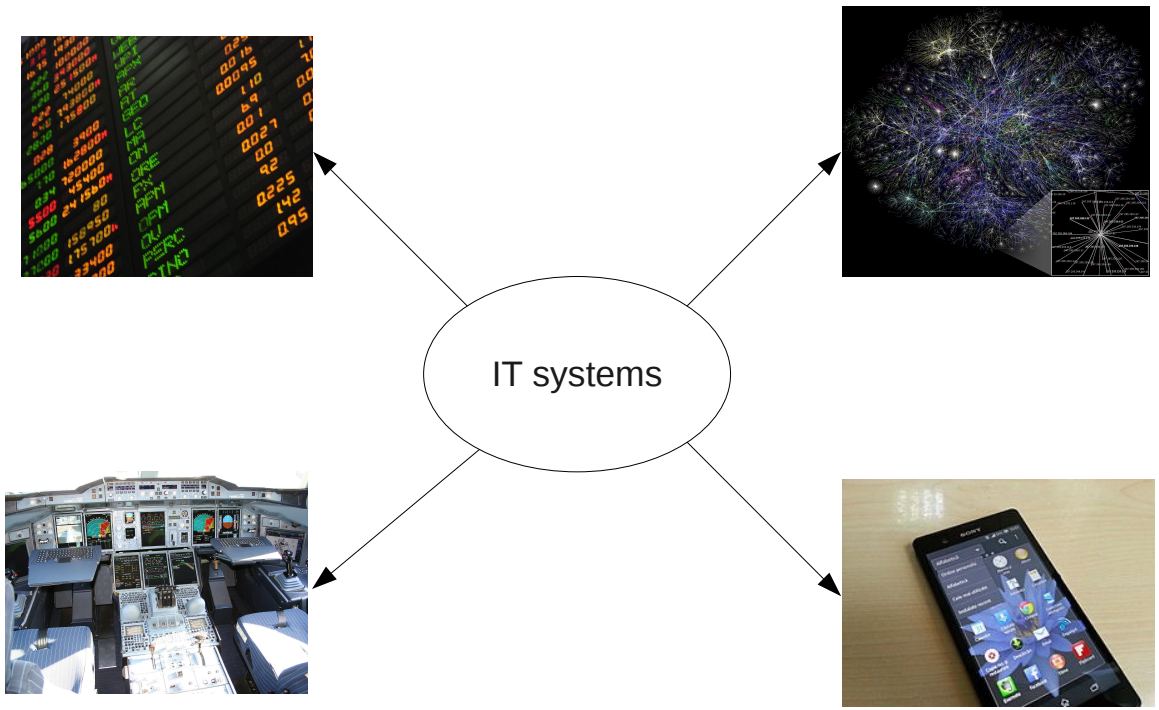
A Quick Tour on Statistical Model Checking

Axel Legay

12 novembre 2015

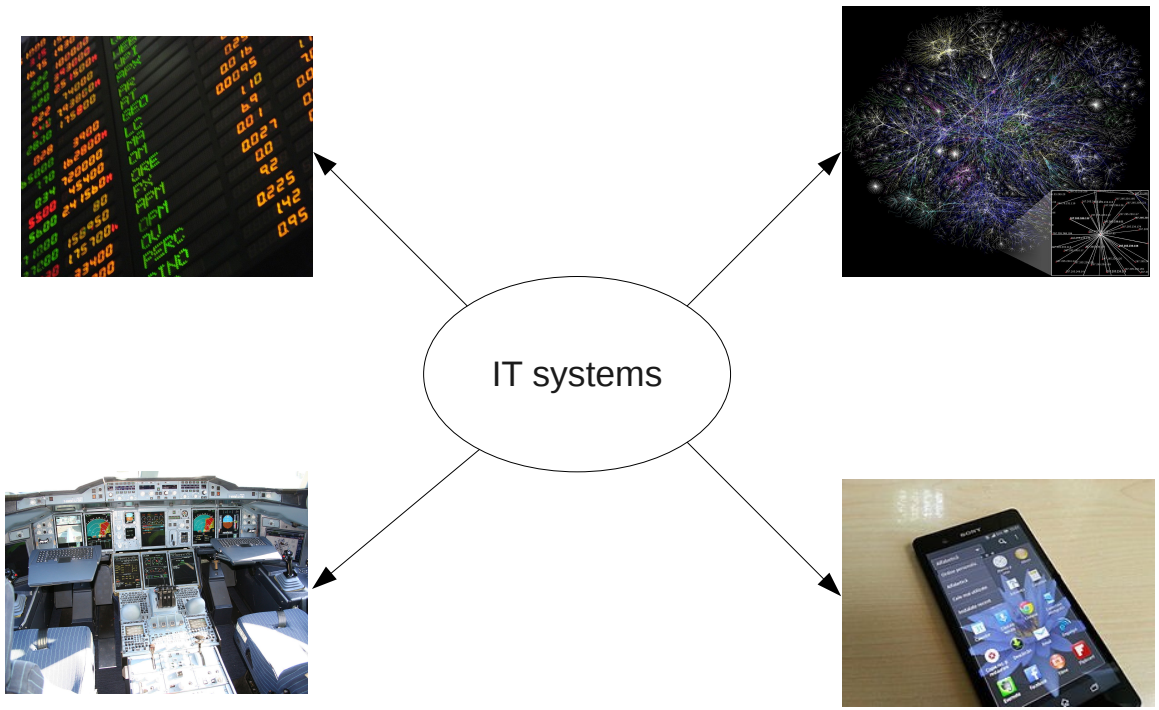
Context of System Verification

- IT systems
  - Omnipresent
  - More and more complex
- Major quality indications: absence of errors



Context of System Verification

- IT systems
  - Omnipresent
  - More and more complex
- Major quality indications: absence of errors



Context of System Verification

- IT systems
  - Omnipresent
  - More and more complex
- Major quality indications: absence of errors

Motivation for System Verification: economical issues

Software bugs can be costly

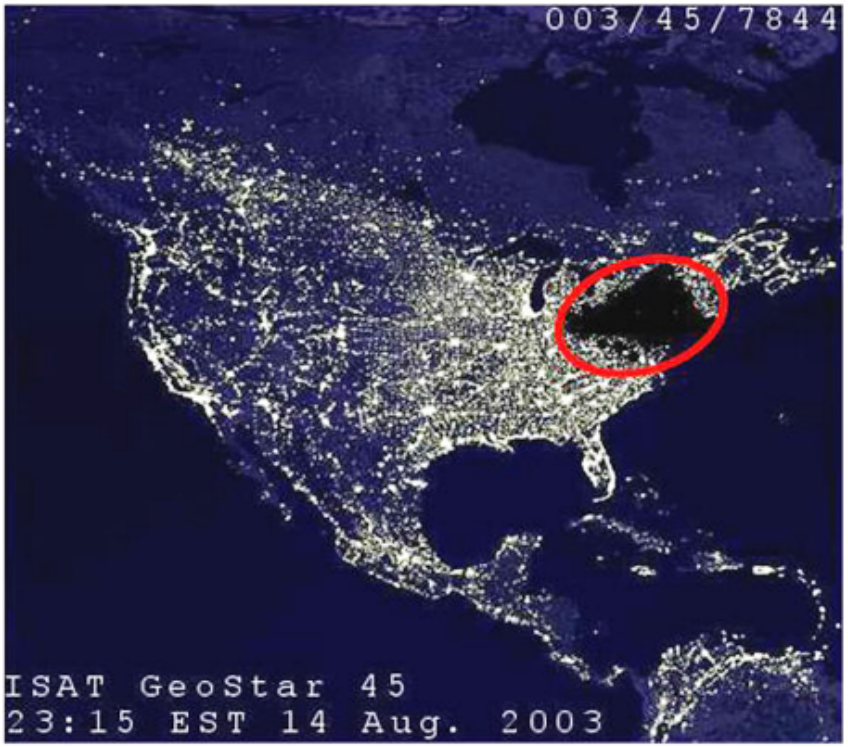
- Worldwide economy: Software errors cost 312 billion dollars in 2012

- The 2003 North-American blackout
- Concurrency bug
- Breakdown of 256 power plants
- Cost: 6 billion dollars

Motivation for System Verification: economical issues

Software bugs can be costly

- Worldwide economy: Software errors cost 312 billion dollars in 2012

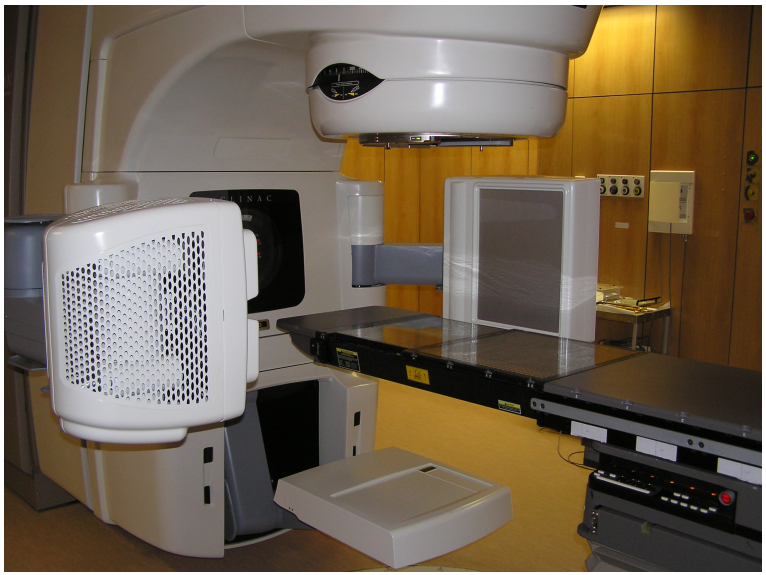


- The 2003 North-American blackout
- Concurrency bug
- Breakdown of 256 power plants
- Cost: 6 billion dollars

Motivation for System Verification: fatal defects

Errors can have dramatic consequences

Ariane 5, Therac 25...



Sample of Validation Techniques

- Various techniques:
  - Code reviewing
  - Testing
  - Formal methods: static analysis, model checking...
- Focus on Model Checking
  - Automated Technique
  - Exhaustive Verification
  - Property specification: Temporal Logic

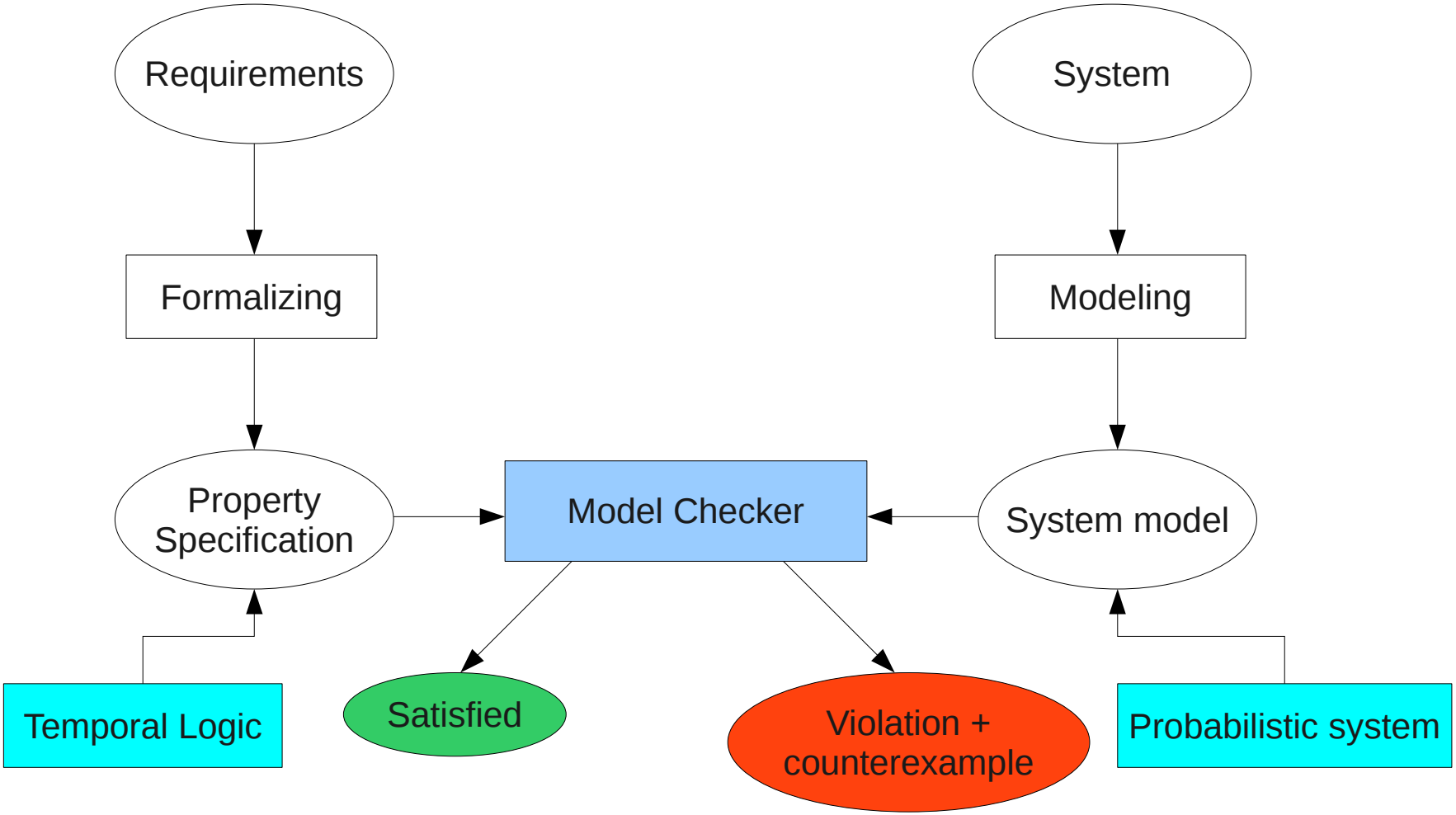
Sample of Validation Techniques

- Various techniques:
  - Code reviewing
  - Testing
  - Formal methods: static analysis, model checking...
- Focus on Model Checking
  - Automated Technique
  - Exhaustive Verification
  - Property specification: Temporal Logic

Sample of Validation Techniques

- Various techniques:
  - Code reviewing
  - Testing
  - Formal methods: static analysis, model checking...
- Focus on Model Checking
  - Automated Technique
  - Exhaustive Verification
  - Property specification: Temporal Logic

(Probabilistic) Model Checking overview



Stochastic Systems

Systems that contain probabilistic features

Probabilistic aspects are central in:

- Performance Analysis, Queuing theory
- Systems biology
- Social and economical systems

=> Extension of verification methods for quantitative analysis of Markovian systems

Stochastic Systems

Systems that contain probabilistic features

Probabilistic aspects are central in:

- Performance Analysis, Queuing theory
- Systems biology
- Social and economical systems

=> Extension of verification methods for quantitative analysis of Markovian systems

Stochastic Systems

Systems that contain probabilistic features

Probabilistic aspects are central in:

- Performance Analysis, Queuing theory
- Systems biology
- Social and economical systems

=> Extension of verification methods for quantitative analysis of Markovian systems

Specification of IT systems

- **Strict requirements**
  - offers a strict guarantee that “there is no failure”
- **Fault tolerance**
  - “90% of the components of the system are operational.”
- Qualitative analysis
  - “Is the probability of false alarm lower than 0.001?”
- Quantitative analysis
  - “What is the probability of false alarm?”

Properties should be described in a precise manner.

Temporal Logic appropriate for specifying requirements over time.

Specification of IT systems

- **Strict requirements**
  - offers a strict guarantee that “there is no failure”
- **Fault tolerance**
  - “90% of the components of the system are operational.”
- Qualitative analysis
  - “Is the probability of false alarm lower than 0.001?”
- Quantitative analysis
  - “What is the probability of false alarm?”

Properties should be described in a precise manner.

Temporal Logic appropriate for specifying requirements over time.

Specification of IT systems

- **Strict requirements**
  - offers a strict guarantee that “there is no failure”
- **Fault tolerance**
  - “90% of the components of the system are operational.”
- Qualitative analysis
  - “Is the probability of false alarm lower than 0.001?”
- Quantitative analysis
  - “What is the probability of false alarm?”

Properties should be described in a precise manner.

Temporal Logic appropriate for specifying requirements over time.

Specification of IT systems

- **Strict requirements**
  - offers a strict guarantee that “there is no failure”
- **Fault tolerance**
  - “90% of the components of the system are operational.”
- Qualitative analysis
  - “Is the probability of false alarm lower than 0.001?”
- Quantitative analysis
  - “What is the probability of false alarm?”

Properties should be described in a precise manner.

Temporal Logic appropriate for specifying requirements over time.

Specification of IT systems

- **Strict requirements**
  - offers a strict guarantee that “there is no failure”
- **Fault tolerance**
  - “90% of the components of the system are operational.”
- Qualitative analysis
  - “Is the probability of false alarm lower than 0.001?”
- Quantitative analysis
  - “What is the probability of false alarm?”

Properties should be described in a precise manner.

Temporal Logic appropriate for specifying requirements over time.

Specification of IT systems

- **Strict requirements**
  - offers a strict guarantee that “there is no failure”
- **Fault tolerance**
  - “90% of the components of the system are operational.”
- Qualitative analysis
  - “Is the probability of false alarm lower than 0.001?”
- Quantitative analysis
  - “What is the probability of false alarm?”

Properties should be described in a precise manner.

Temporal Logic appropriate for specifying requirements over time.

Probabilistic Bounded Linear Temporal Logic

- **BLTL:** $\phi := \alpha \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \bigcirc\phi \mid \Diamond^{\leq t}\phi \mid \Box^{\leq t}\phi \mid \phi U^{\leq t}\phi$
  - $\bigcirc$: "**next**" operator
  - $\Diamond$: "**eventually**" operator
  - $\Box$: "**always**" operator
  - U: "**until**" operator
  - $\Diamond^{\leq t}(x \geq 0)$: x will be eventually greater than 0 within t time units
- **Probabilistic BLTL.** extension of BLTL with a probabilistic operator
  - $P_{\sim\theta}\,\phi$ with $\sim \in \{<, >, =\}$ and $\theta \in [0, 1]$
  - $P_{\leq 0.2}(\Diamond^{\leq t}\Box^{\leq s}(x \geq 0))$: The probability that x will be eventually greater than 0 during s time units within t time units is lower than 0.2.

Probabilistic Bounded Linear Temporal Logic

- **BLTL:** $\phi := \alpha \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \bigcirc\phi \mid \Diamond^{\leq t}\phi \mid \Box^{\leq t}\phi \mid \phi U^{\leq t}\phi$
  - $\bigcirc$: "**next**" operator
  - $\Diamond$: "**eventually**" operator
  - $\Box$: "**always**" operator
  - U: "**until**" operator
  - $\Diamond^{\leq t}(x \geq 0)$: x will be eventually greater than 0 within t time units
- **Probabilistic BLTL.** extension of BLTL with a probabilistic operator
  - $P_{\sim\theta}\,\phi$ with $\sim \in \{<, >, =\}$ and $\theta \in [0, 1]$
  - $P_{\leq 0.2}(\Diamond^{\leq t}\Box^{\leq s}(x \geq 0))$: The probability that x will be eventually greater than 0 during s time units within t time units is lower than 0.2.

Probabilistic Bounded Linear Temporal Logic

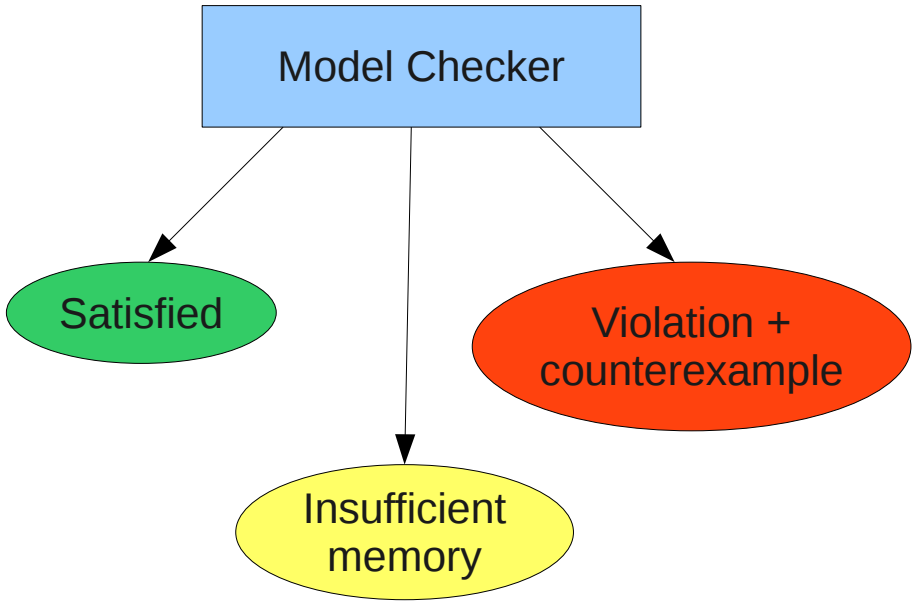
- **BLTL:** $\phi := \alpha \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \bigcirc\phi \mid \Diamond^{\leq t}\phi \mid \Box^{\leq t}\phi \mid \phi U^{\leq t}\phi$
  - $\bigcirc$: "**next**" operator
  - $\Diamond$: "**eventually**" operator
  - $\Box$: "**always**" operator
  - U: "**until**" operator
  - $\Diamond^{\leq t}(x \geq 0)$: x will be eventually greater than 0 within t time units
- **Probabilistic BLTL.** extension of BLTL with a probabilistic operator
  - $P_{\sim\theta}\,\phi$ with $\sim \in \{<, >, =\}$ and $\theta \in [0, 1]$
  - $P_{\leq 0.2}(\Diamond^{\leq t}\Box^{\leq s}(x \geq 0))$: The probability that x will be eventually greater than 0 during s time units within t time units is lower than 0.2.

Probabilistic Bounded Linear Temporal Logic

- **BLTL:** $\phi := \alpha \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \bigcirc\phi \mid \Diamond^{\leq t}\phi \mid \Box^{\leq t}\phi \mid \phi U^{\leq t}\phi$
  - $\bigcirc$: "**next**" operator
  - $\Diamond$: "**eventually**" operator
  - $\Box$: "**always**" operator
  - U: "**until**" operator
  - $\Diamond^{\leq t}(x \geq 0)$: x will be eventually greater than 0 within t time units
- **Probabilistic BLTL.** extension of BLTL with a probabilistic operator
  - $P_{\sim\theta}\,\phi$ with $\sim \in \{<, >, =\}$ and $\theta \in [0, 1]$
  - $P_{\leq 0.2}(\Diamond^{\leq t}\Box^{\leq s}(x \geq 0))$: The probability that x will be eventually greater than 0 during s time units within t time units is lower than 0.2.

Limitation of Model Checking

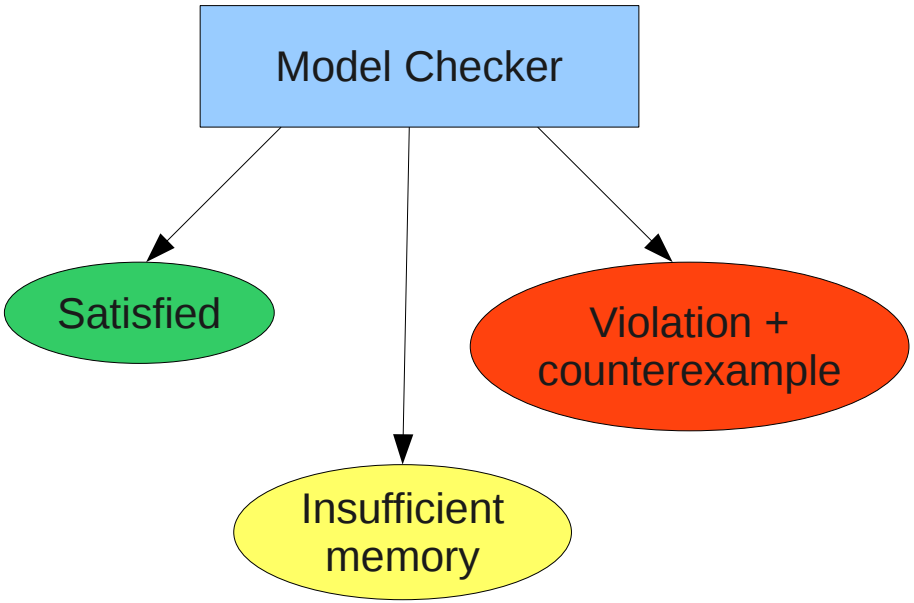
- State space explosion problem



- Prompted the recourse to statistical verification

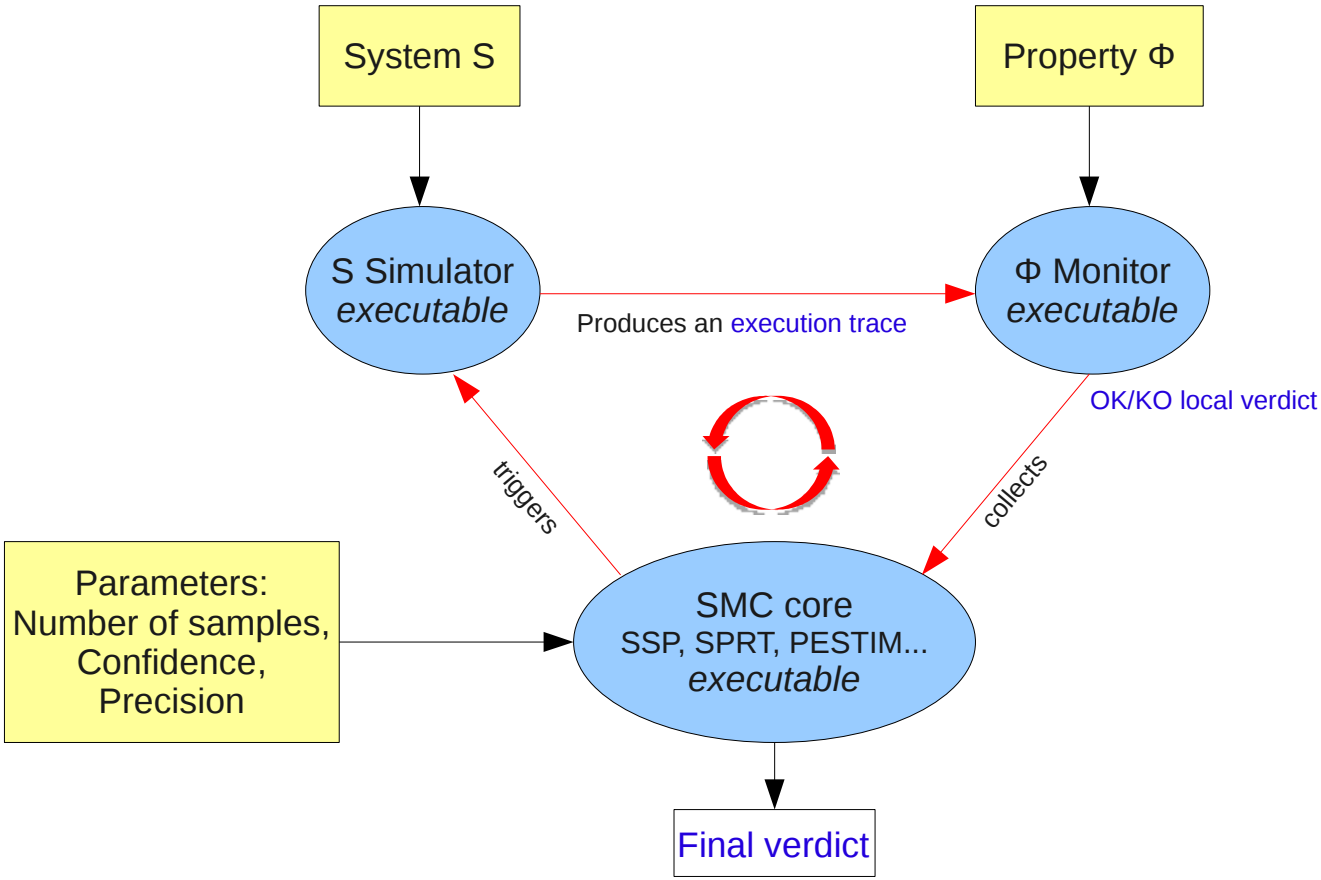
Limitation of Model Checking

- State space explosion problem

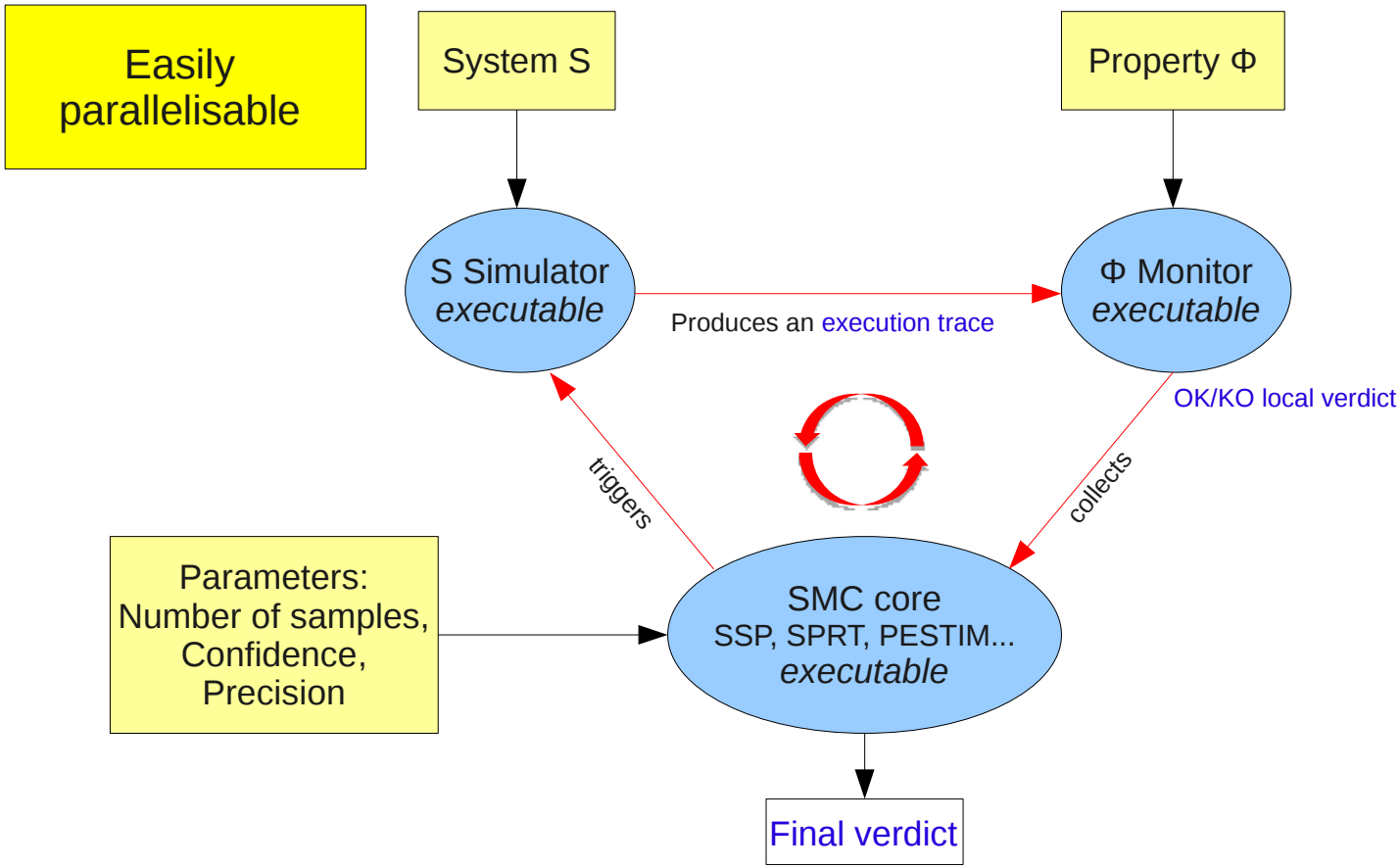


- Prompted the recourse to statistical verification

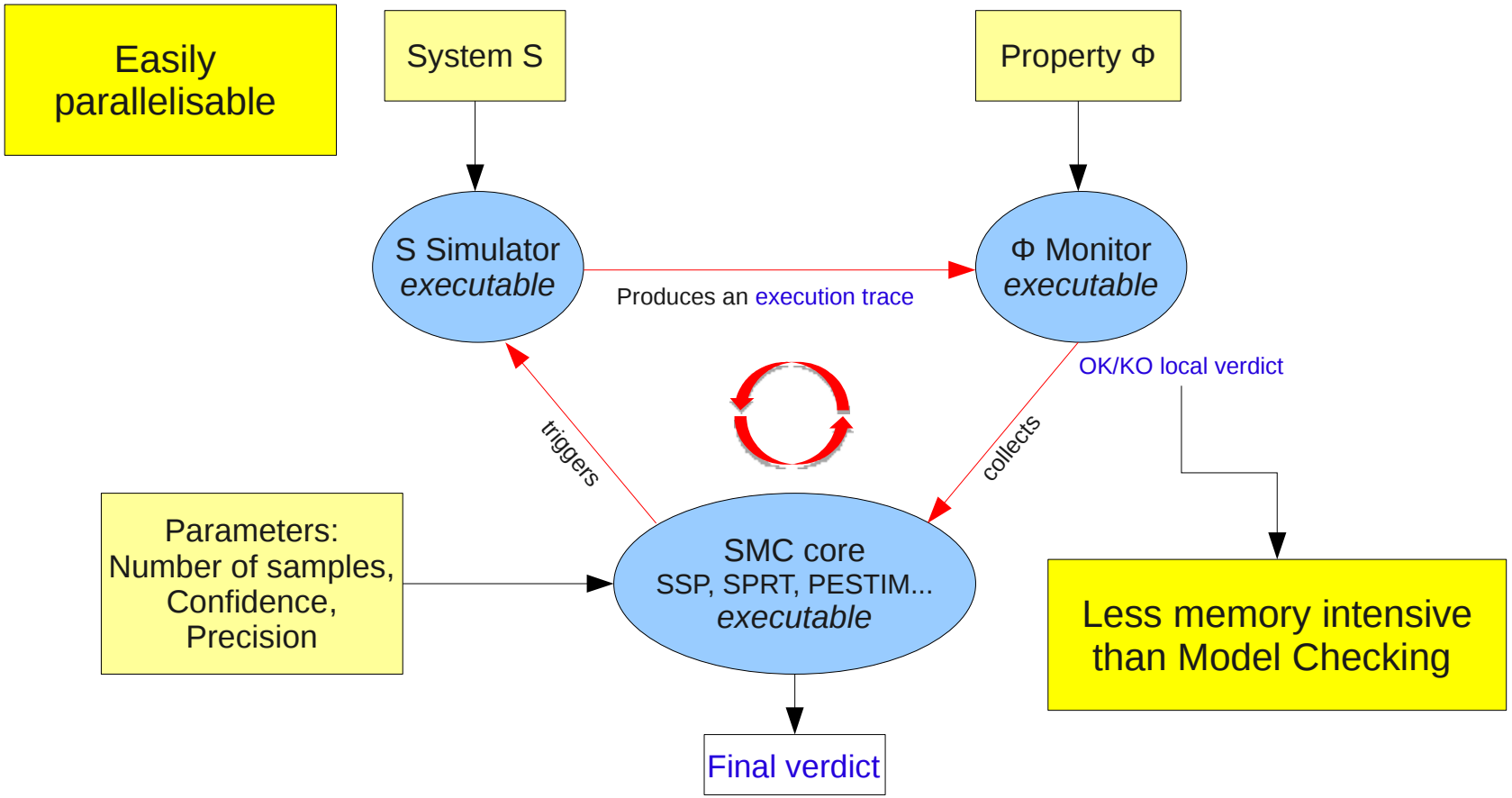
Statistical Model Checking



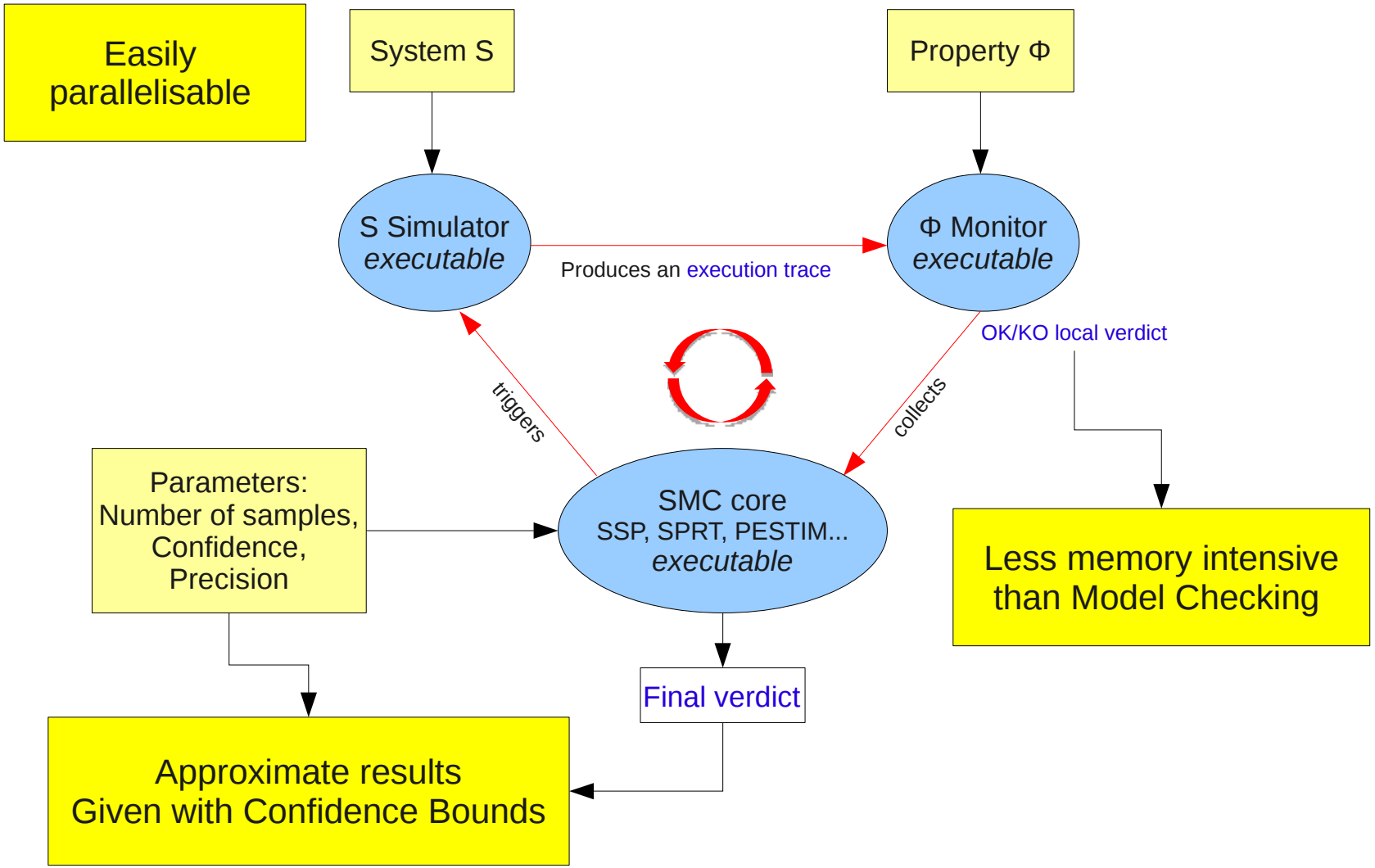
Statistical Model Checking



Statistical Model Checking



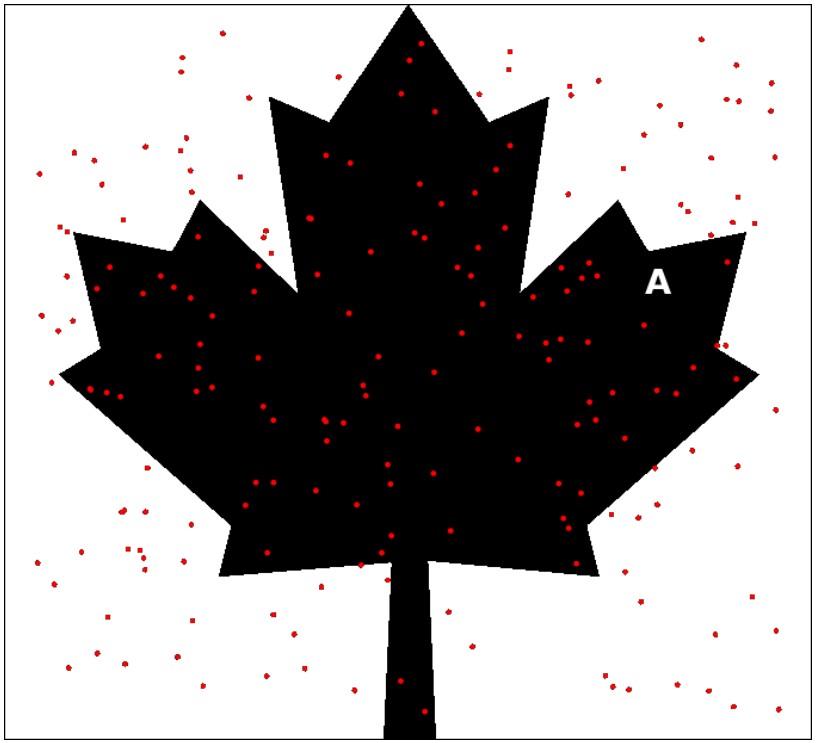
Statistical Model Checking



Algorithms

- Quantitative algorithms to estimate a probability (e.g. Monte Carlo)
- Qualitative algorithms to compare probabilities (e.g. Hypothesis Testing)
- Rare event simulation
- Cu-Sum detection

Monte Carlo estimation



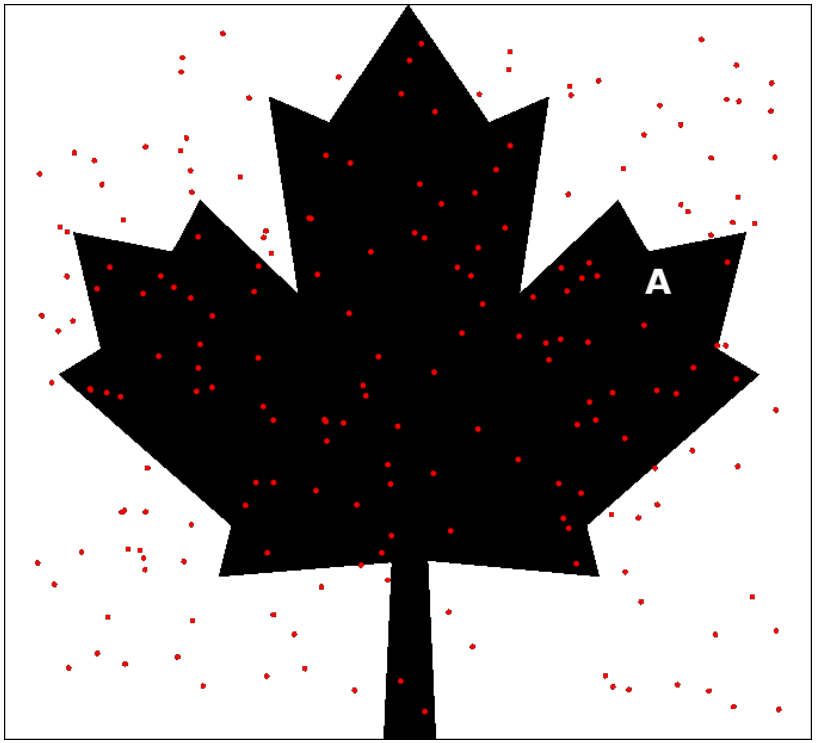
- Ω a set of paths ending uniformly in a square
- $z(\omega) = 1$ if ω ends in A, 0 otherwise.
- $\gamma = P(\text{“end in A”})$.
- Monte-Carlo estimator:

$$\tilde{\gamma}_N = \frac{1}{N} \sum_{i=1}^{N} z(\omega_i)$$

- Standard Confidence Interval = Estimator +/- Absolute Error

$$AE \propto Var(\tilde{\gamma}_N) = \frac{\sqrt{\gamma(1-\gamma)}}{\sqrt{N}}$$

Monte Carlo estimation



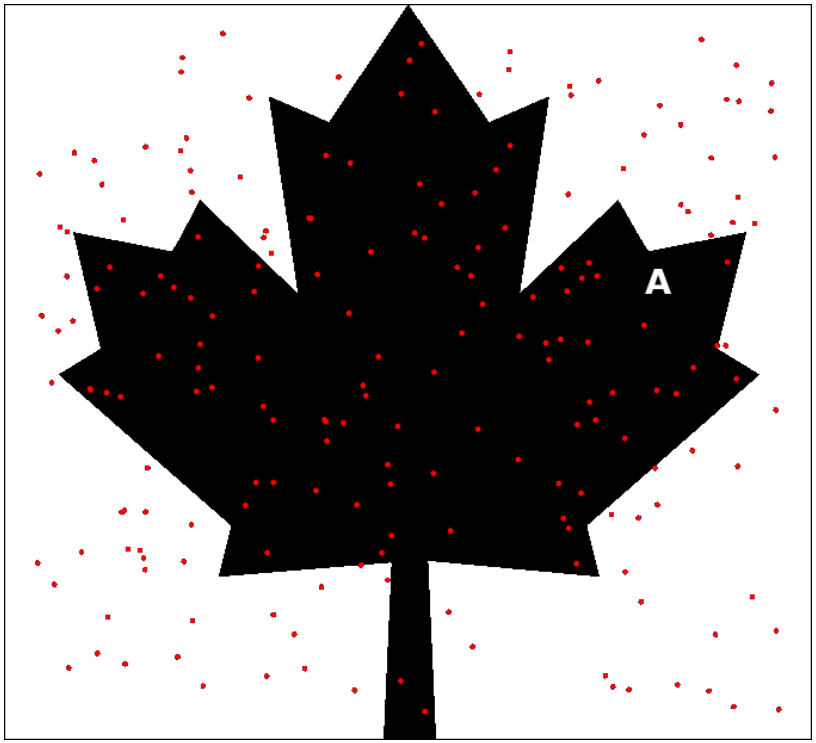
- Ω a set of paths ending uniformly in a square
- $z(\omega) = 1$ if ω ends in A, 0 otherwise.
- $\gamma = P(\text{“end in A”})$.
- Monte-Carlo estimator:

$$\tilde{\gamma}_N = \frac{1}{N} \sum_{i=1}^{N} z(\omega_i)$$

- Standard Confidence Interval = Estimator +/- Absolute Error

$$AE \propto Var(\tilde{\gamma}_N) = \frac{\sqrt{\gamma(1-\gamma)}}{\sqrt{N}}$$

Monte Carlo estimation



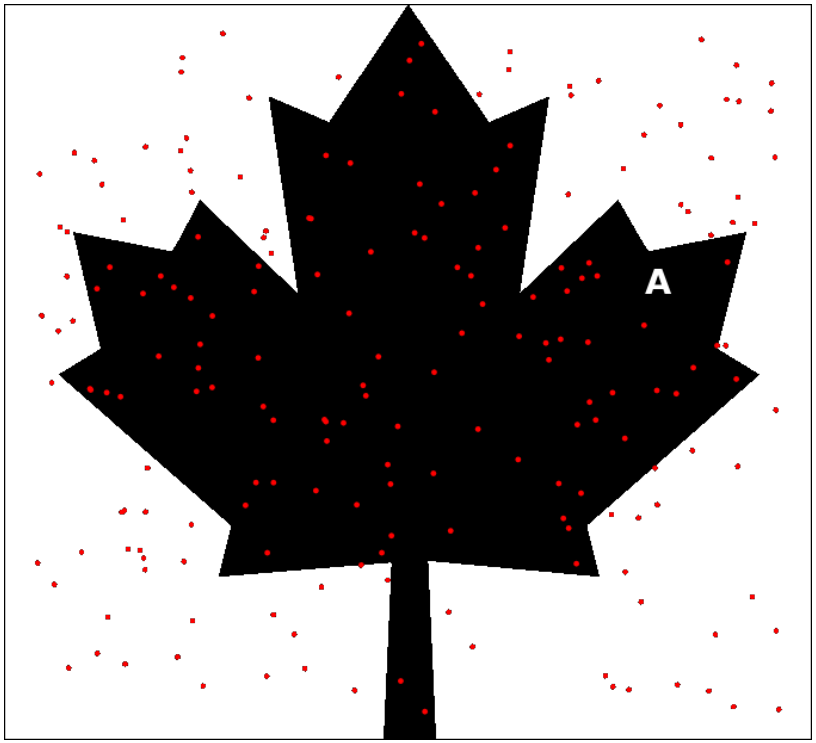
- Ω a set of paths ending uniformly in a square
- $z(\omega) = 1$ if ω ends in A, 0 otherwise.
- $\gamma = P(\text{“end in A”})$.
- Monte-Carlo estimator:

$$\tilde{\gamma}_N = \frac{1}{N} \sum_{i=1}^{N} z(\omega_i)$$

- Standard Confidence Interval = Estimator +/- Absolute Error

$$AE \propto Var(\tilde{\gamma}_N) = \frac{\sqrt{\gamma(1-\gamma)}}{\sqrt{N}}$$

Monte Carlo estimation



- Ω a set of paths ending uniformly in a square
- $z(\omega) = 1$ if ω ends in A, 0 otherwise.
- $\gamma = P(\text{“end in A”})$.
- Monte-Carlo estimator:

$$\tilde{\gamma}_N = \frac{1}{N} \sum_{i=1}^{N} z(\omega_i)$$

- Standard Confidence Interval = Estimator +/- Absolute Error

$$AE \propto Var(\tilde{\gamma}_N) = \frac{\sqrt{\gamma(1-\gamma)}}{\sqrt{N}}$$

Hypothesis Testing

$$\text{Test } H_1 : p \leq \theta \text{ against } H_0 : p > \theta$$

With (Type error):

1. α : the probability to accept H_1 while H_0 is true;
2. β : the probability to accept H_0 while H_1 is true.

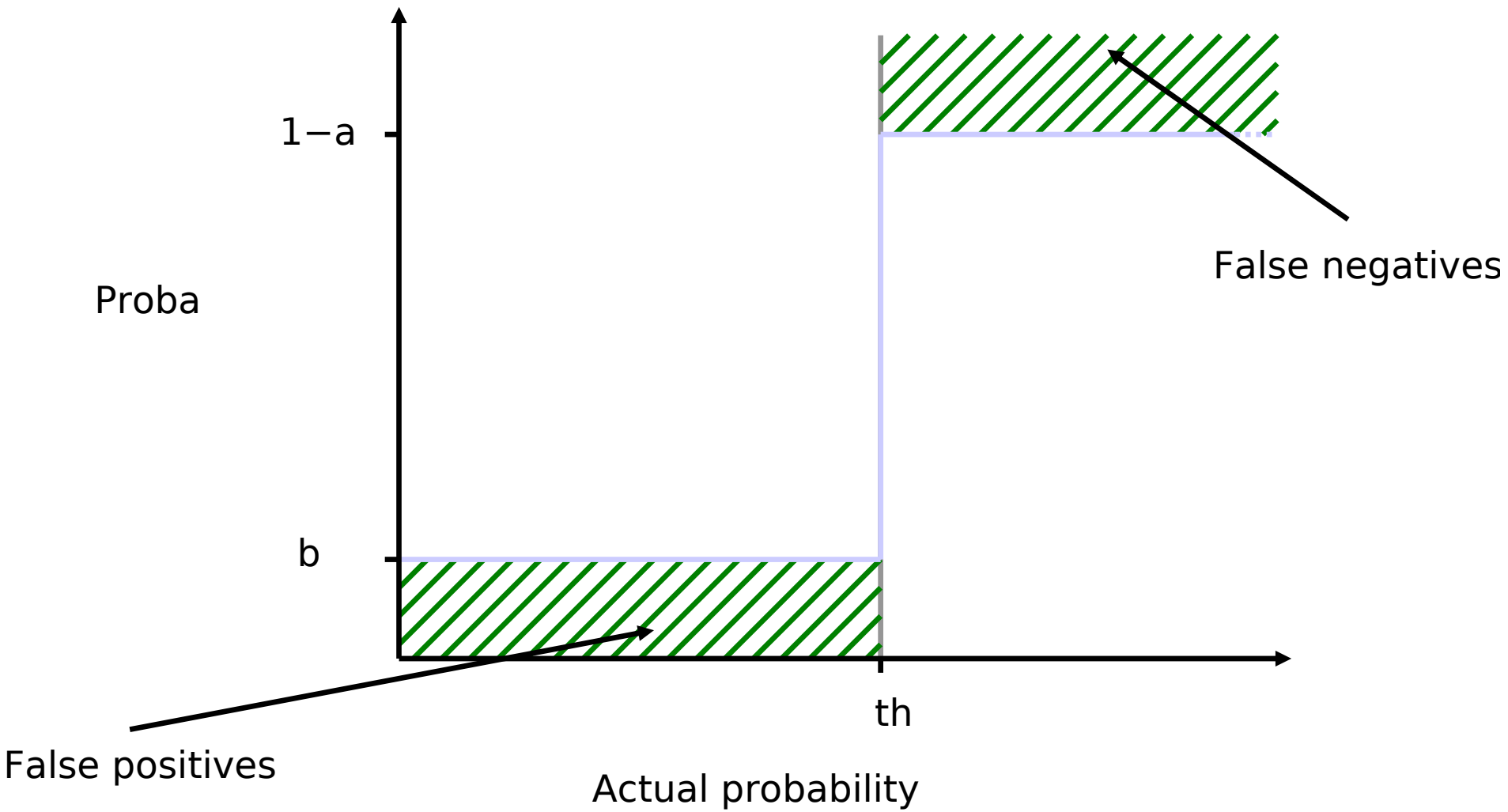
Hypothesis Testing

$$\text{Test } H_1 : p \leq \theta \text{ against } H_0 : p > \theta$$

With (Type error):

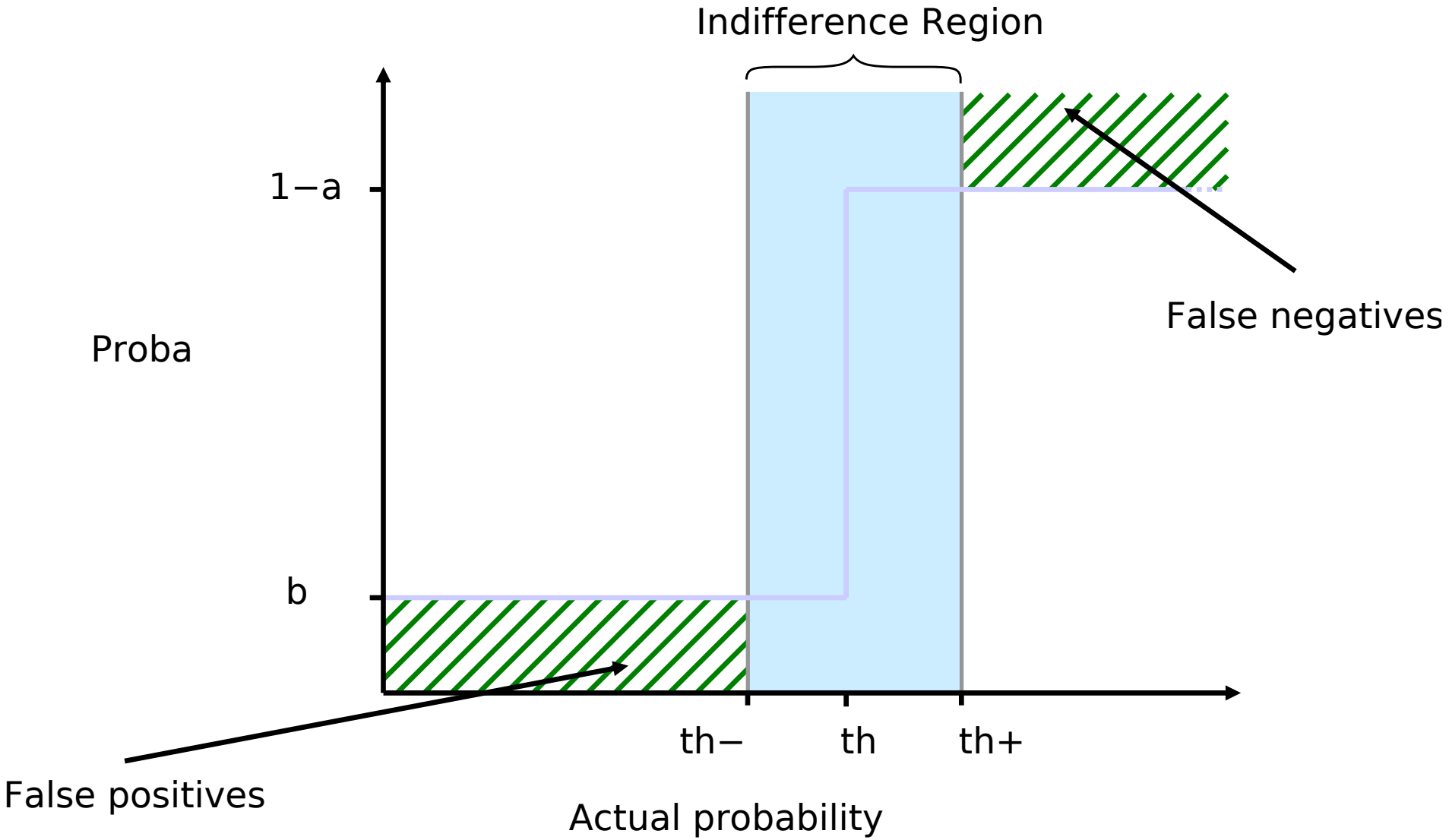
1. α : the probability to accept H_1 while H_0 is true;
2. β : the probability to accept H_0 while H_1 is true.

Performance of Test



Needs an infinite number of samples to get ideal performances !

Performance of Test



If $p \in [\theta - \delta, \theta + \delta]$, we say we are *indifferent* to know if $p \geq \theta$

Summary

We want to test :

$$H_0 : p \geq p_0 \text{ against } H_1 : p < p_1, \text{ where}$$
$$p_0 = \theta + \delta \text{ and } p_1 = \theta - \delta.$$

With:

- Type erros α and β, and
- Indifference region 2δ.

Summary

We want to test :

$$H_0 : p \geq p_0 \text{ against } H_1 : p < p_1, \text{ where}$$
$$p_0 = \theta + \delta \text{ and } p_1 = \theta - \delta.$$

With:

- Type erros α and β, and
- Indifference region 2δ.

Hypothesis Testing

$H_1 : p \leq \theta$, $H_0 : p > \theta$

Compute

$$W = \prod_{i=1}^{m} \frac{Pr(Z_i = z_i \mid p = \theta - \delta)}{Pr(Z_i = z_i \mid p = \theta + \delta)} = \frac{(\theta - \delta)^{d_m}(1 - \theta + \delta)^{m - d_m}}{(\theta + \delta)^{d_m}(1 - \theta - \delta)^{m - d_m}}, \quad (1)$$

where $d_m = \sum_{i=1}^{m} z_i$.

Stop when :

- $W \geq (1 - \beta)/\alpha$: H_1 is accepted;
- $W \leq \beta/(1 - \alpha)$: H_0 is accepted.

Hypothesis Testing

$H_1 : p \leq \theta$, $H_0 : p > \theta$

Compute

$$W = \prod_{i=1}^{m} \frac{Pr(Z_i = z_i \mid p = \theta - \delta)}{Pr(Z_i = z_i \mid p = \theta + \delta)} = \frac{(\theta - \delta)^{d_m}(1 - \theta + \delta)^{m - d_m}}{(\theta + \delta)^{d_m}(1 - \theta - \delta)^{m - d_m}}, \quad (1)$$

where $d_m = \sum_{i=1}^{m} z_i$.

Stop when :

- $W \geq (1 - \beta)/\alpha$: H_1 is accepted;
- $W \leq \beta/(1 - \alpha)$: H_0 is accepted.

Hypothesis Testing

$H_1 : p \leq \theta$, $H_0 : p > \theta$

Compute

$$W = \prod_{i=1}^{m} \frac{Pr(Z_i = z_i \mid p = \theta - \delta)}{Pr(Z_i = z_i \mid p = \theta + \delta)} = \frac{(\theta - \delta)^{d_m}(1 - \theta + \delta)^{m - d_m}}{(\theta + \delta)^{d_m}(1 - \theta - \delta)^{m - d_m}}, \quad (1)$$

where $d_m = \sum_{i=1}^{m} z_i$.

Stop when :

- $W \geq (1 - \beta)/\alpha$: H_1 is accepted;
- $W \leq \beta/(1 - \alpha)$: H_0 is accepted.

Conclusion

- A new approach for the verification of stochastic systems
- Can be more efficient than probabilistic model checking
- Can be more general than probabilistic model checking
- The price to pay is the confidence interval.

Next steps:

- Rare event simulation
- Implementation
- Three applications