilistic model checking



Lower bound NBA $\rightarrow$ GFG by Henzinger,Piterman, 2006

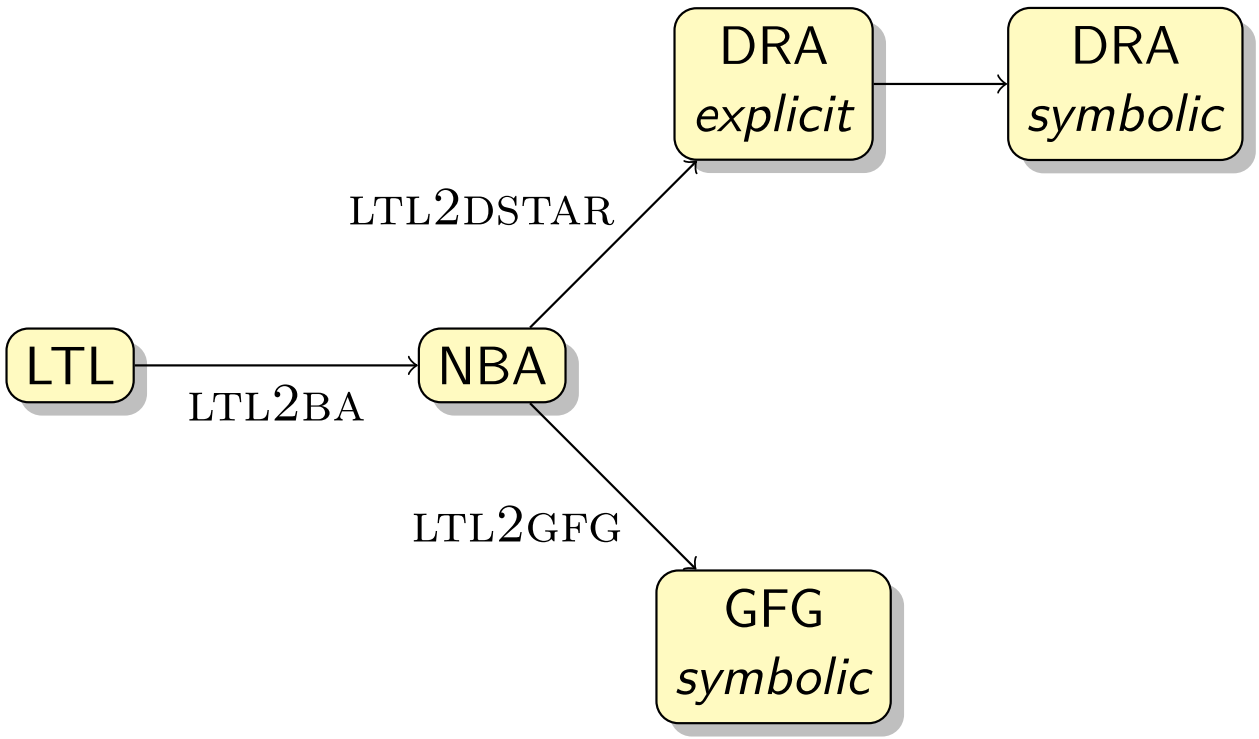
Probabilistic model checking



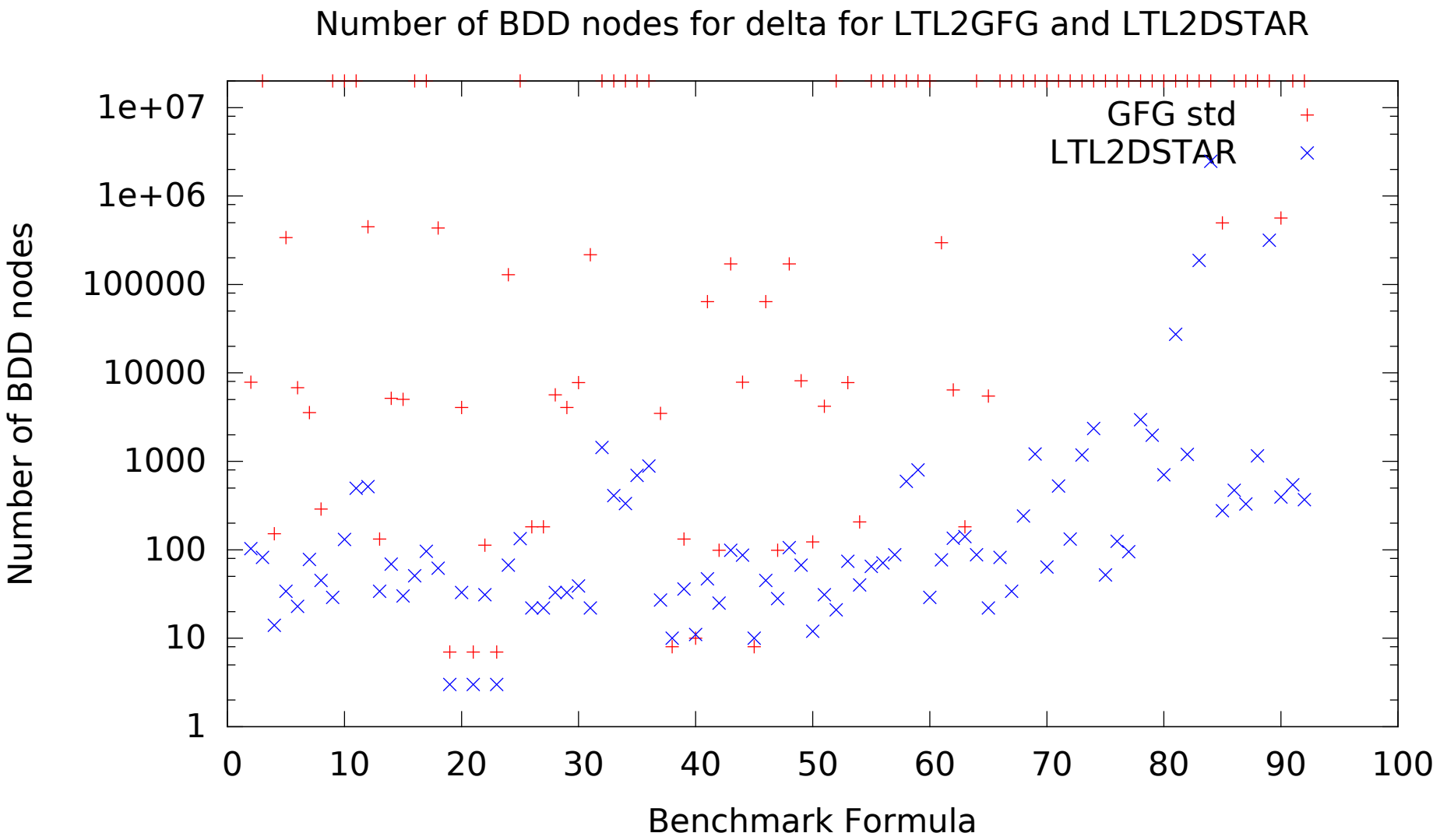
Lower bound NBA $\rightarrow$ GFG by Henzinger,Piterman, 2006

Lower bound LTL $\rightarrow$ GFG: straightforward adaption of Kupferman,Rosenberg 2011

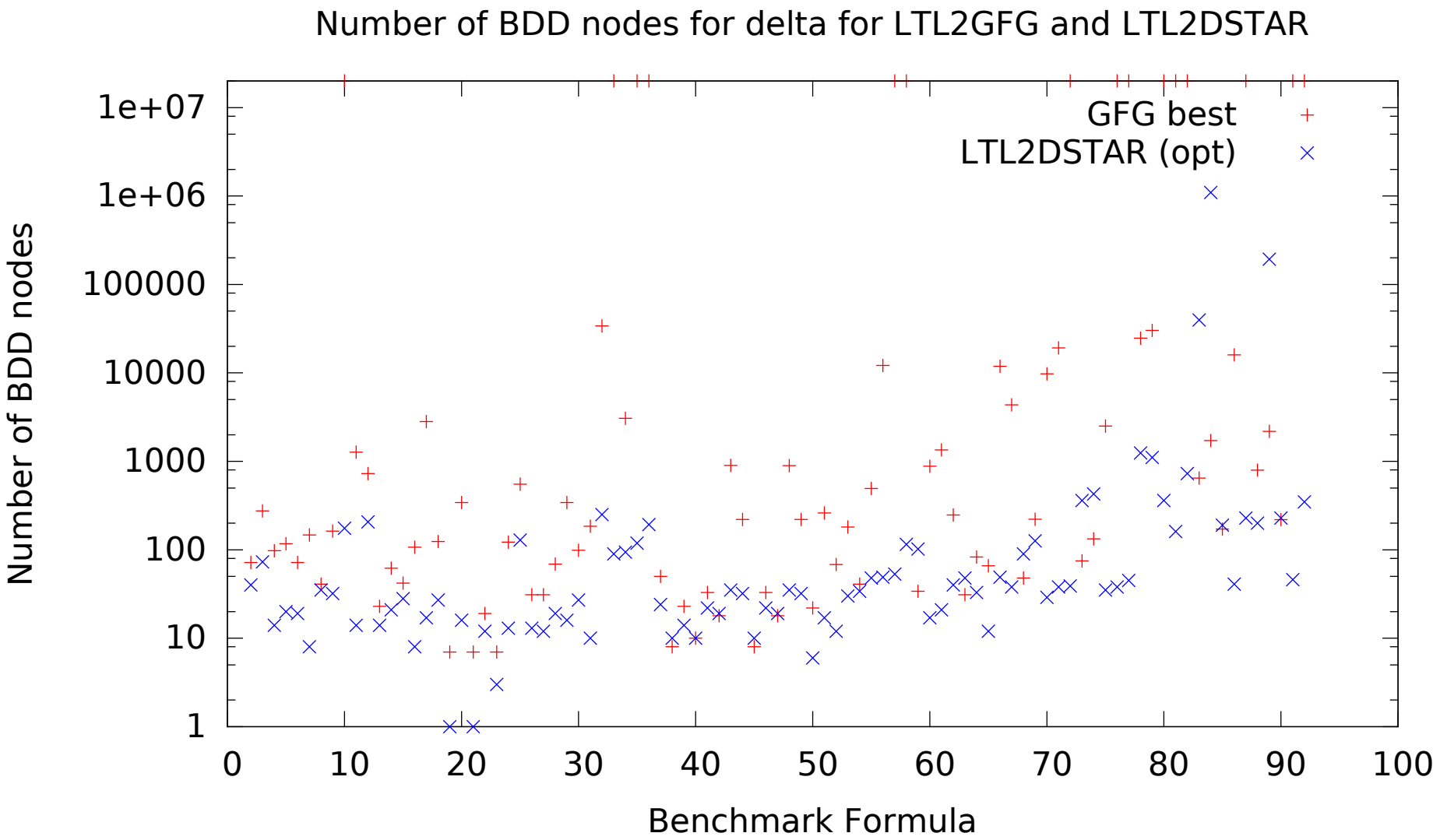
Implementations



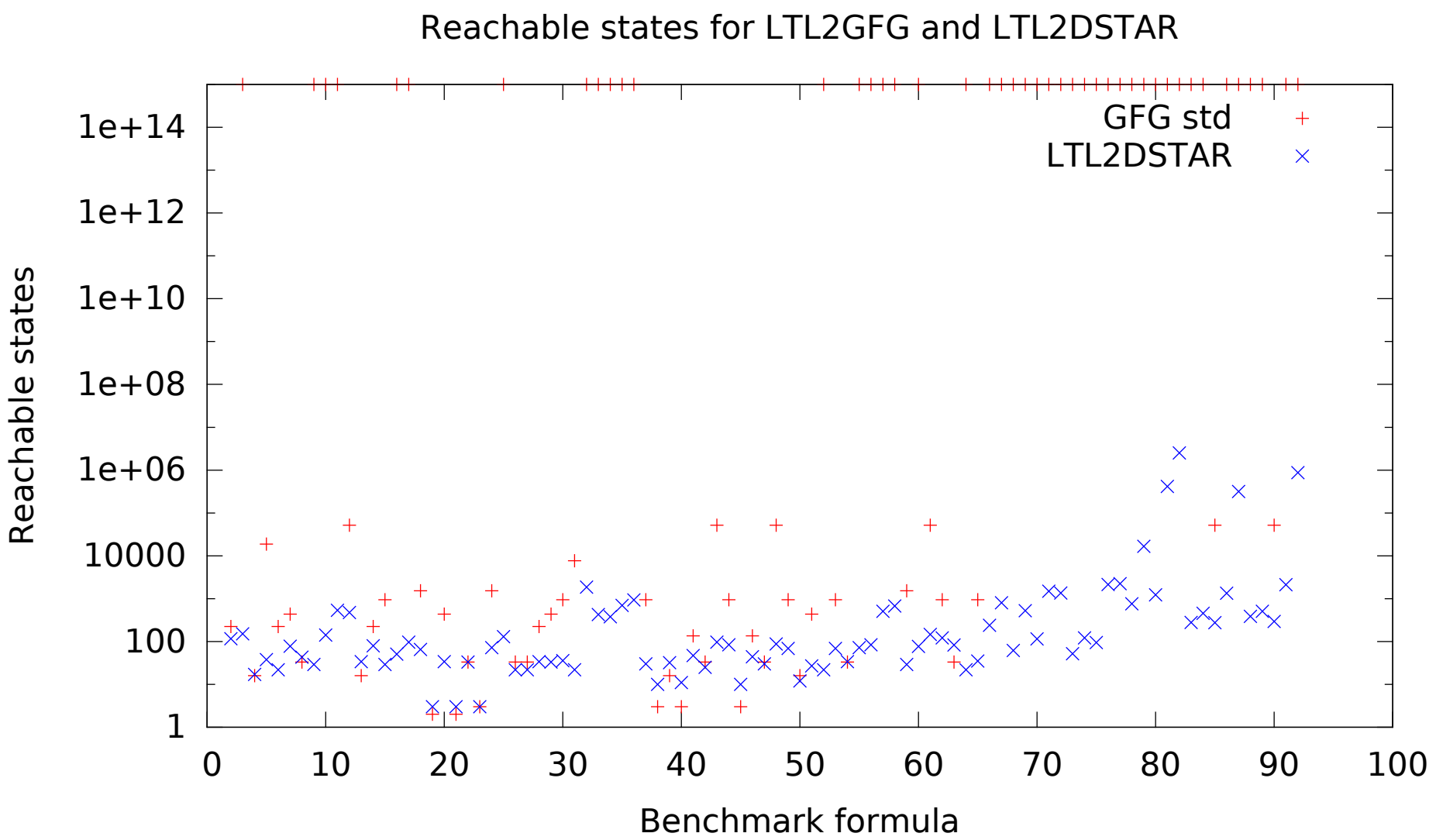
Size of symbolic BDD representation



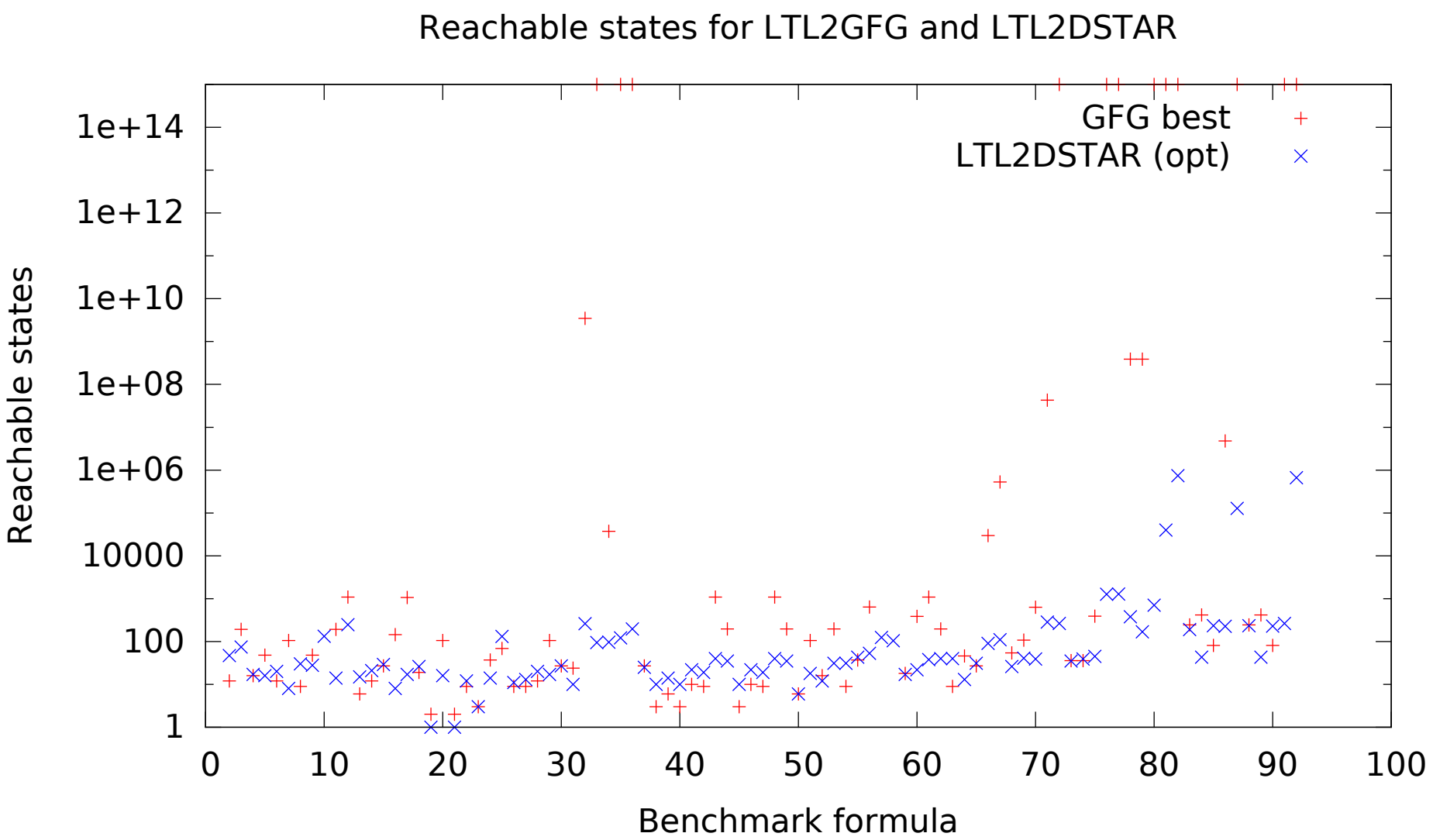
Size of symbolic BDD representation with heuristics



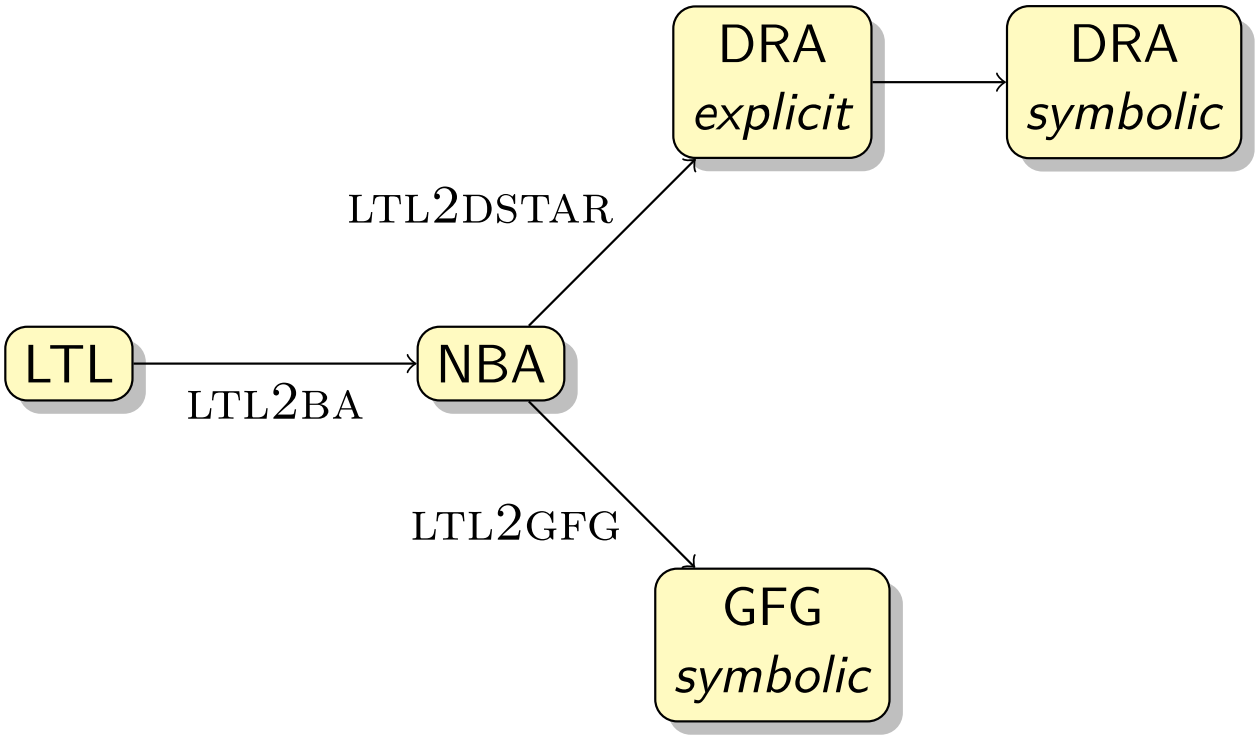
Reachable states



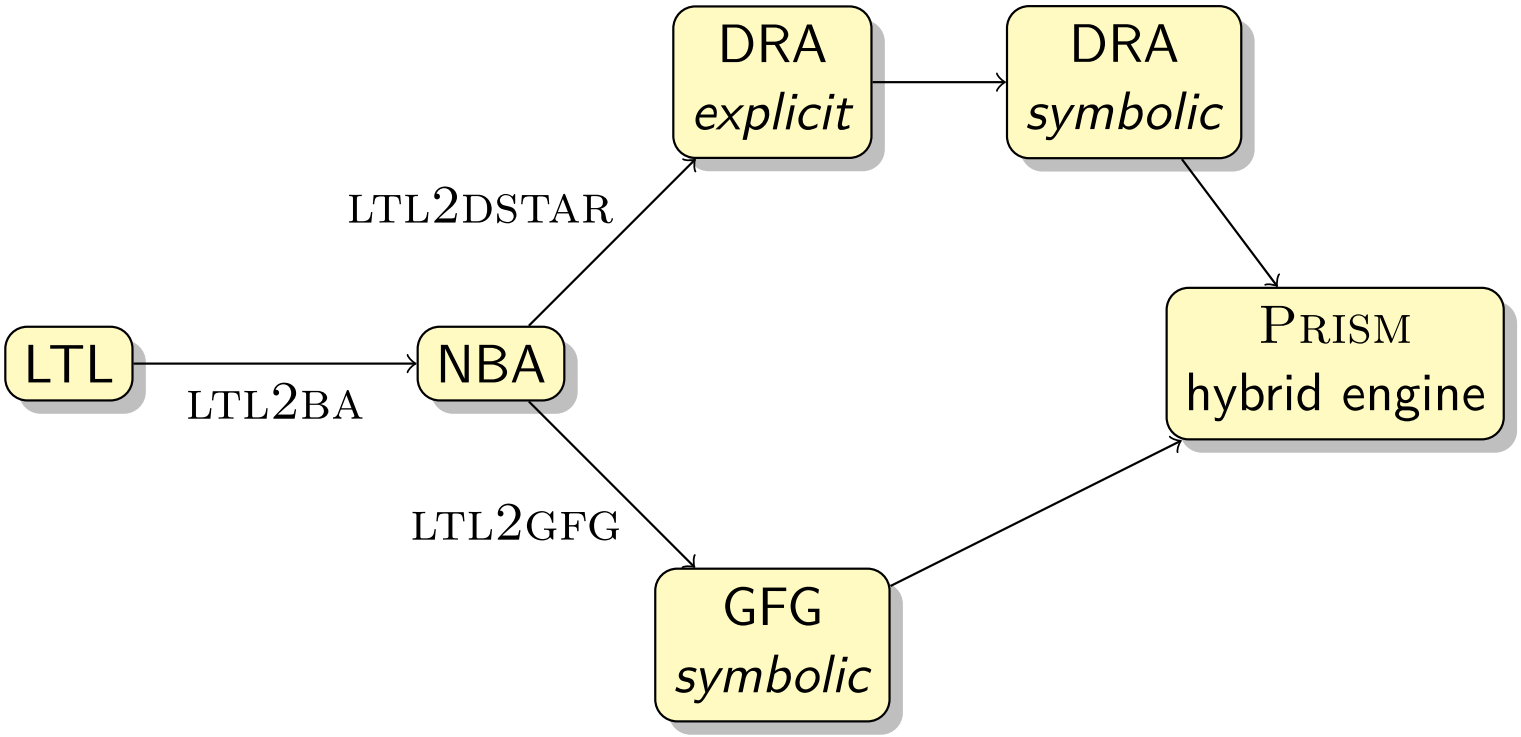
Reachable states with heuristics



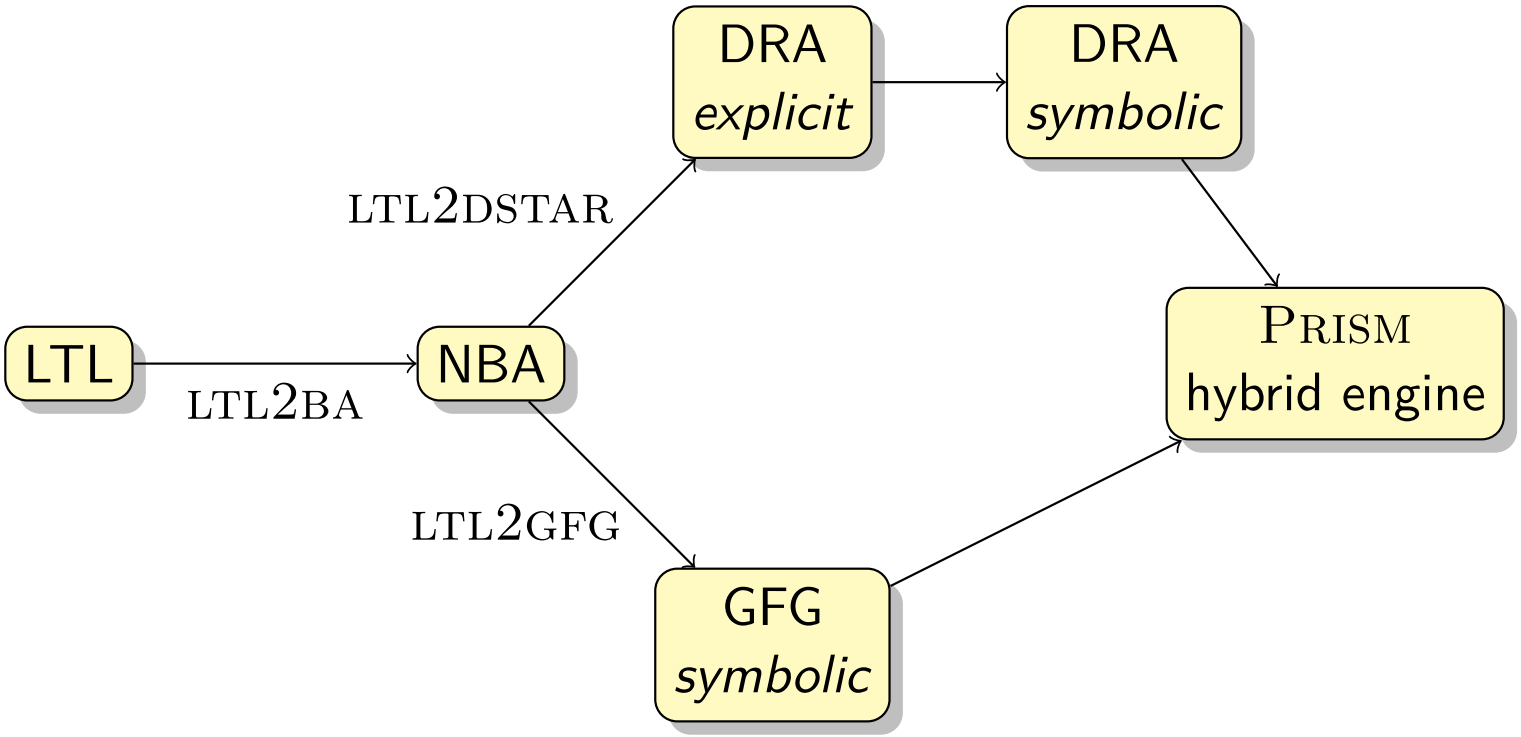
Implementation in probabilistic model checker PRISM



Implementation in probabilistic model checker PRISM



Implementation in probabilistic model checker PRISM



Case study: parts of WiFi carrier-sense protocol of IEEE 802.11 (Kwiatkowska, Norman and Sproston, 2002)

Benchmark: IEEE 802.11

$\Pr^{min}_{\mathcal{P}}(\varphi)$	time	
	LTL2GFG	PRISM std.
$\Diamond$Done	23.4 s	5.7 s
$\neg\Diamond\Box$error handling	38.4 s	0.7 s
$(\Box\Diamond\texttt{Backoff}_1) \rightarrow (\Box\Diamond\texttt{vuln}_1)$	63.4 s	2.9 s
$\Box(\texttt{colision} \rightarrow \Diamond\texttt{msg\_send}_1)$	43.6 s	3.1 s
$\Diamond\texttt{Done} \wedge \Box\#\texttt{collision} < 2$	97.6 s	42.9 s

Conclusion

- Established the use of *good-for-games automata* for model checking *Markov decision processes*

Conclusion

- Established the use of *good-for-games automata* for model checking *Markov decision processes*
- Evaluation of the transformation from NBA to GFG and the model checking process in PRISM

Thank you for your attention!

Questions or remarks?