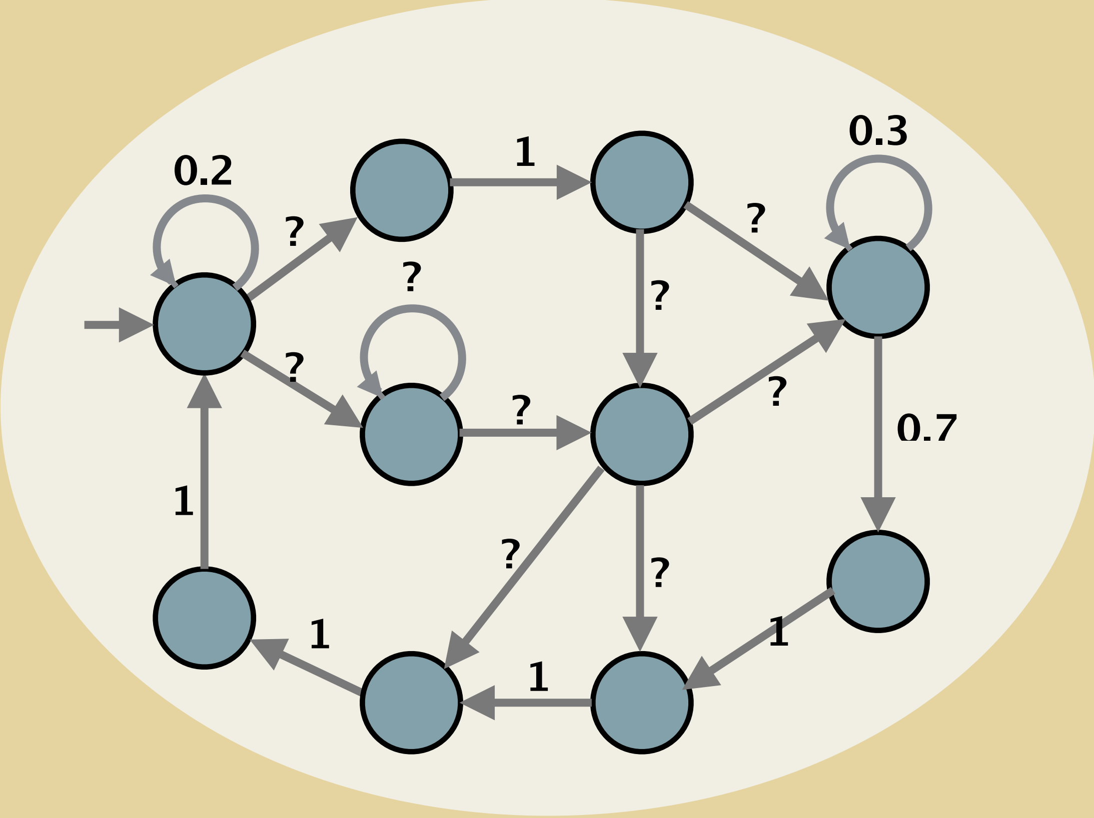
# Data-efficient Bayesian verification of parametric Markov Chains

E. Polgreen, V. Wijesuriya, S. Haesaert, A. Abate
Department of Computer Science, University of Oxford

QEST 2016

Model checking of systems with full models is well established ...

...but complete, accurate models are HARD to get.

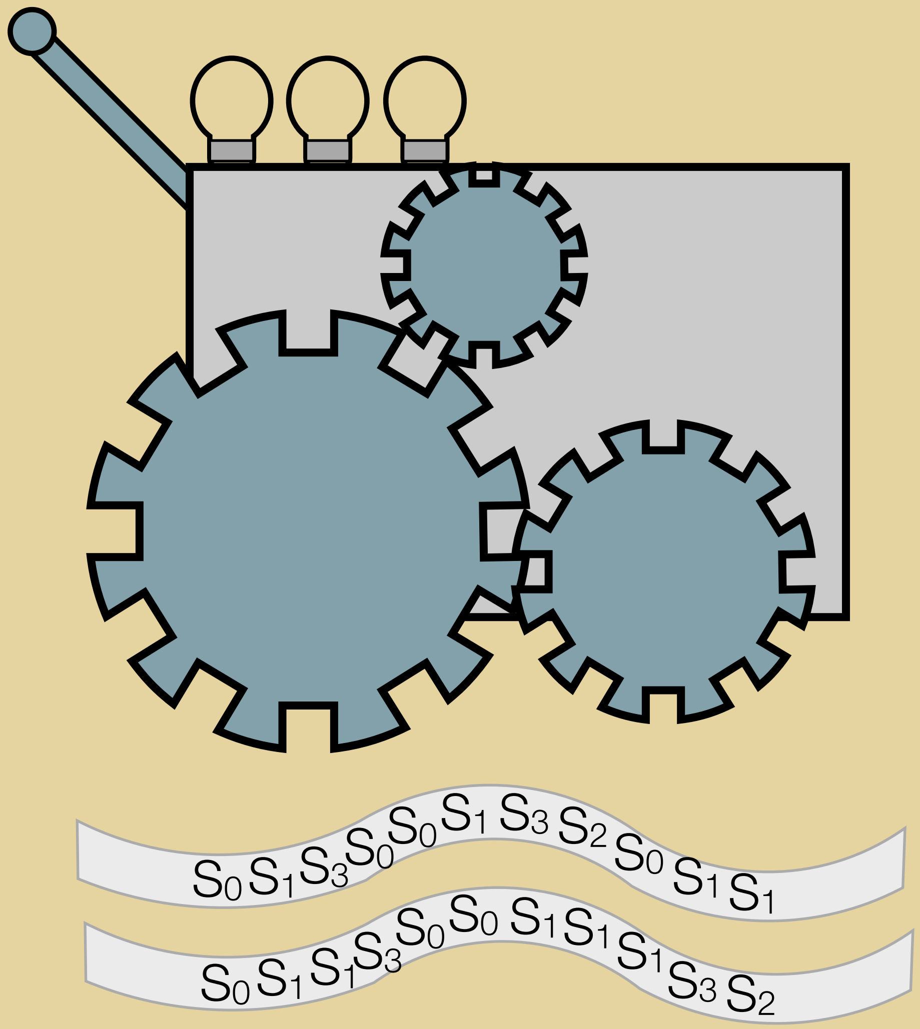


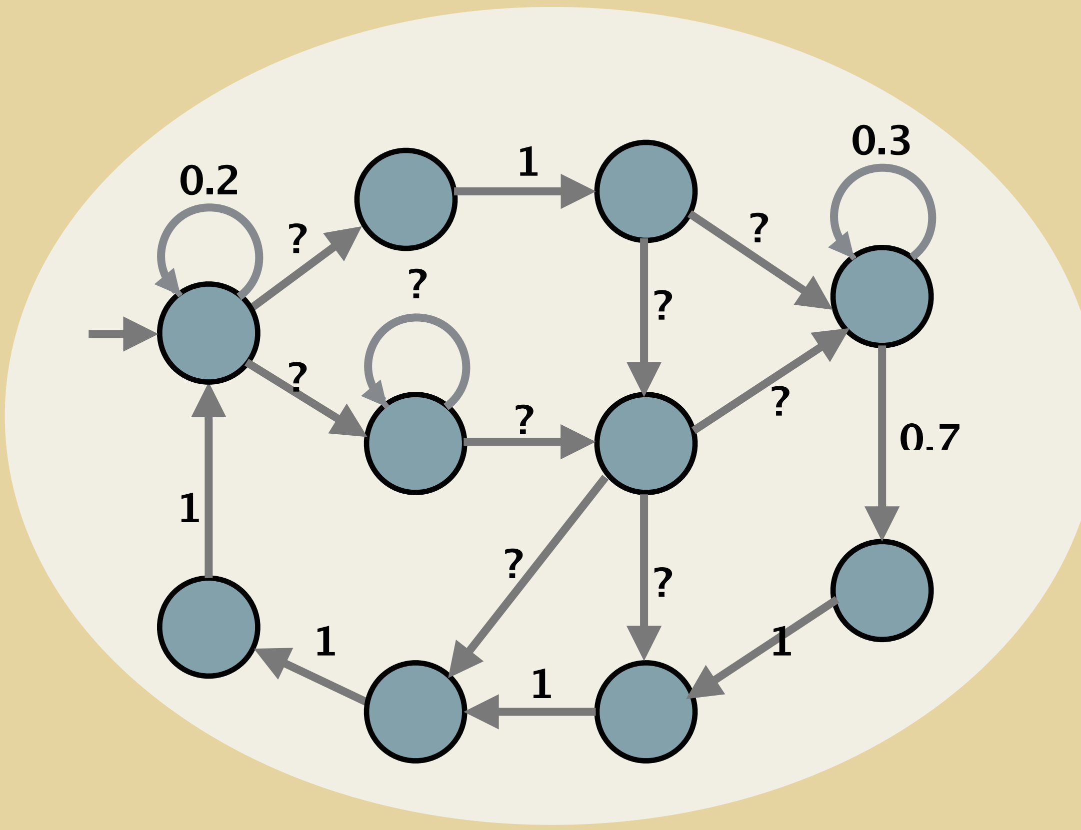

What can we do with a partial model?

Suppose we have a system,

we are given a partial model,

and a limited amount of system-generated data

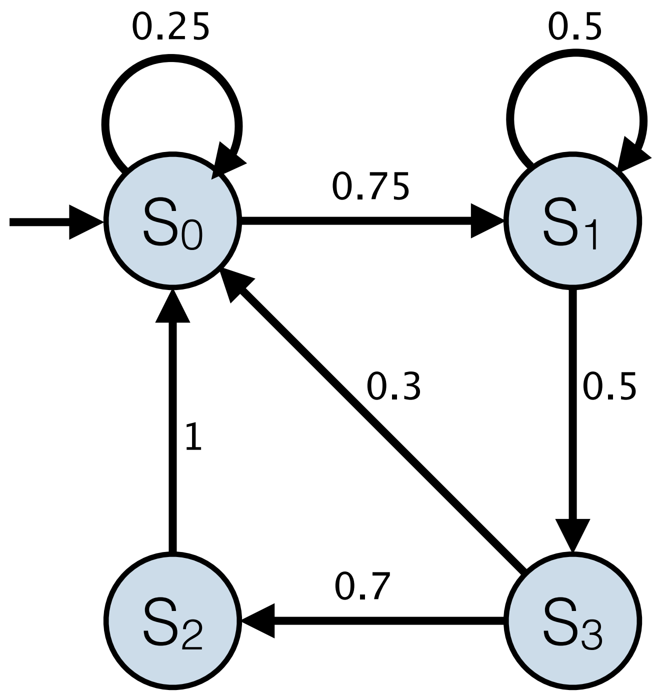



Can we check the system satisfies a PCTL property?

# Related work: “white-box” model checking

Explicit model checking: evaluate all possible paths in the **model**

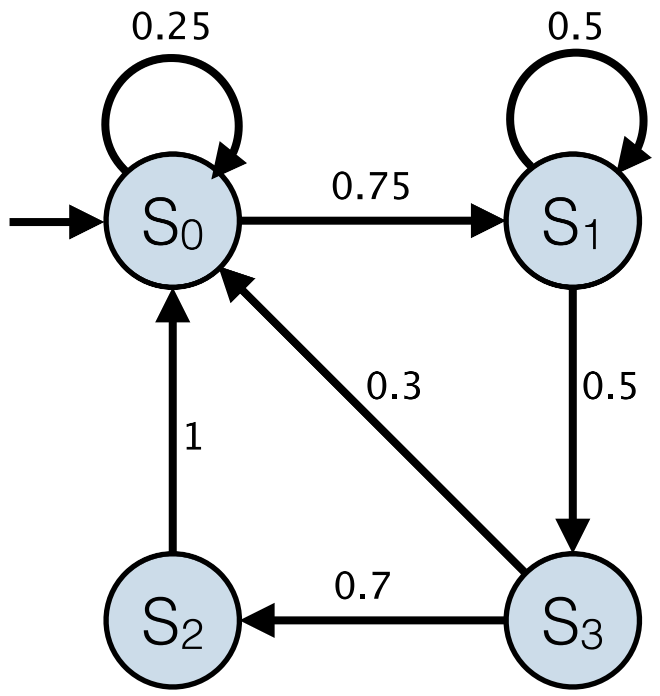



Proves that the **model** satisfies property

Relies on the model being correct and complete

# Related work: “white-box” model checking

Symbolic model checking: reason about all possible paths in the **model**

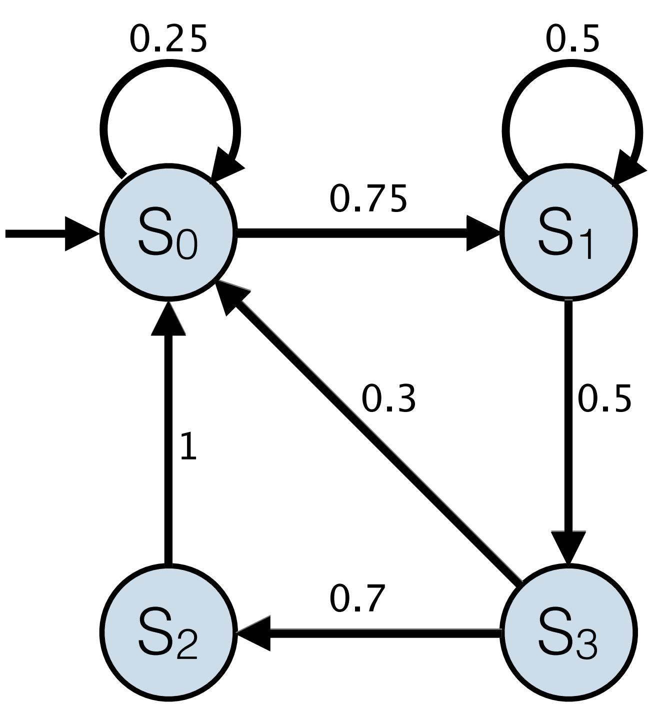



Proves that the **model** satisfies property

Relies on the model being correct and complete

# Related work: “white-box” model checking

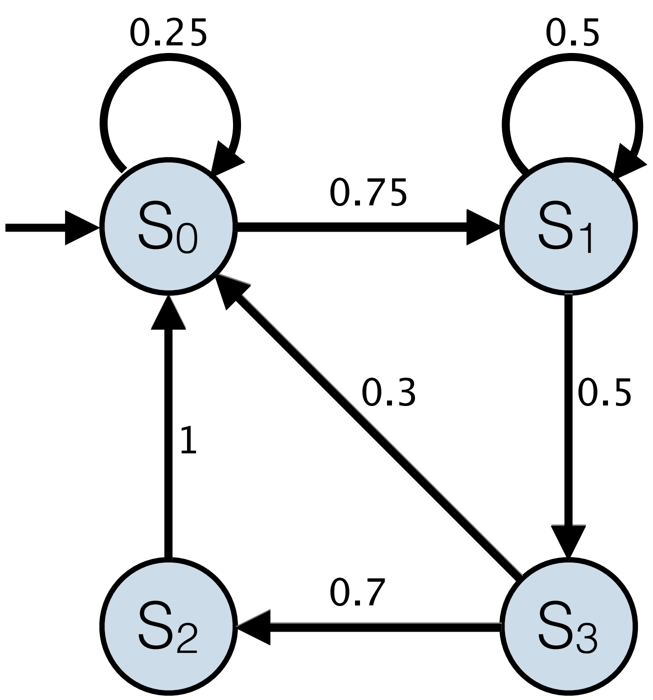
Statistical Model Checking (SMC): generate sample data from the **model**




| |
|---|
| $S_0\ S_1\ S_3\ S_2\ S_0\ S_0\ S_1 S_1$ |
| $S_0\ S_0\ S_1\ S_1\ S_3\ S_0\ S_1 S_1$ |
| $S_0\ S_1\ S_3\ S_0\ S_0\ S_1 S_1 S_1$ |
| $S_0\ S_1\ S_3\ S_2\ S_0\ S_0\ S_1 S_1$ |

# Related work: "white-box" model checking

Statistical Model Checking (SMC): generate sample data from the **model**

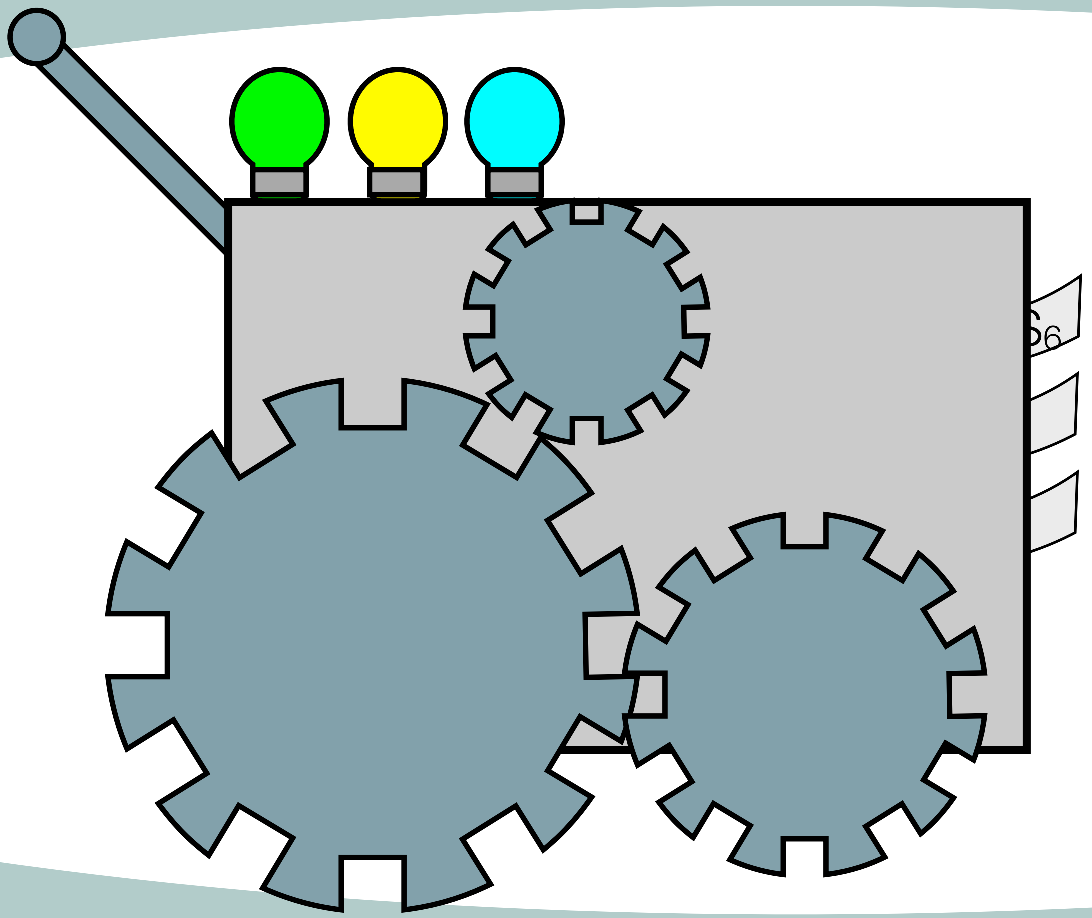



Gives probability that the **model** satisfies property, for BIG models

Relies on the model being correct and complete

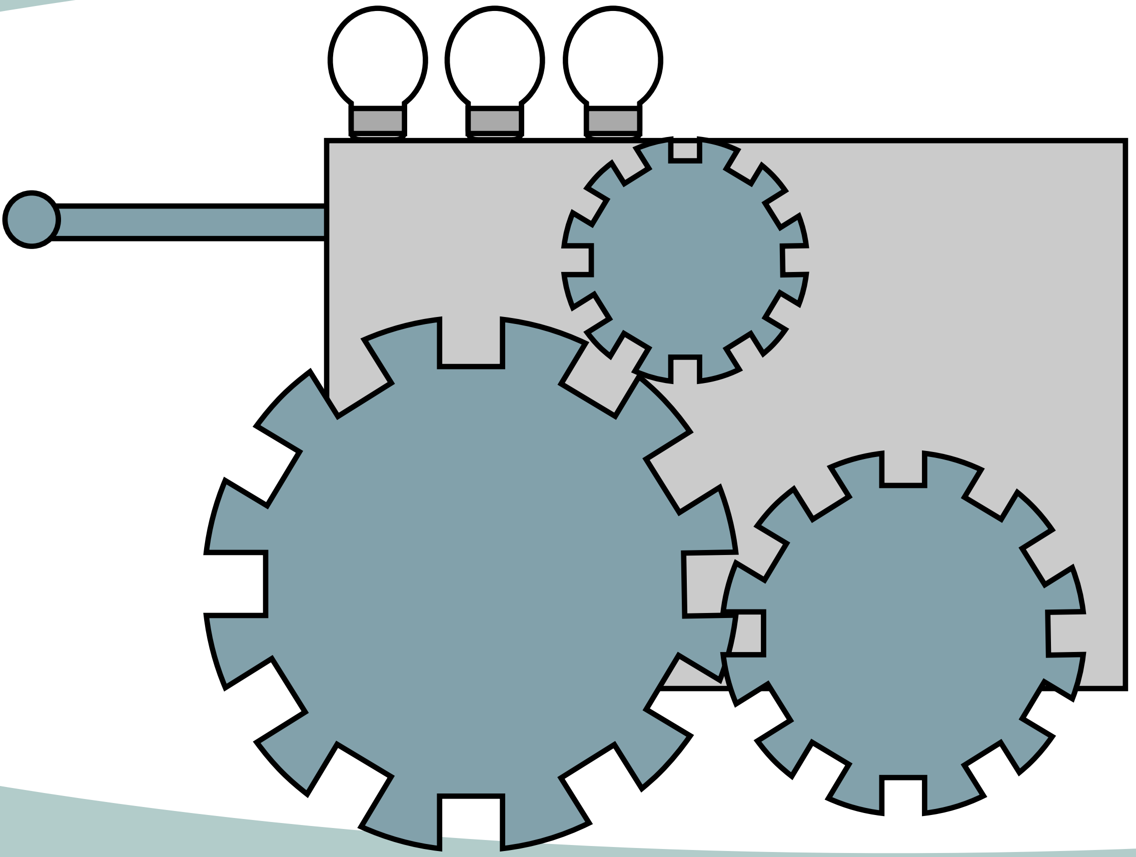
# Related work: "black-box" model checking

Statistical Model Checking (SMC): collect sample data from the **system**



# Related work: “black-box” model checking

Statistical Model Checking (SMC): collect sample data from the **system**



Gives probability that **system** satisfies property.

Needs a lot of data.

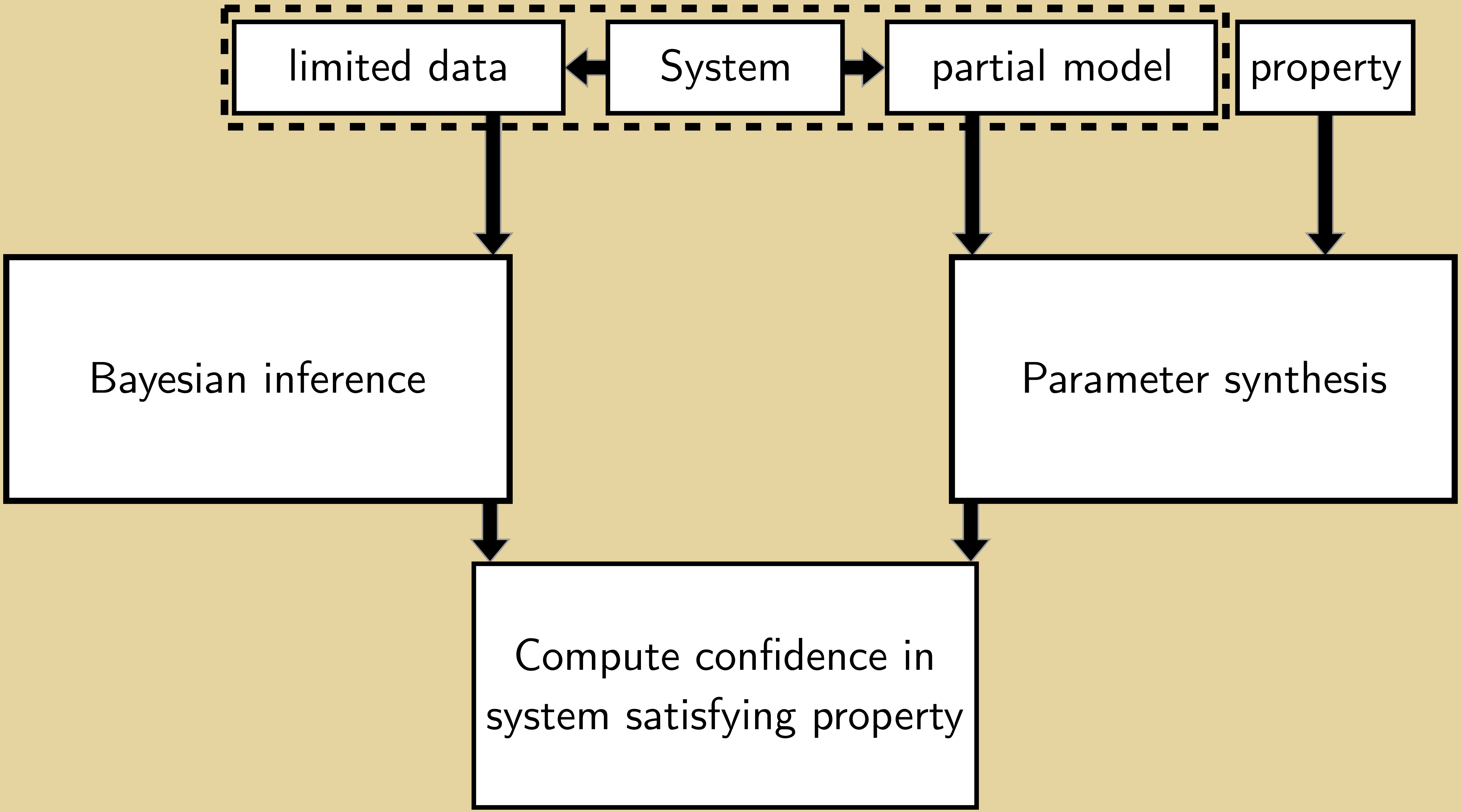
# Our approach:

Consider a scenario with limited data, so we can't use "black-box" SMC,

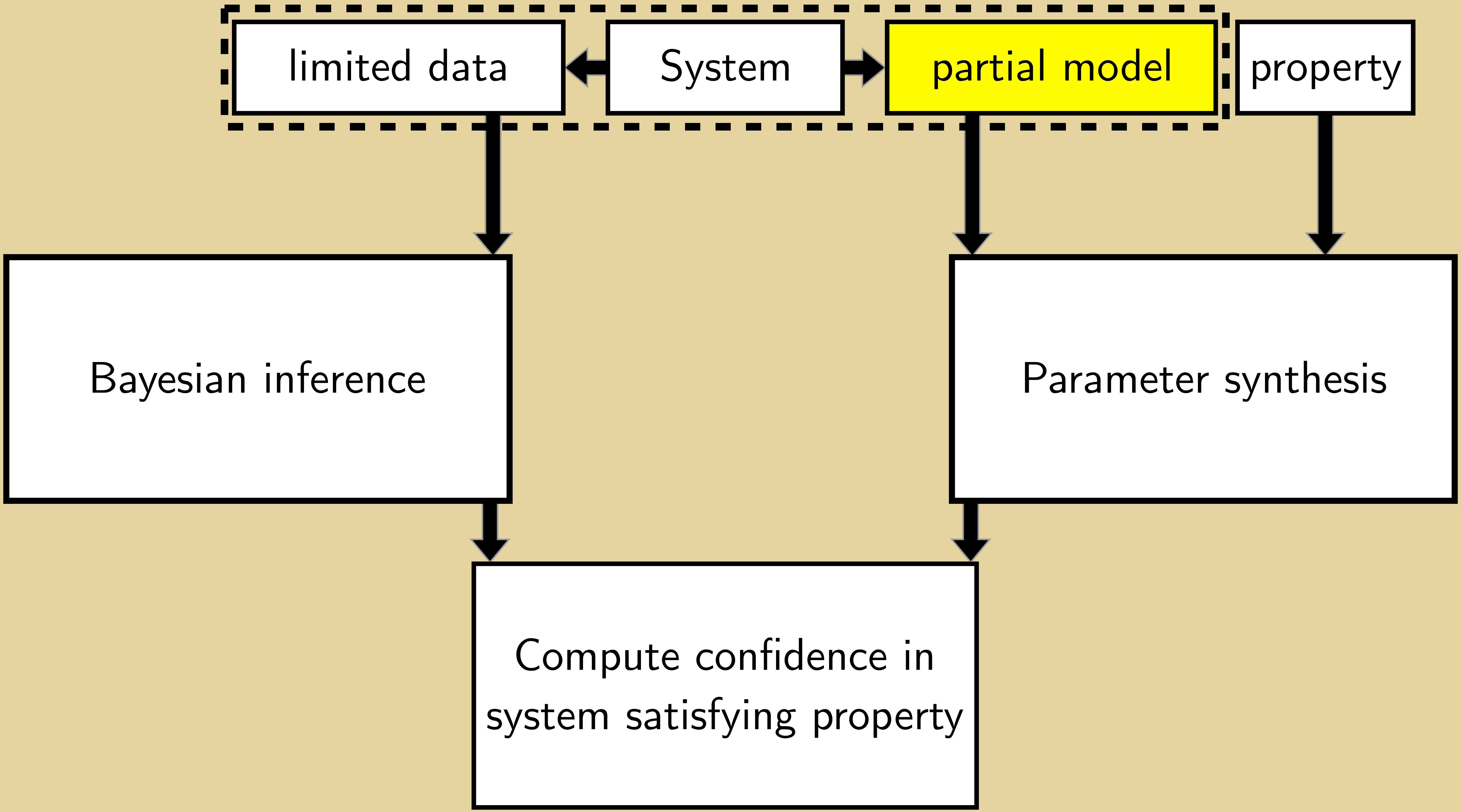
and only a partial model, so we can't use "white-box" model checking.

We combine parameter synthesis + data-based learning to compute the confidence the system satisfies a property.

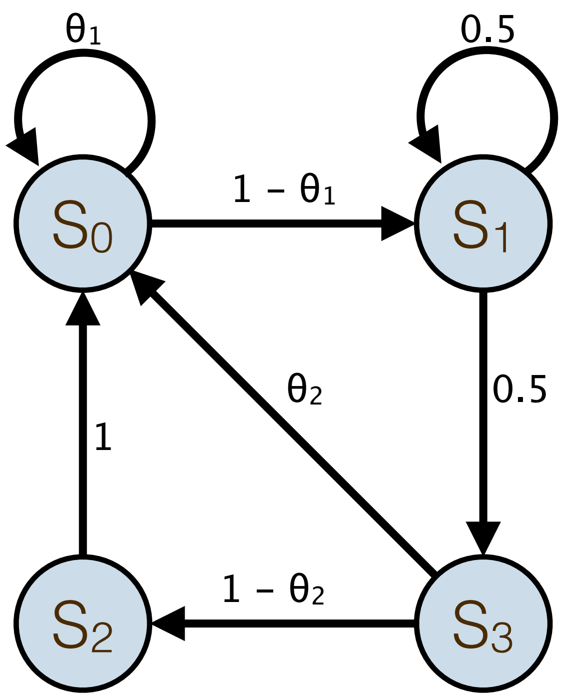
# Overview


limited data
System
partial model
property
Bayesian inference
Parameter synthesis
Compute confidence in system satisfying property

# Overview


limited data
System
partial model
property
Bayesian inference
Parameter synthesis
Compute confidence in system satisfying property

# Parametric Markov chains




basic pMC

*initial state*

*atomic propositions*

S = *finite set of states*

*labelling function*

$$\mathbf{M}_{\Theta} = (S, \mathbb{T}_{\theta}, \iota_{init}, \mathrm{AP}, L, \Theta)$$

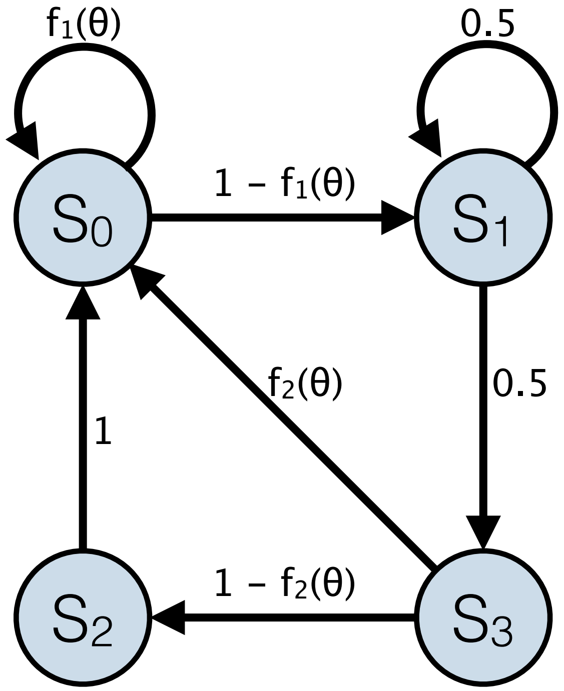
$\mathbb{T}_{\theta}$ = *parameterised transition function*

$\theta$ = *parameter vector*

$\Theta$ = *set of all possible valuations of* $\theta$

Basic pMC - transition probabilities are known constants or single parameters

# Parametric Markov chains




linear pMC

$$\mathbf{M}_{\Theta} = (S, \mathbb{T}_{\theta}, \iota_{init}, \mathrm{AP}, L, \Theta)$$

S = *finite set of states*

*initial state*

*atomic propositions*

*labelling function*

$\mathbb{T}_{\theta}$ = *parameterised transition function*

$\theta$ = *parameter vector*

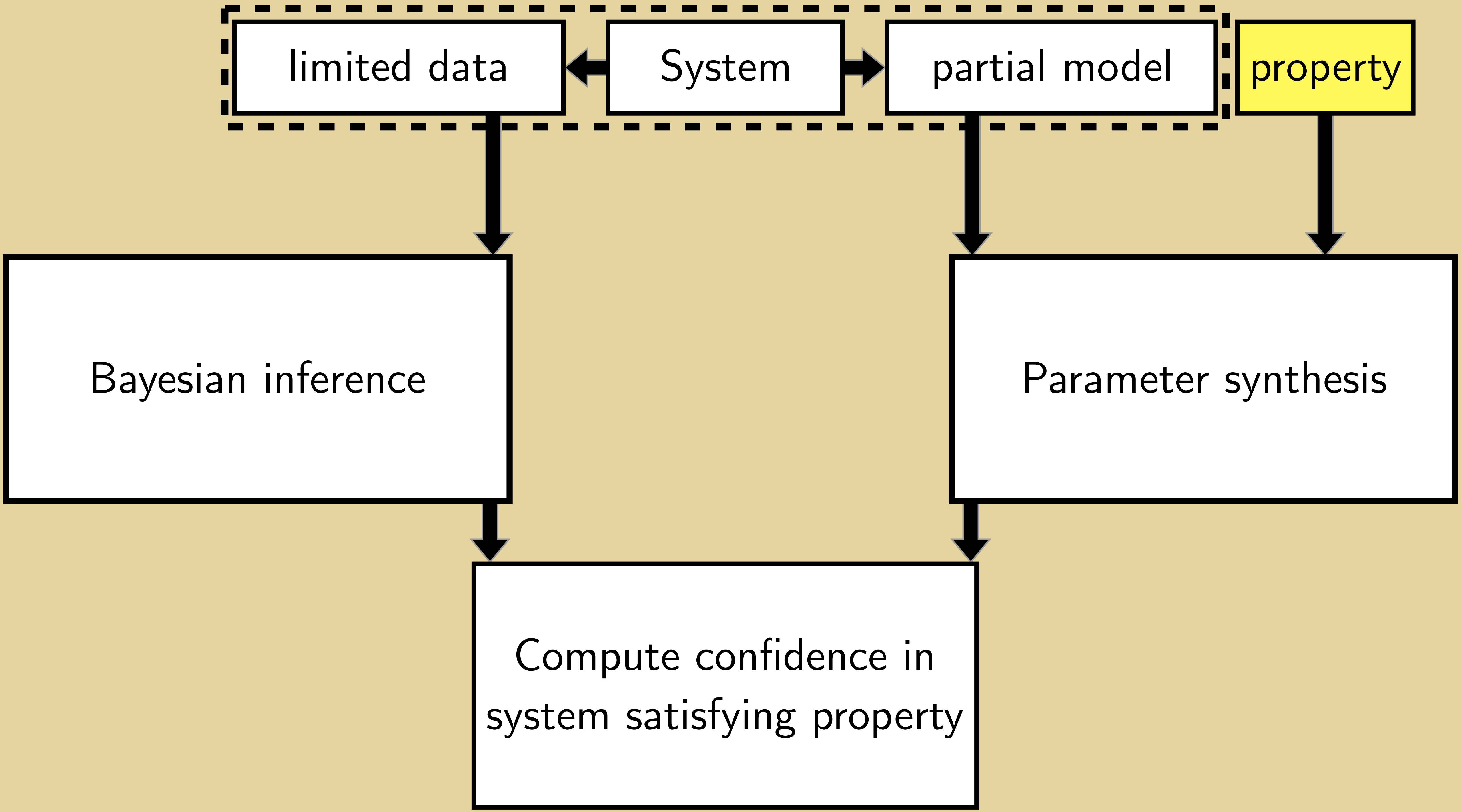
$\Theta$ = *set of all possible valuations of* $\theta$

Linear pMC - transition probabilities are linear functions of parameters

$$f_l(\theta) = k_0 + k_1\theta_1 + k_2\theta_2 + \ldots + k_n\theta_n$$

$$k_i \in [0,1] \qquad \sum k_i \leq 1$$

# Overview

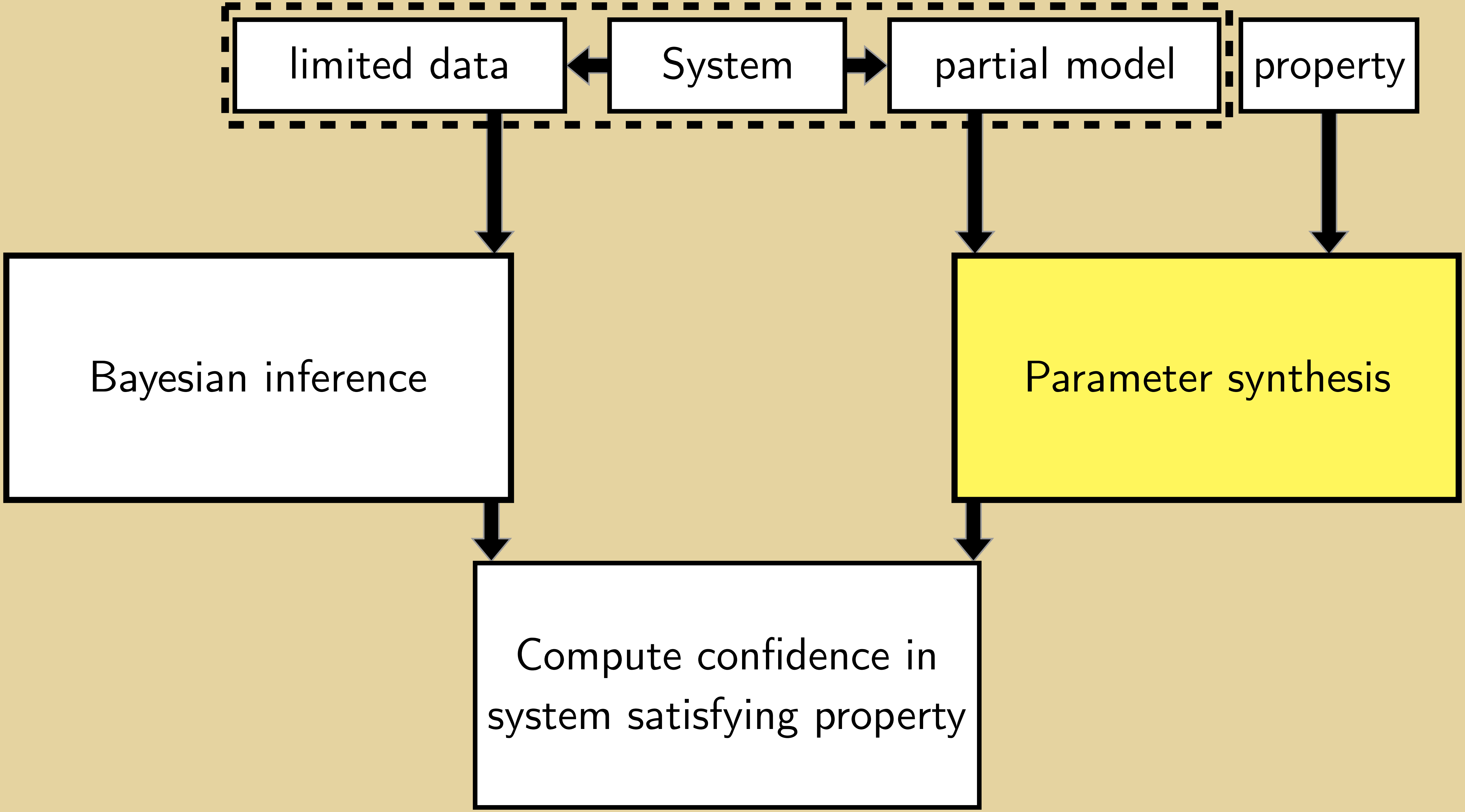

limited data
System
partial model
property
Bayesian inference
Parameter synthesis
Compute confidence in
system satisfying property

# PCTL properties

We are able to consider any property that is compatible with the PRISM parameter synthesis tool. We focus on non-nested PCTL

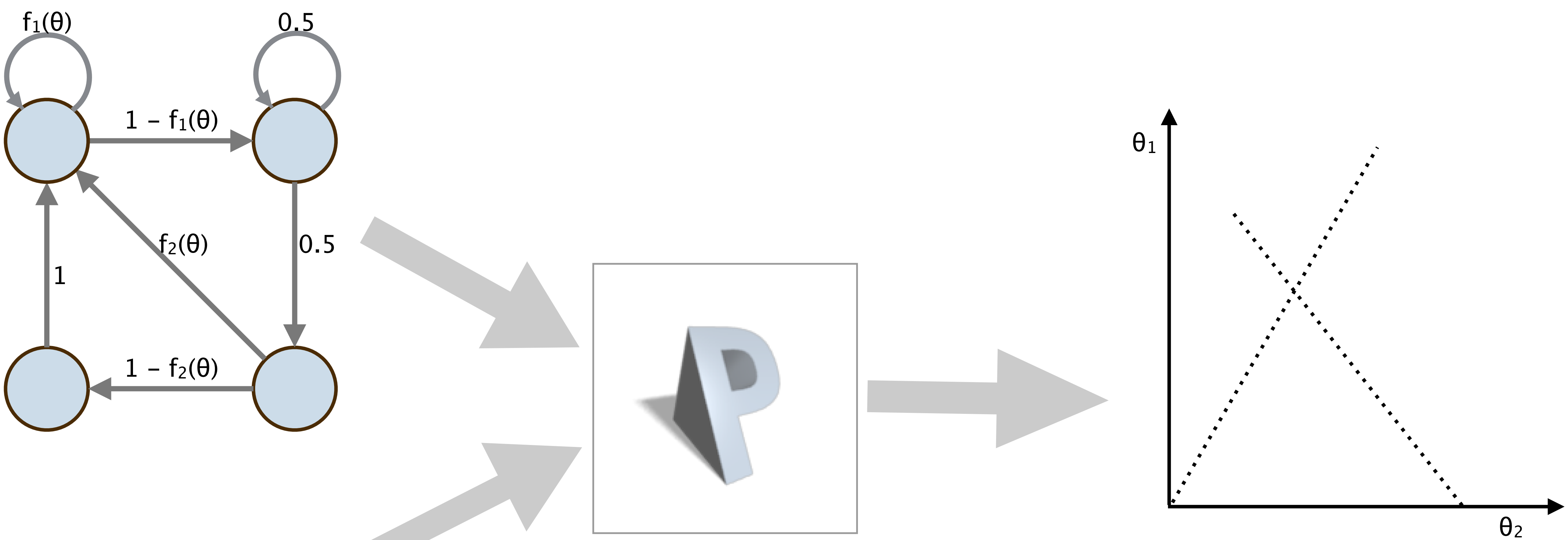
$$\phi = P_{>0.5}[G \neg s_2]$$

# Overview


limited data
System
partial model
property
Bayesian inference
Parameter synthesis
Compute confidence in system satisfying property

# Parameter Synthesis

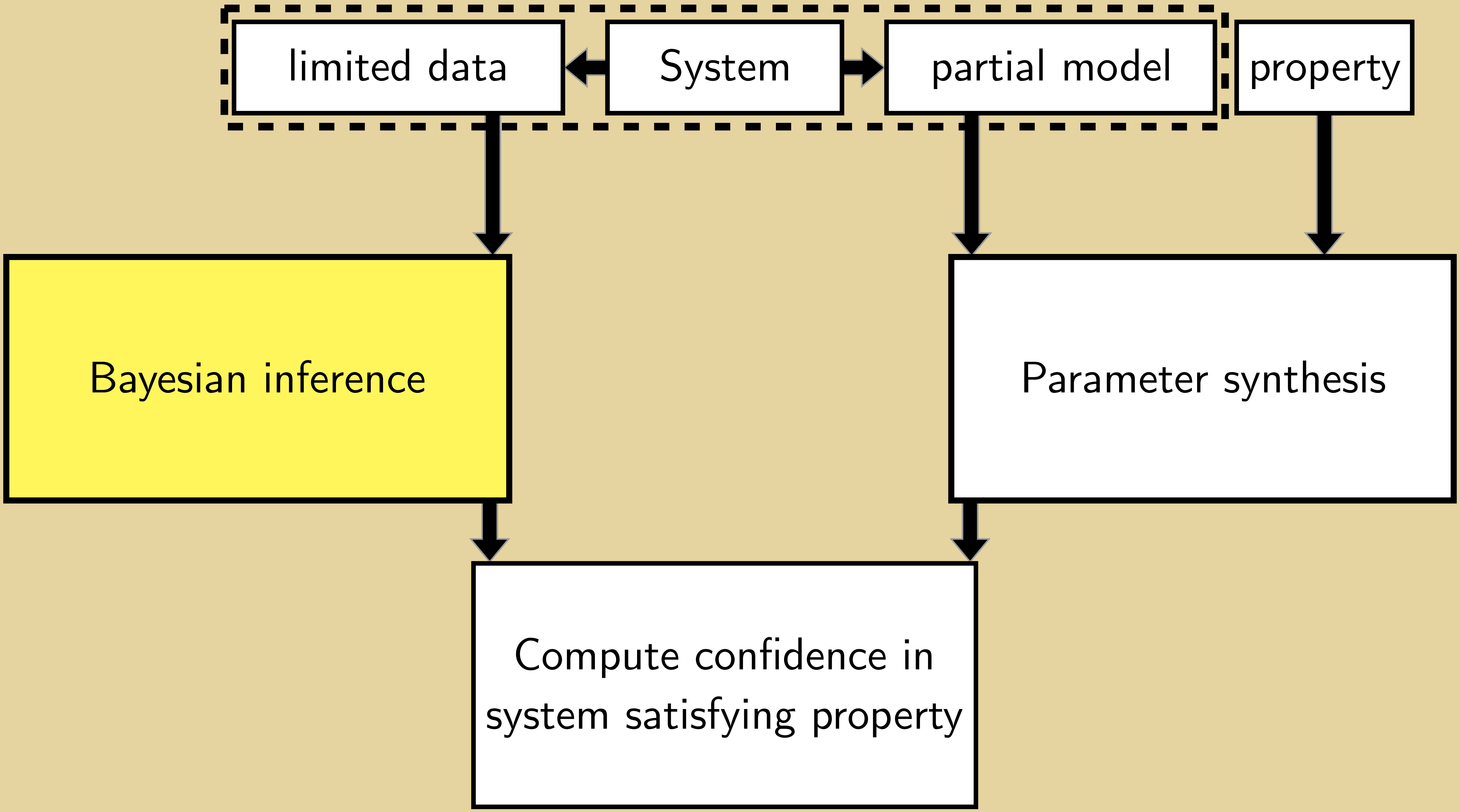
We use PRISM to synthesise the feasible set of parameters, for which the model satisfies the property



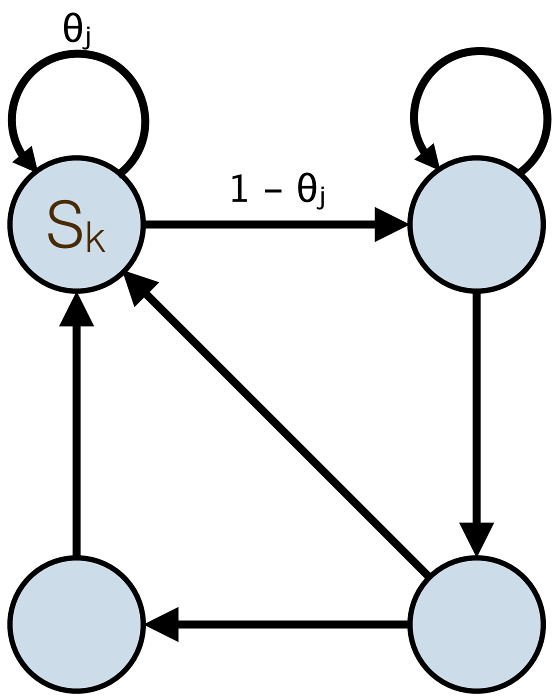

$\phi = P_{>0.5}[G \neg s_2]$

$$\Theta_\phi = \{\theta \in \Theta : \mathbf{M}(\theta) \models \phi\}$$

# Overview


limited data
System
partial model
property
Bayesian inference
Parameter synthesis
Compute confidence in system satisfying property

# Bayesian Inference: basic pMC

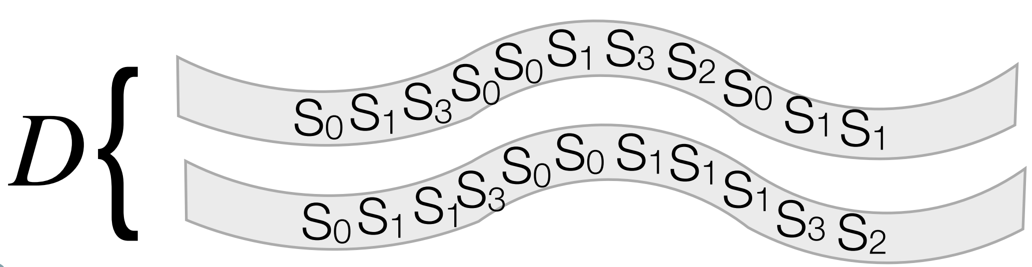


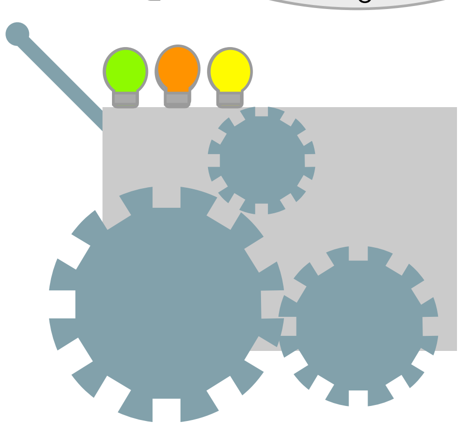

*observed data* *prior*

$$p(\theta_j \mid D) = \frac{\mathbb{P}(D \mid \theta_j)p(\theta_j)}{\mathbb{P}(D)}$$

$$= \frac{p(\theta_j)\boxed{\prod_{s' \in S} \mathbb{T}_\theta(s_k, s')^{D_{s_k}^{s'}}}}{\mathbb{P}(D_{s_k})}$$

binomial distribution

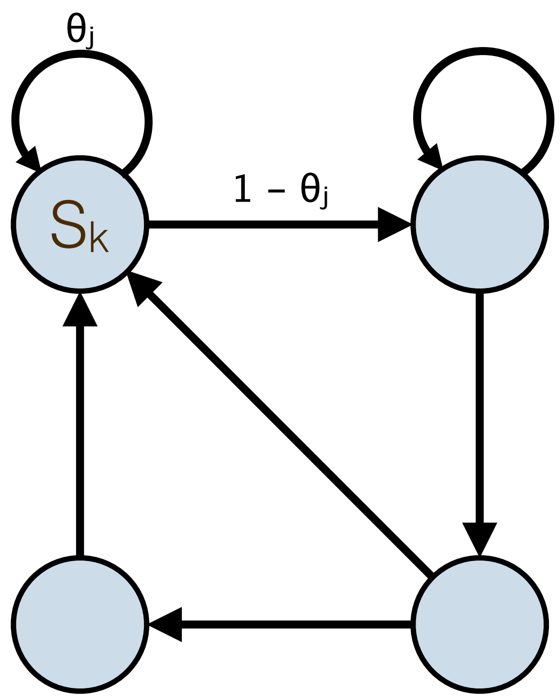


Conjugate prior = Dirichlet

$$\mathrm{Dir}(\theta_j \mid \alpha) = \tfrac{1}{B(\alpha)}\theta_j^{\alpha_1 - 1}(1-\theta_j)^{\alpha_2 - 1}$$

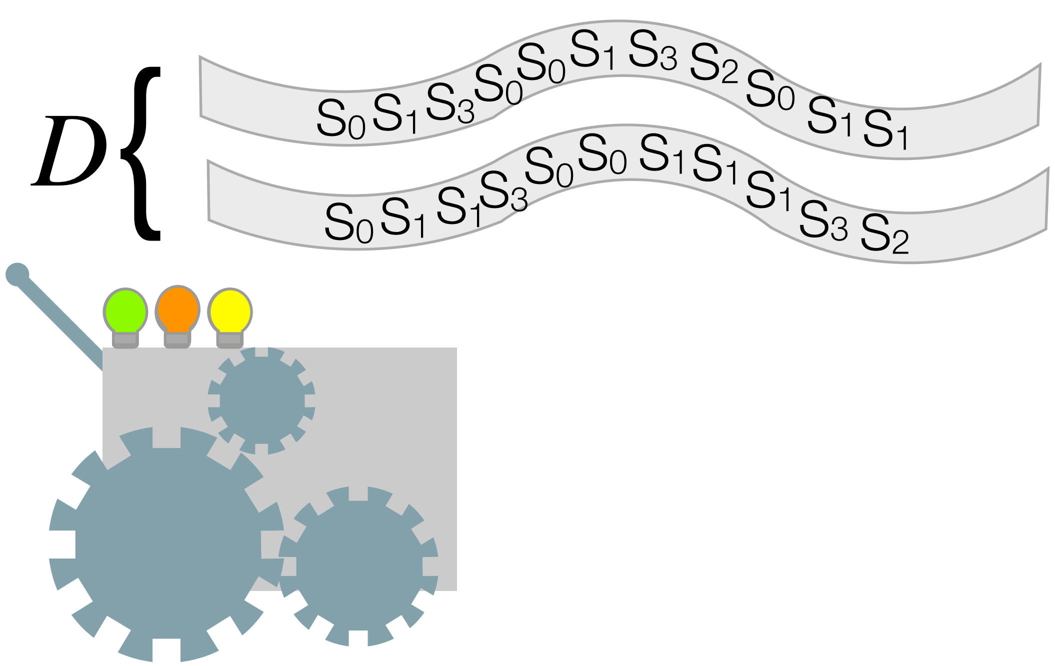
# Bayesian Inference: basic pMC




*observed data* *prior*

$$p(\theta_j \mid D) = \frac{\mathbb{P}(D \mid \theta_j)p(\theta_j)}{\mathbb{P}(D)}$$

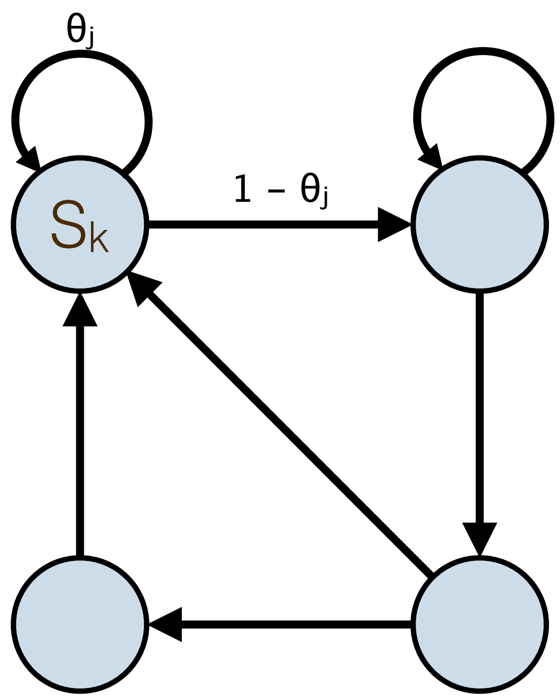
$$p(\theta_j \mid D) \propto \theta_j^{\alpha_1 - 1}(1-\theta_j)^{\alpha_2 - 1}\theta_j^{D_{s_k}^{s_1}}(1-\theta_j)^{D_{s_k}^{s_2}}$$



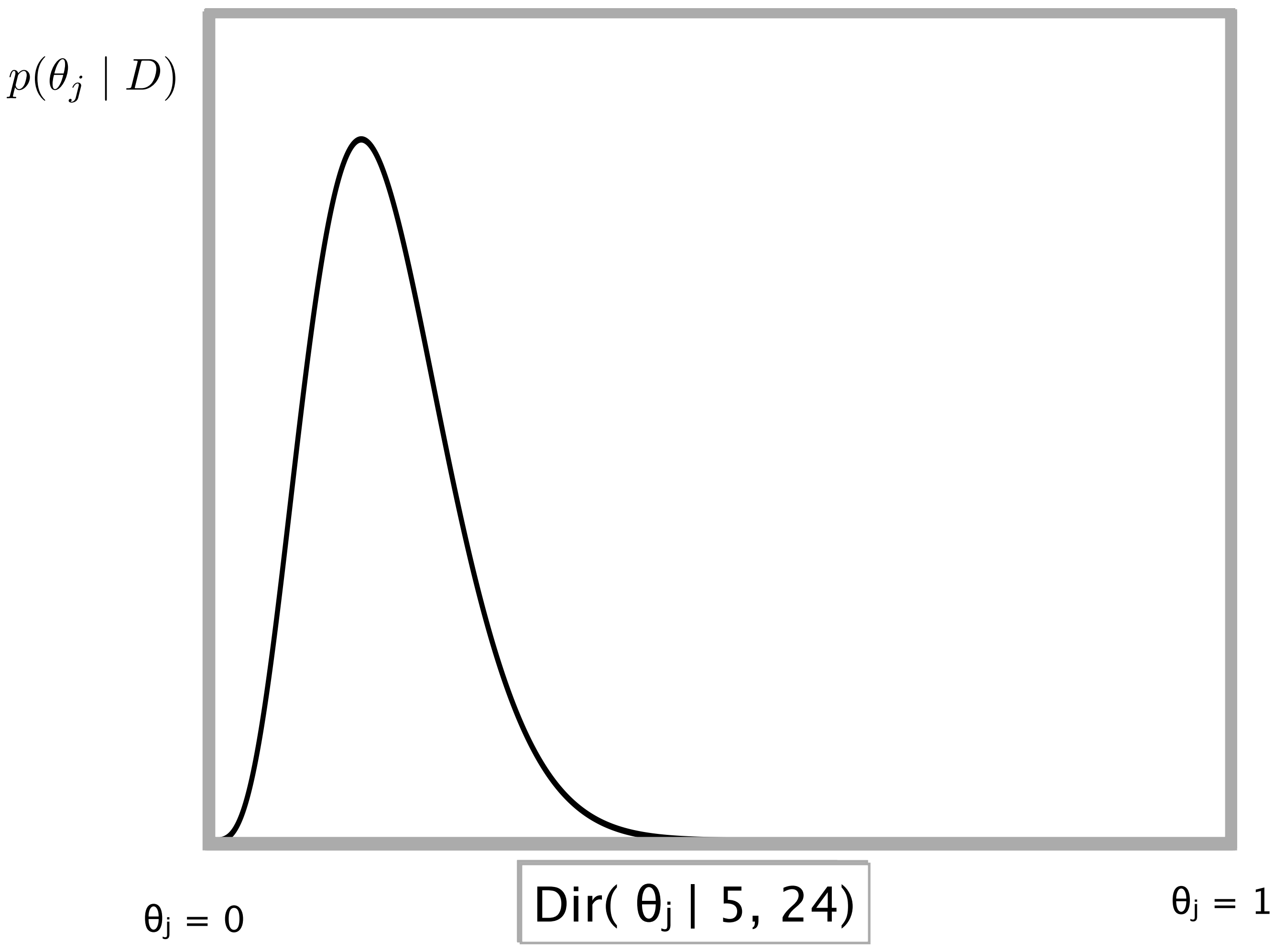

Hence, updating posterior = adding transition count to Dirichlet hyper-parameters

$$p(\theta \mid D) = \prod_{s_i} \mathrm{Dir}(\theta_{s_i} \mid D_{s_i} + \alpha)$$

# Bayesian Inference: basic pMC




| | |
|---|---|
| COUNT [1 – θj] | 23 |
| COUNT [θj] | 4 |

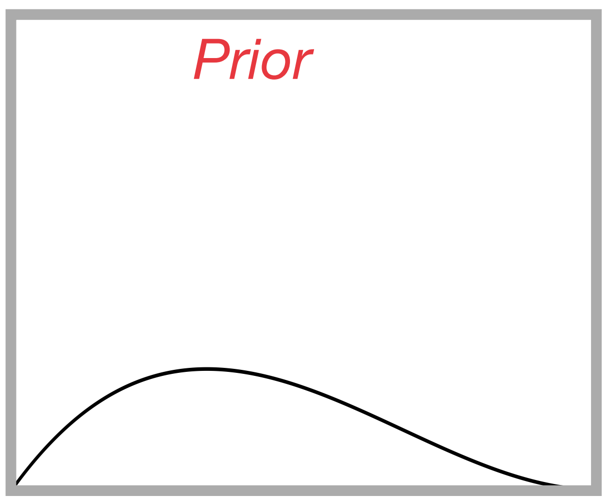


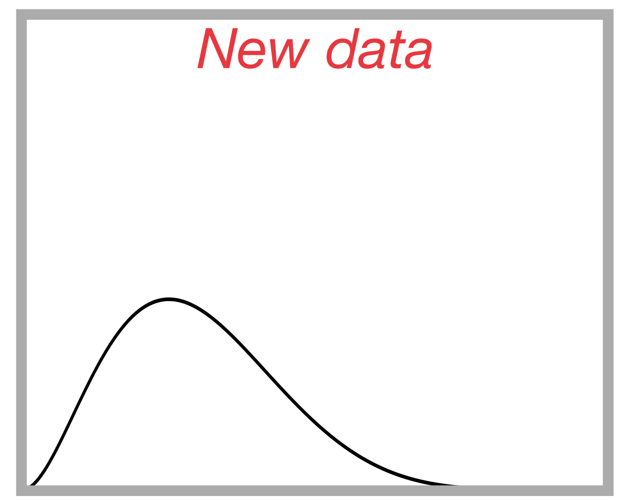
# Combining posterior distributions

Note we can combine posterior distributions from multiple identically parameterised transitions by summing the hyper-parameters

COUNT [1 – $\theta_j$] 2

COUNT [$\theta_j$] 1




COUNT [1 – $\theta_j$] 8

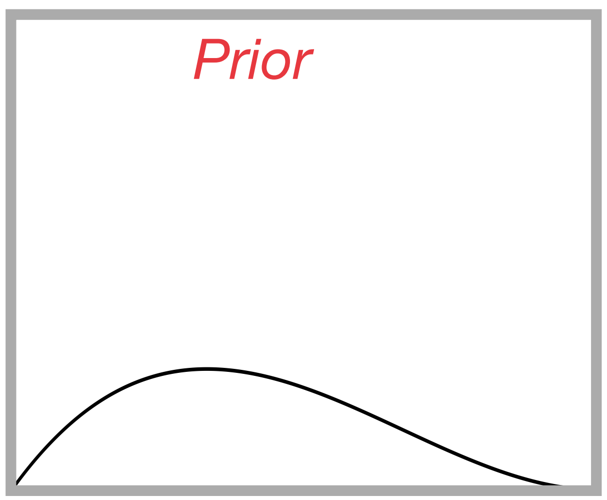
COUNT [$\theta_j$] 3

$$p(\theta_j) = \mathrm{Dir}(D_{s_1} + D_{s_2} + \alpha_{s_1})$$

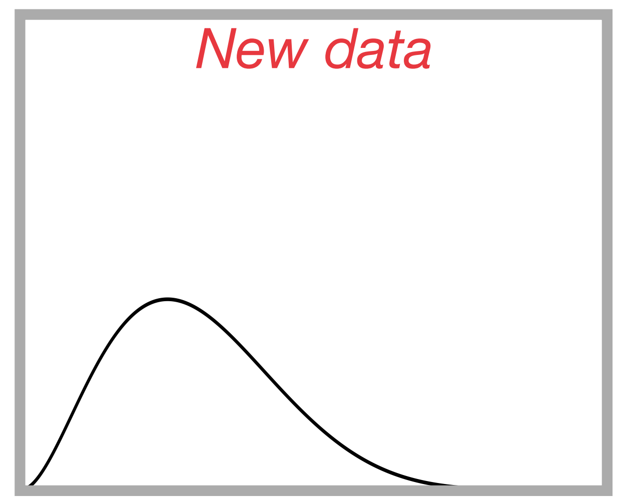
# Combining posterior distributions

Note we can combine posterior distributions from multiple identically parameterised transitions by summing the hyper-parameters

$$\frac{\text{COUNT}\,[1 - \theta_j] \quad 2}{\text{COUNT}\,[\theta_j] \quad 1}$$

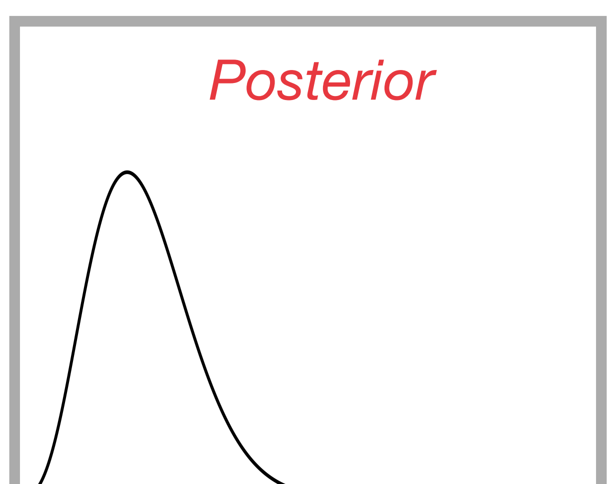



+



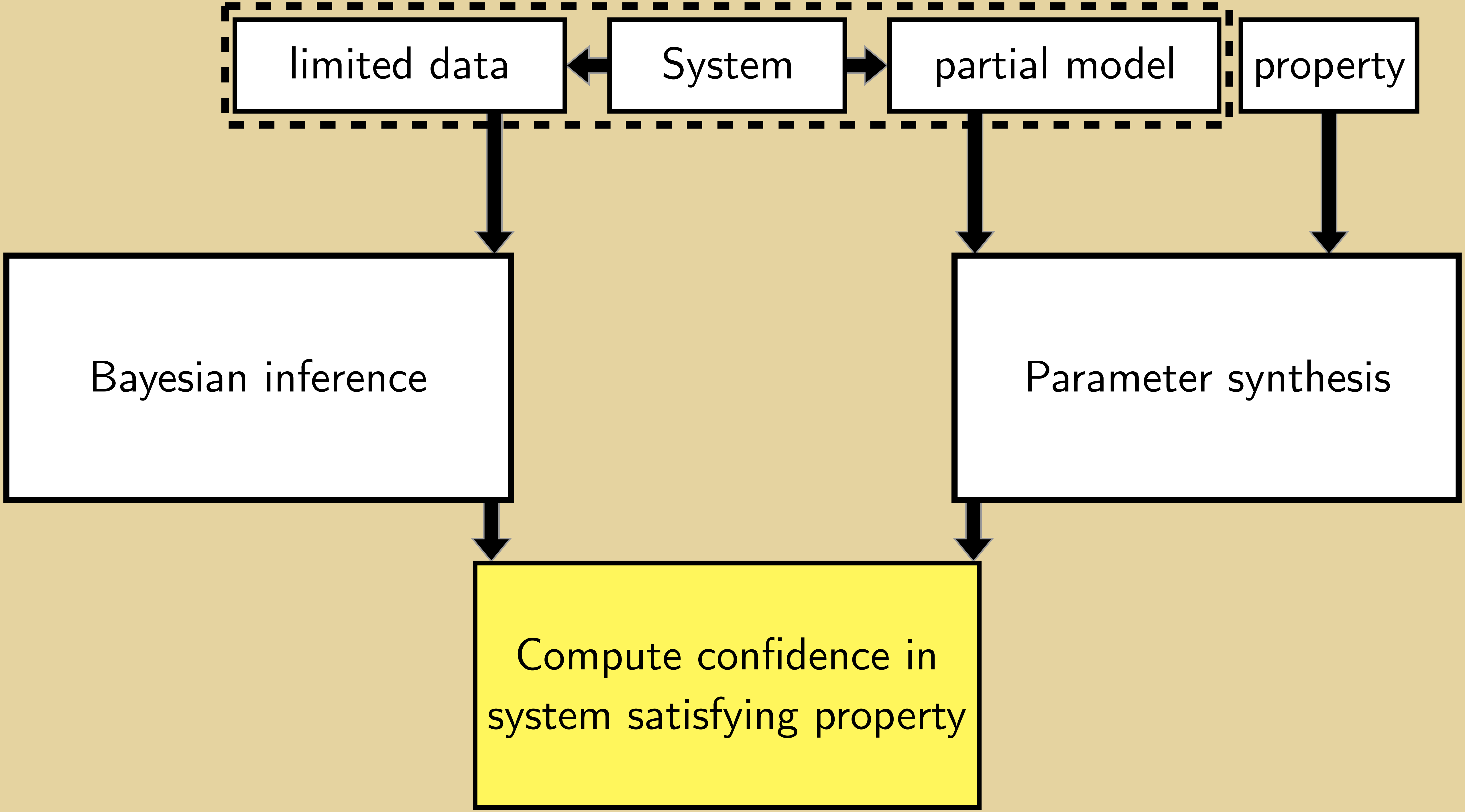

$$\frac{\text{COUNT}\,[1 - \theta_j] \quad 8}{\text{COUNT}\,[\theta_j] \quad 3}$$

$$p(\theta_j) = \mathrm{Dir}(D_{s_1} + D_{s_2} + \alpha_{s_1})$$




$$\frac{\text{COUNT}\,[1 - \theta_j] \quad 10}{\text{COUNT}\,[\theta_j] \quad 4}$$

# Overview



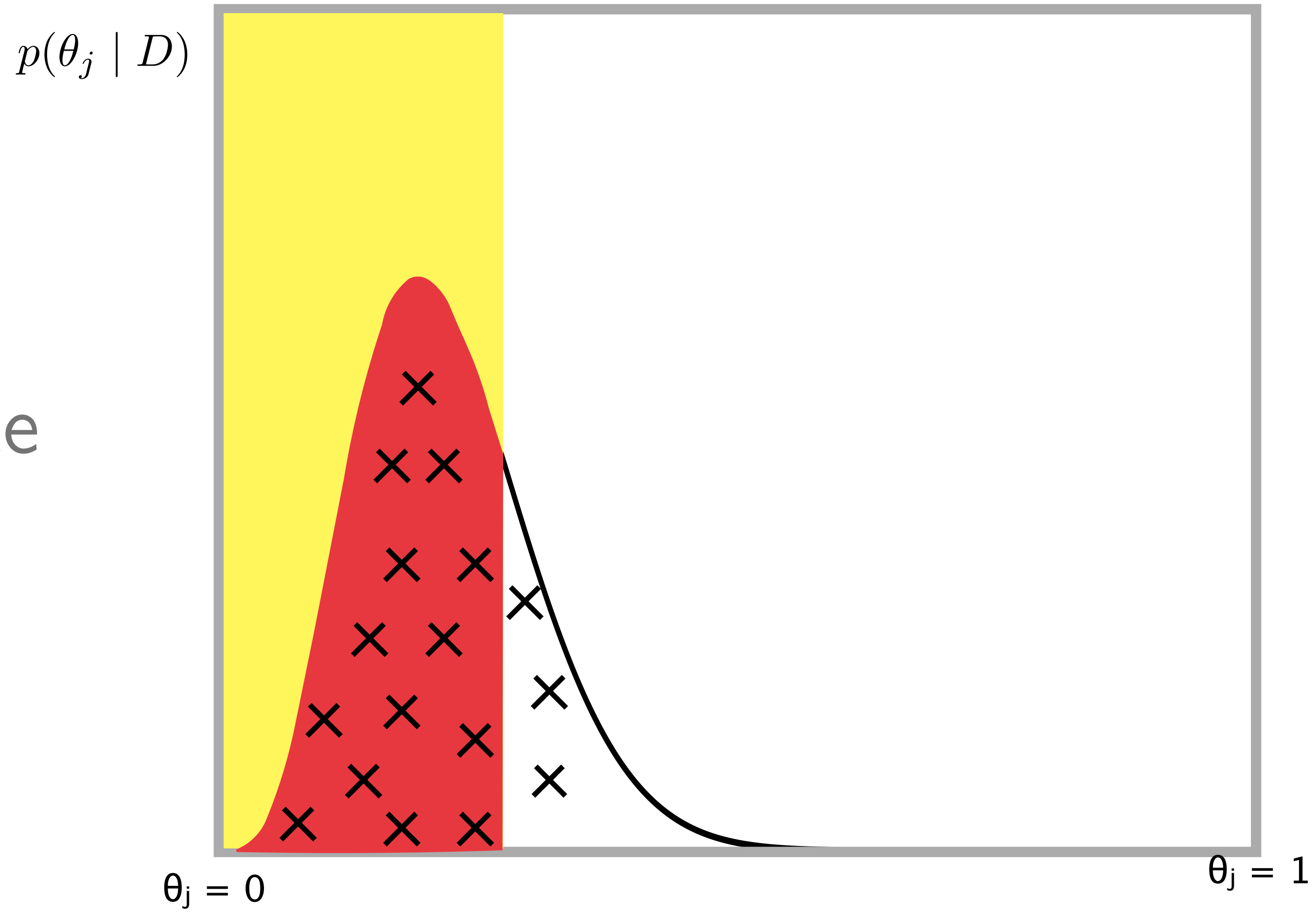
limited data
System
partial model
property
Bayesian inference
Parameter synthesis
Compute confidence in system satisfying property

# Confidence Calculation: basic pMC

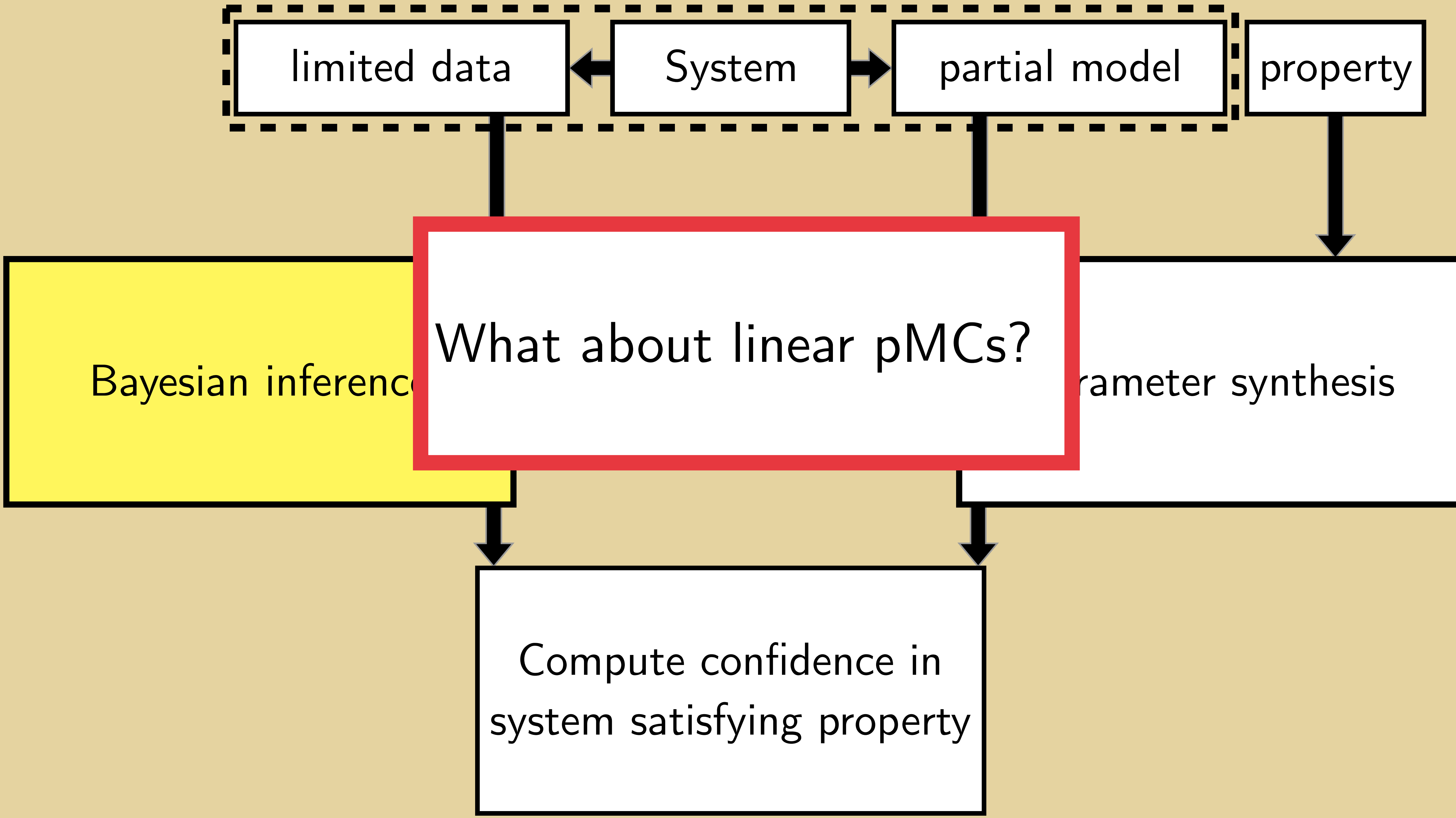
$$\mathbb{P}(\mathbf{S} \models \phi \mid D) = \int_{\Theta_\phi} p(\theta \mid D) d\theta$$

$$\Theta_\phi = \{\theta \in \Theta : \mathbf{M}(\theta) \models \phi\}$$

Note: we use simple Monte Carlo to compute the integral.

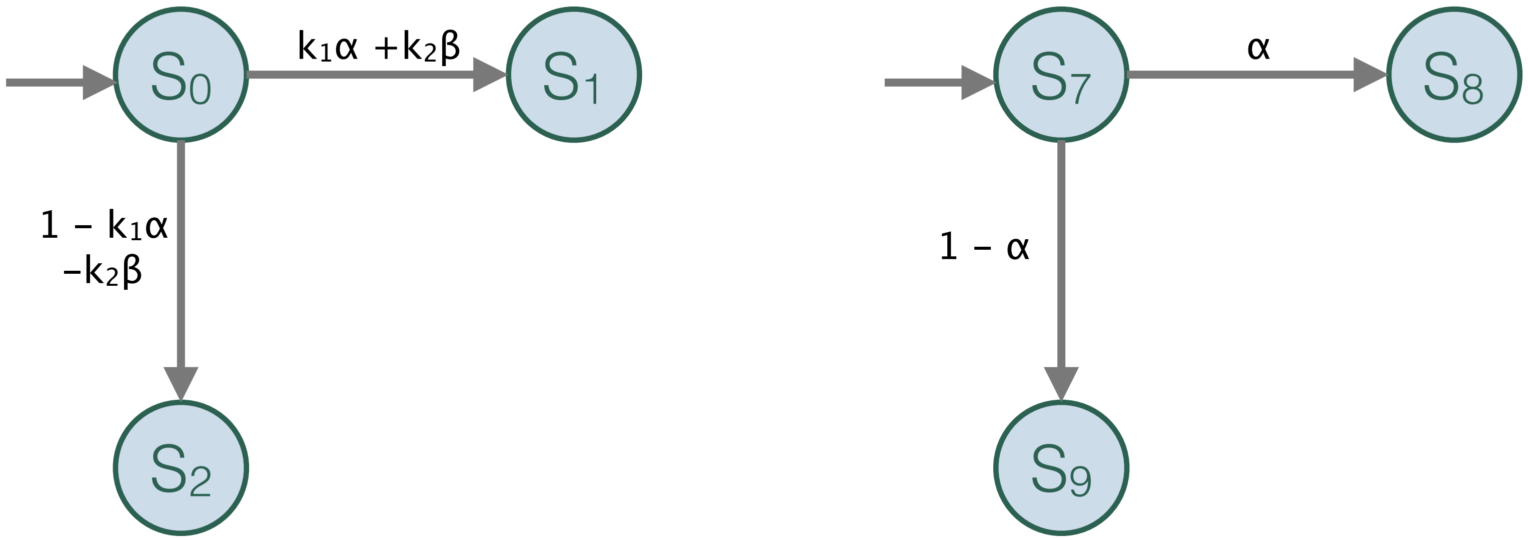


# Overview


limited data
System
partial model
property
Bayesian inference
What about linear pMCs?
rameter synthesis
Compute confidence in system satisfying property

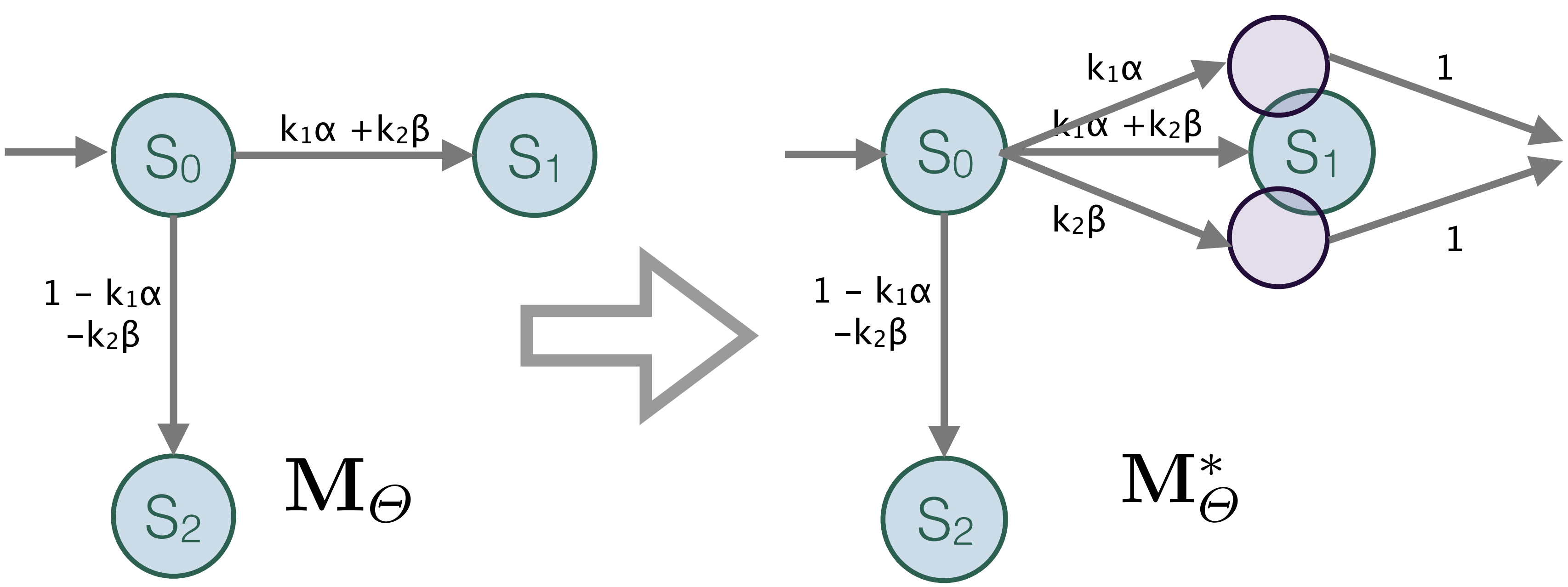
# Markov chain expansion

What if a parameter appears multiple times in a linear pMC, in different linear equations? How do we combine the posterior distributions?



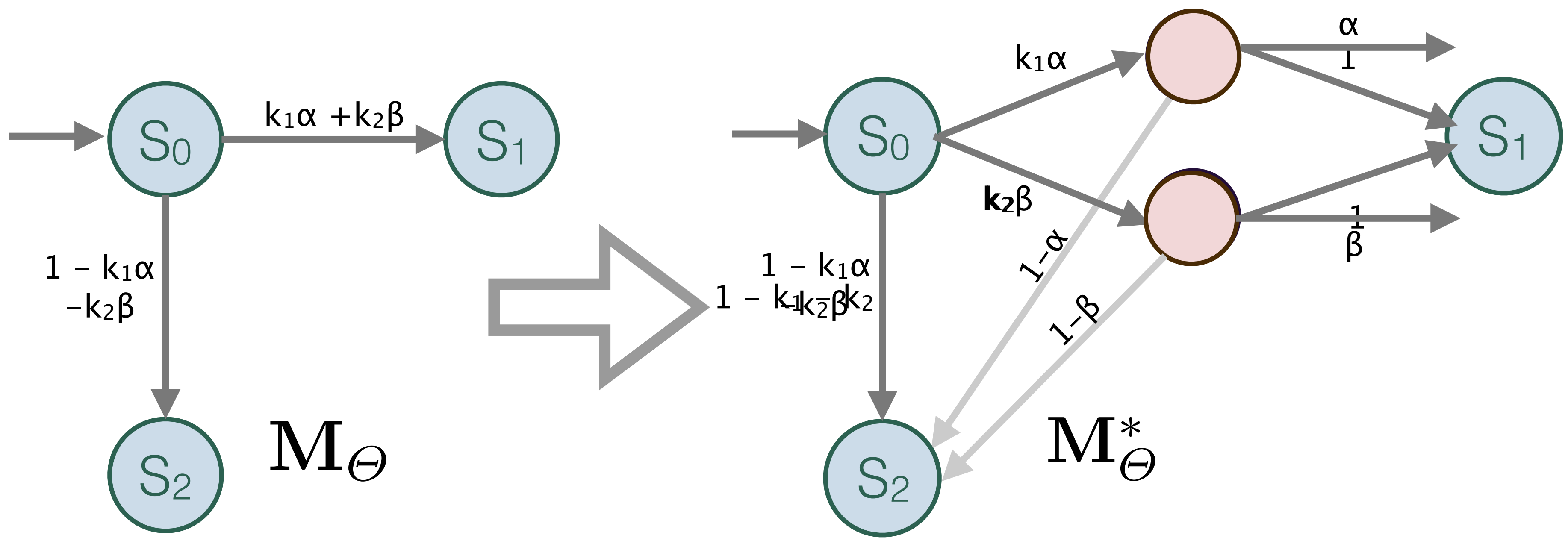
# Markov chain expansion

We "expand" the transitions with linear parameterisation, to turn the MC into a basic pMC. i.e., transitions have only one parameter.



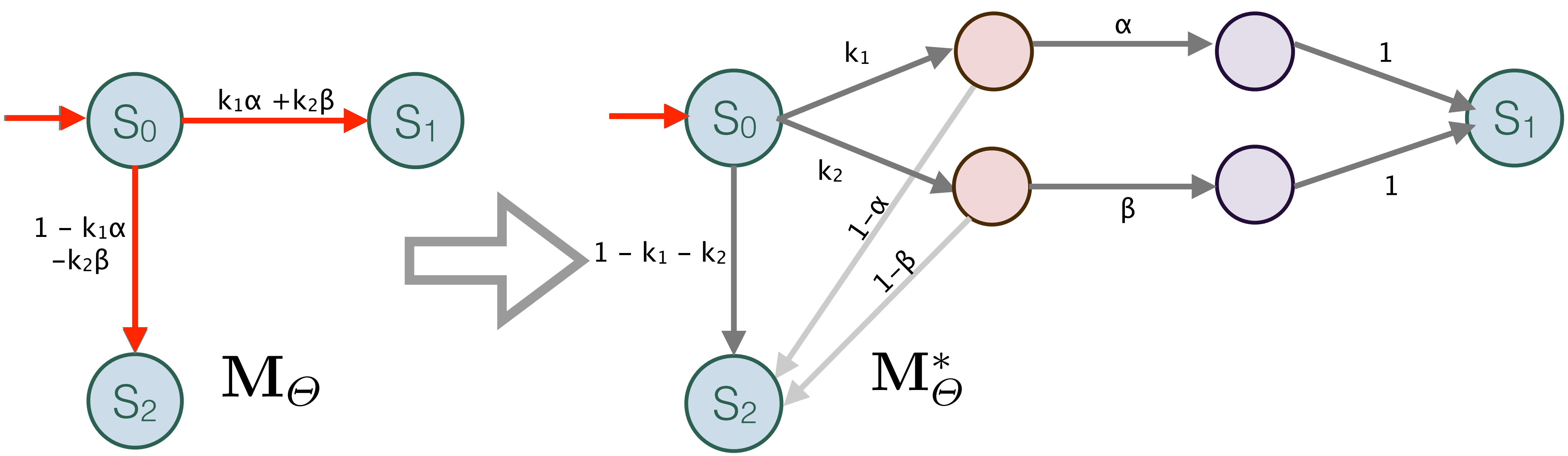
# Markov chain expansion

We "expand" the transitions with linear parameterisation, to turn the MC into a basic pMC. i.e., transitions have only one parameter.



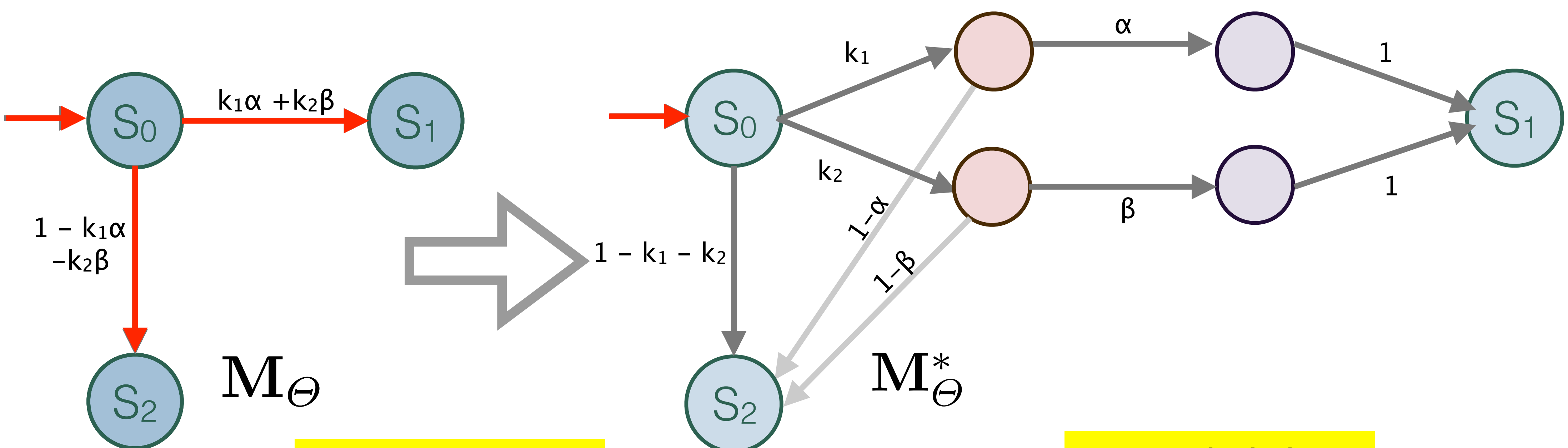
# Hidden Data

We now have a data set with gaps in. We know the transitions counts only for the original transitions.



# Hidden Data

We apply Bayes' rule




original data

expanded data

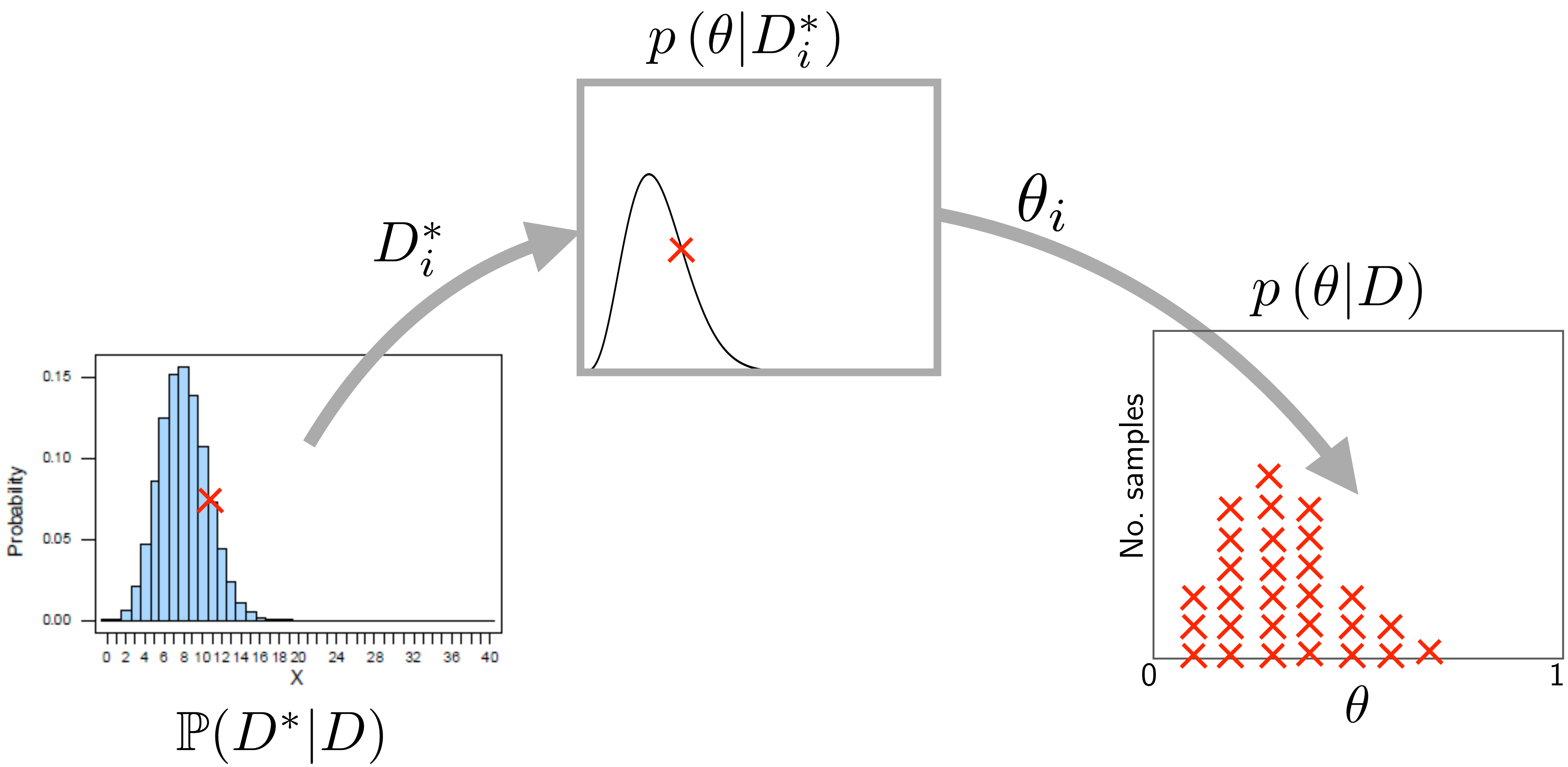
$$p\left(\theta|D\right) = \sum_{D^* \in \mathcal{D}^*} p\left(\theta|D^*\right)\mathbb{P}(D^*|D)$$

set of all possible completions of the expanded data

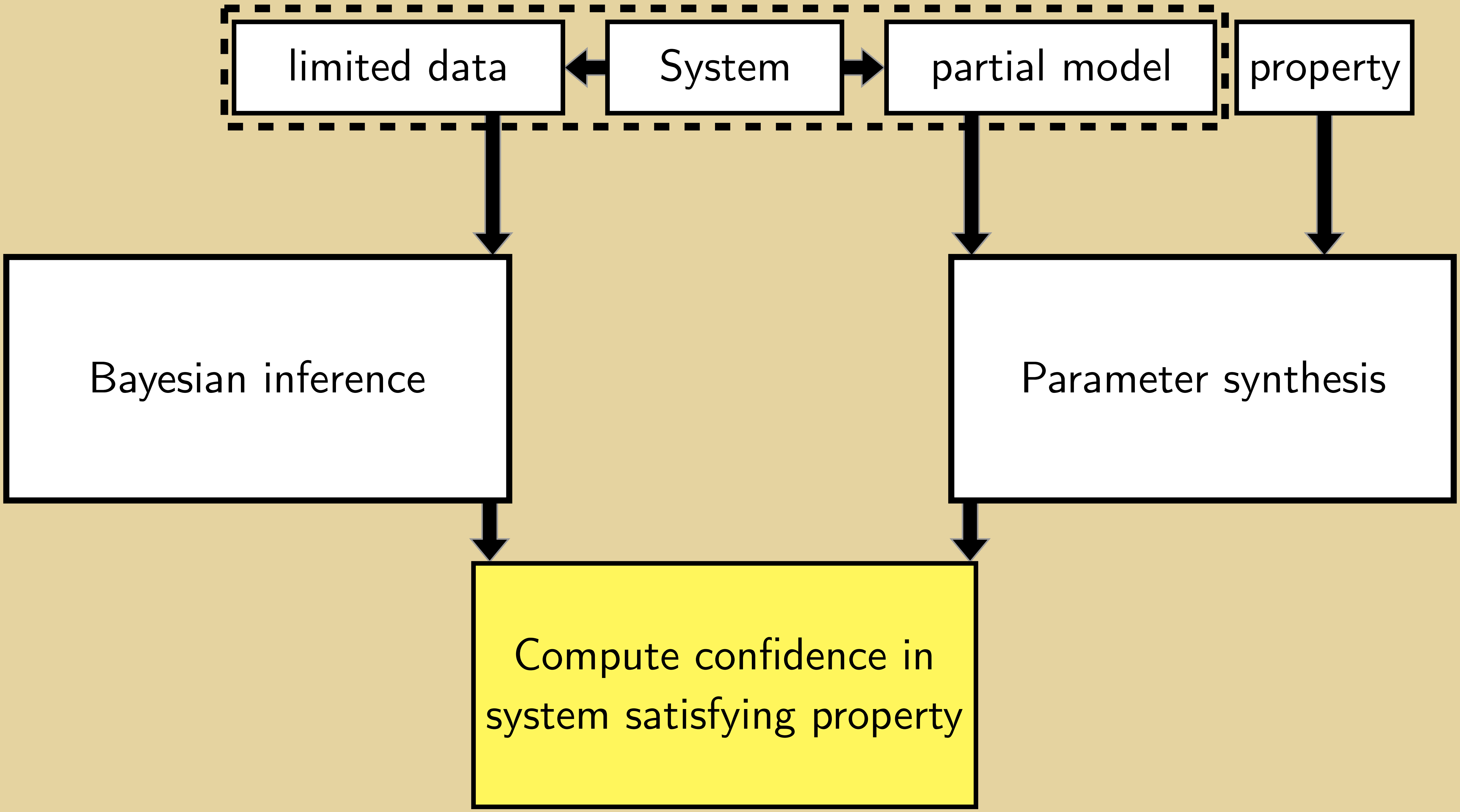
# Hidden Data

We use sampling to obtain a realisation of the posterior distribution, without evaluating the integral

$$p\left(\theta|D\right) = \sum_{D^* \in \mathcal{D}^*} p\left(\theta|D^*\right) \mathbb{P}(D^*|D)$$



# Overview

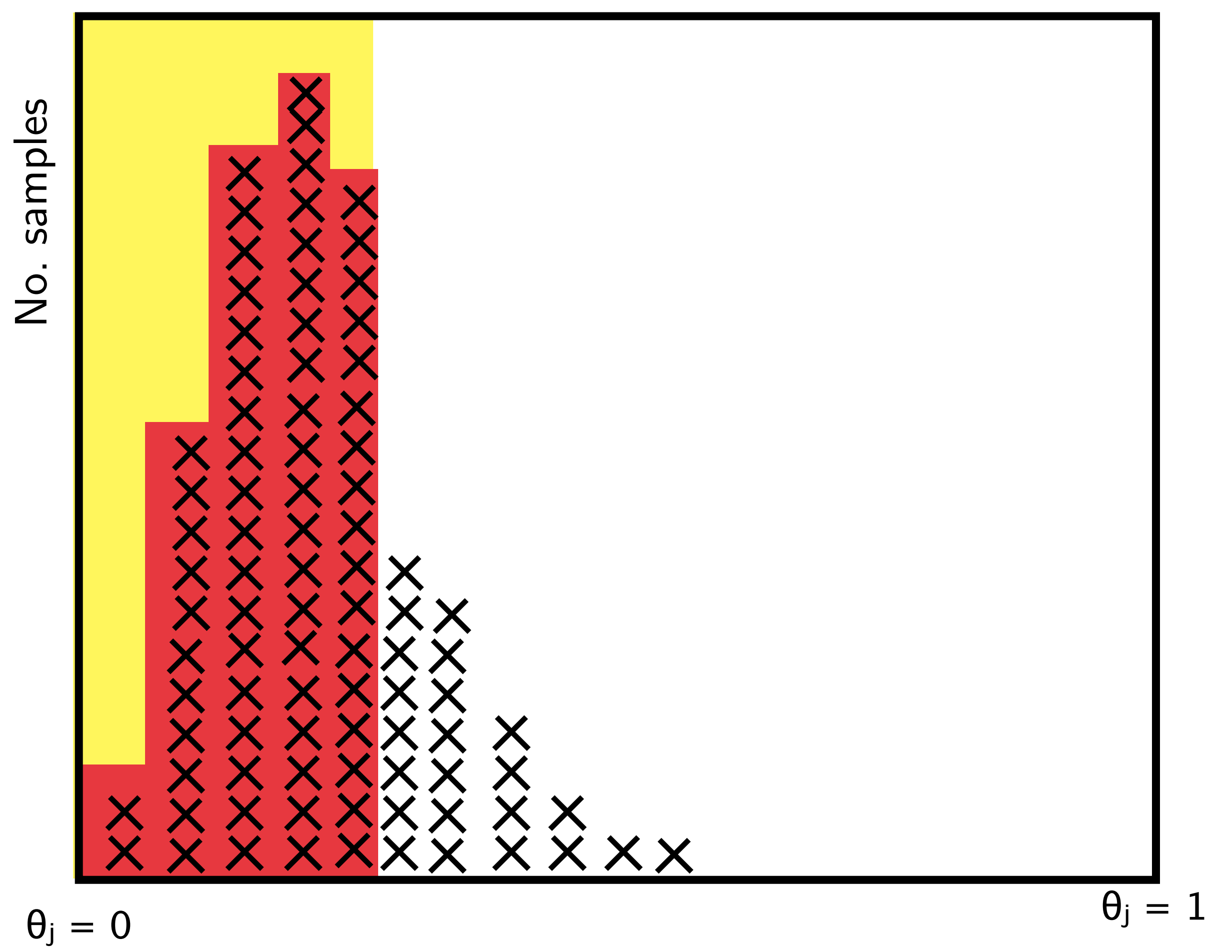


limited data
System
partial model
property
Bayesian inference
Parameter synthesis
Compute confidence in system satisfying property

# Confidence Calculation

$$\mathbb{P}(\mathbf{S} \models \phi \mid D) = \int_{\Theta_\phi} p(\theta \mid D)\, d\theta$$

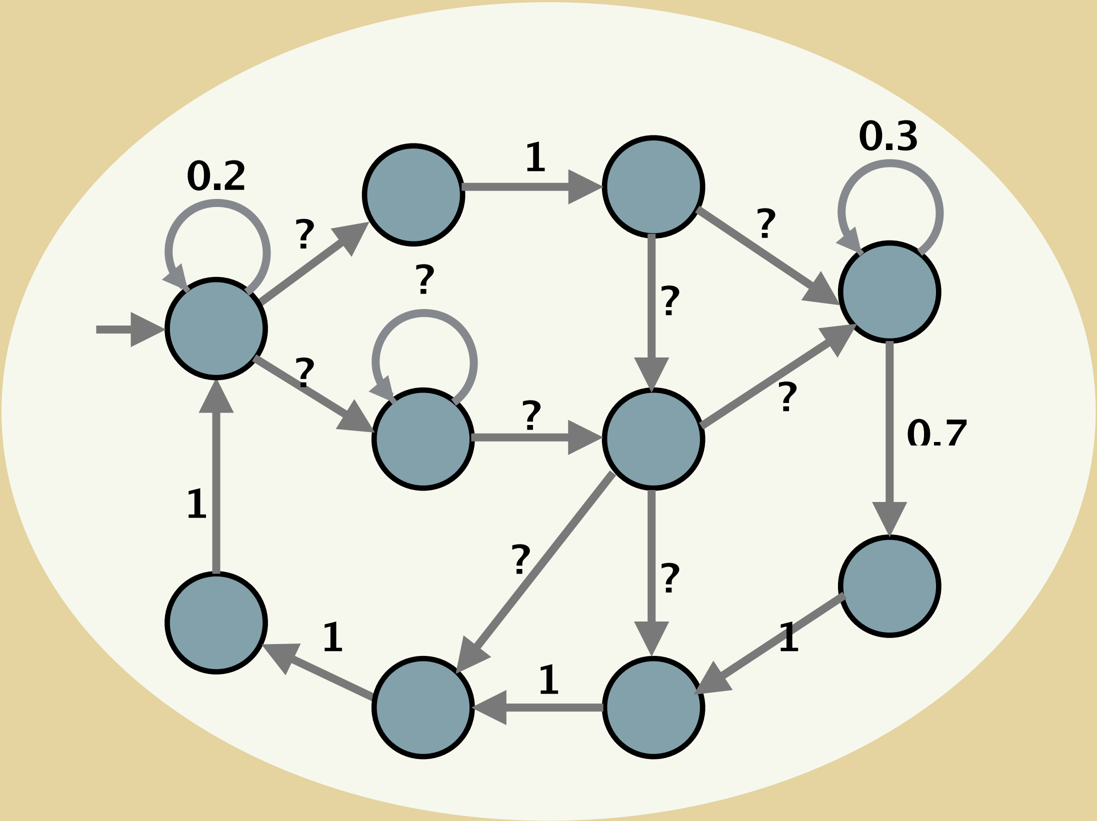
$$\Theta_\phi = \{\theta \in \Theta : \mathbf{M}(\theta) \models \phi\}$$



# Case Study

We run our approach over linear and basic parametric Markov chains, with a range of parameter values.

We implement a simple “black-box” statistical model checking algorithm for comparison



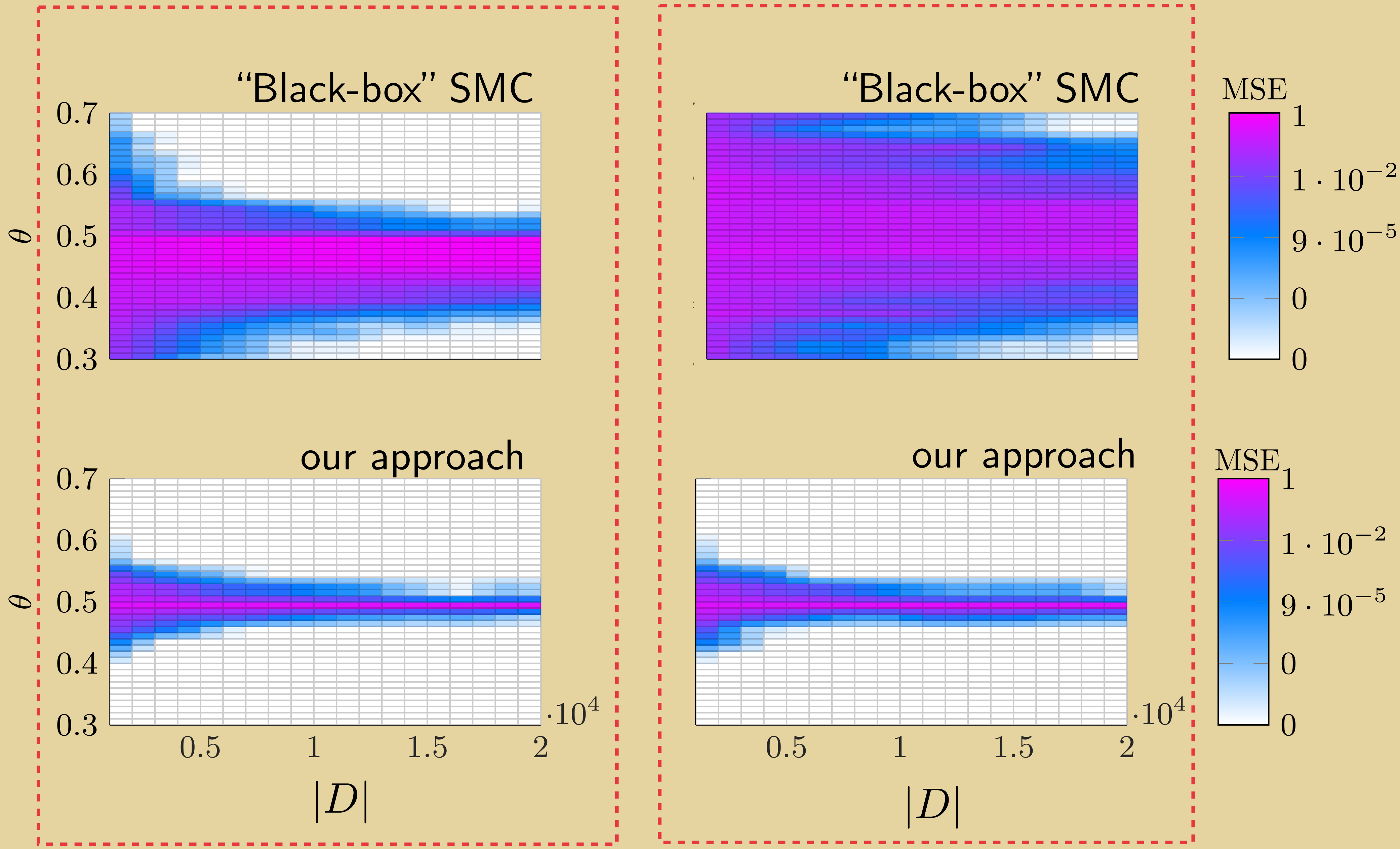
# Case Study

For our pMC and property: $\Theta_\phi = [0.5, 1]$

We compute a mean squared error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^{\overset{\text{number of experiments}}{n}} (\underbrace{Y_{true}}_{Y_{true} = \begin{cases} 0 & \text{if } \theta \leq 0.5, \\ 1 & \text{if } \theta > 0.5, \end{cases}} - \overset{i\text{-th run } \mathbb{P}(\mathbf{M}_\theta \models \phi)}{Y_i})^2$$

# Case Study



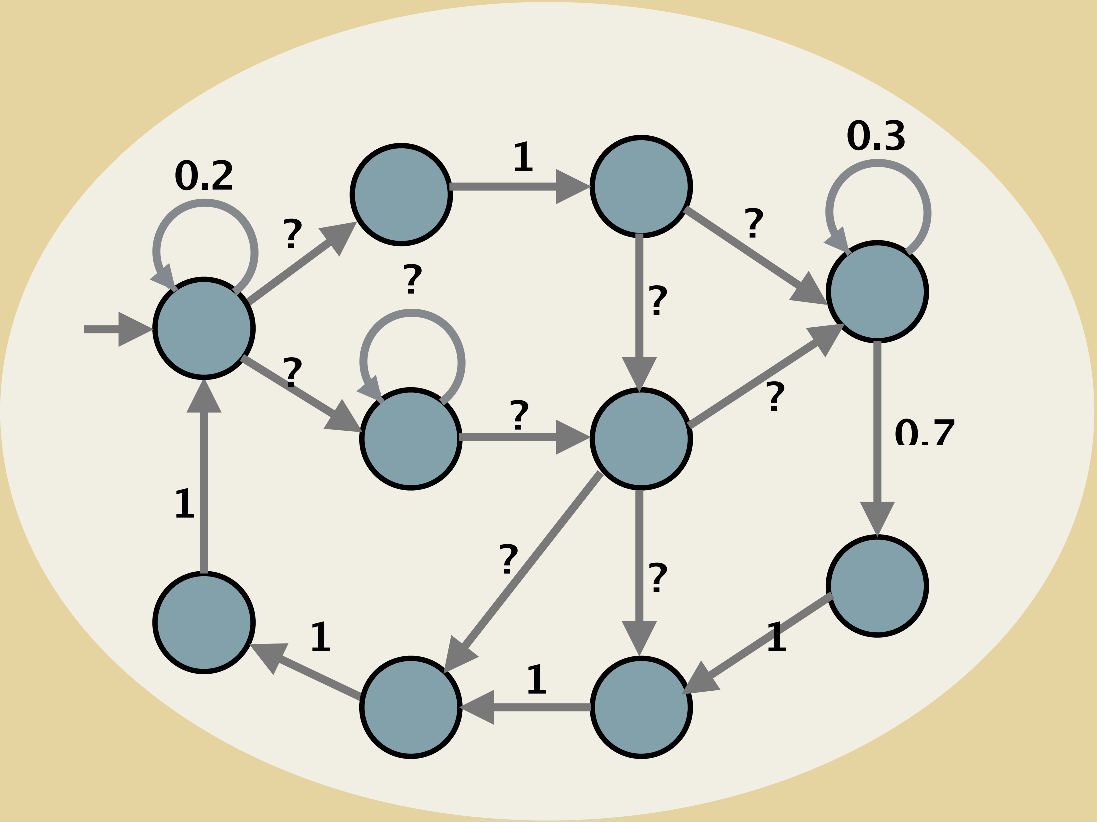

Data made up of traces of length 10

Data made up of traces of length 100

# Case Study

Our approach produces more accurate results with less data

We use information more efficiently: for SMC a “unit” of information is one trace; for us a “unit” of information is one parameterised transition



# Next steps

- Integration of alternative parameter synthesis techniques
- Non-linearly parameterised Markov chains
- External non-determinism (parameterised MDPs)
- Addition of Bayesian hypothesis testing

# Conclusions

- We presented a data-based verification approach
- Addresses model-checking with partial models and limited data
- Promises greater accuracy than black-box SMC

# References

1. This framework was originally proposed in *"Data-driven property verification of grey-box systems by Bayesian experiment design", S. Haesaert, P.M.J. Van den Hof, and A. Abate*

2. We use PRISM's parametric model checking tool: *"PARAM: a model checker for parametric Markov models", E.M. Hahn, H. Hermanns, B. Wachter, and L. Zhang*