A Quick Tour on Statistical Model Checking

Axel Legay

12 novembre 2015

The Rare Event problem in SMC

Rare events (computing very small probabilities) are challenging

- Require a lot of samples (to see the event at least once)
- Relative error explodes

How to overcome the Rare Event problem in SMC?

- **Importance Sampling**: Tackle the problem by reasoning on the model
- **Importance Splitting**: Tackle the problem by reasoning on the property
  We focus on the second one.

The Rare Event problem in SMC

Rare events (computing very small probabilities) are challenging

- Require a lot of samples (to see the event at least once)
- Relative error explodes

How to overcome the Rare Event problem in SMC?

- **Importance Sampling**: Tackle the problem by reasoning on the model
- **Importance Splitting**: Tackle the problem by reasoning on the property
  We focus on the second one.

The Rare Event problem in SMC

Rare events (computing very small probabilities) are challenging

- Require a lot of samples (to see the event at least once)
- Relative error explodes

How to overcome the Rare Event problem in SMC?

- **Importance Sampling**: Tackle the problem by reasoning on the model
- **Importance Splitting**: Tackle the problem by reasoning on the property
  We focus on the second one.

The Rare Event problem in SMC

Rare events (computing very small probabilities) are challenging

- Require a lot of samples (to see the event at least once)
- Relative error explodes

How to overcome the Rare Event problem in SMC?

- **Importance Sampling**: Tackle the problem by reasoning on the model
- **Importance Splitting**: Tackle the problem by reasoning on the property
  We focus on the second one.

The Rare Event problem in SMC

Rare events (computing very small probabilities) are challenging

- Require a lot of samples (to see the event at least once)
- Relative error explodes

How to overcome the Rare Event problem in SMC?

- **Importance Sampling**: Tackle the problem by reasoning on the model
- **Importance Splitting**: Tackle the problem by reasoning on the property
  We focus on the second one.

The Rare Event problem in SMC

Rare events (computing very small probabilities) are challenging

- Require a lot of samples (to see the event at least once)
- Relative error explodes

How to overcome the Rare Event problem in SMC?

- **Importance Sampling**: Tackle the problem by reasoning on the model
- **Importance Splitting**: Tackle the problem by reasoning on the property
  We focus on the second one.

The Rare Event problem in SMC

Rare events (computing very small probabilities) are challenging

- Require a lot of samples (to see the event at least once)
- Relative error explodes

How to overcome the Rare Event problem in SMC?

- **Importance Sampling**: Tackle the problem by reasoning on the model
- **Importance Splitting**: Tackle the problem by reasoning on the property
  We focus on the second one.

Basics of Importance Splitting

Let A be a rare event and $(A_k)_{0 \leq k \leq n}$ be a sequence of nested events:

$$A_0 \supset A_1 \supset ... \supset A_n = A$$

By Bayes formula,

$$\gamma \stackrel{def}{=} P(A) = P(A_0)P(A_1 \mid A_0)P(A_2 \mid A_1)...P(A_n \mid A_{n-1})$$

implying that every conditional probability is less rare:

$$\forall k,\ P(A_k \mid A_{k-1}) = \gamma_k \geq \gamma$$

Importance Splitting principles

- Write γ as a product of γ_k
  - How do you define conditional probabilities?
- Estimate separately each γ_k.
  - How do you estimate in practice these conditional probabilities?
- Importance Splitting estimator:

$$\tilde{\gamma} = \prod_{k=0}^{n} \hat{\gamma}_k$$

  - What about the confidence interval?

Importance Splitting principles

- Write γ as a product of γ_k
  - How do you define conditional probabilities?
- Estimate separately each γ_k.
  - How do you estimate in practice these conditional probabilities?
- Importance Splitting estimator:

$$\tilde{\gamma} = \prod_{k=0}^{n} \hat{\gamma}_k$$

  - What about the confidence interval?

Importance Splitting principles

- Write γ as a product of γ_k
  - How do you define conditional probabilities?
- Estimate separately each γ_k.
  - How do you estimate in practice these conditional probabilities?
- Importance Splitting estimator:

$$\tilde{\gamma} = \prod_{k=0}^{n} \hat{\gamma}_k$$

  - What about the confidence interval?

Importance Splitting principles

- Write γ as a product of γ_k
  - How do you define conditional probabilities?
- Estimate separately each γ_k.
  - How do you estimate in practice these conditional probabilities?
- Importance Splitting estimator:

$$\tilde{\gamma} = \prod_{k=0}^{n} \hat{\gamma}_k$$

  - What about the confidence interval?

Importance Splitting principles

- Write γ as a product of γ_k
  - How do you define conditional probabilities?
- Estimate separately each γ_k.
  - How do you estimate in practice these conditional probabilities?
- Importance Splitting estimator:

$$\tilde{\gamma} = \prod_{k=0}^{n} \hat{\gamma}_k$$

  - What about the confidence interval?

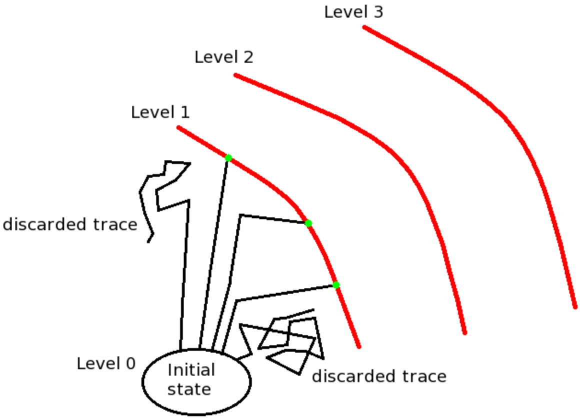
Importance Splitting principles

- Write γ as a product of γ_k
  - How do you define conditional probabilities?
- Estimate separately each γ_k.
  - How do you estimate in practice these conditional probabilities?
- Importance Splitting estimator:

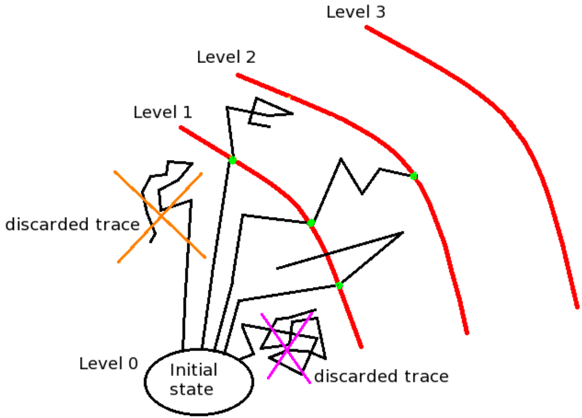
$$\tilde{\gamma} = \prod_{k=0}^{n} \hat{\gamma}_k$$

  - What about the confidence interval?

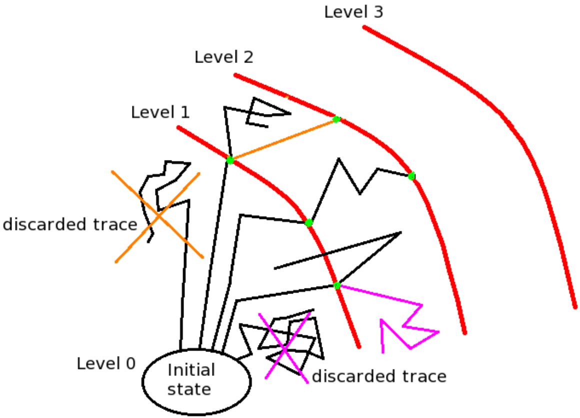
Example: Reaching Level 3 within T



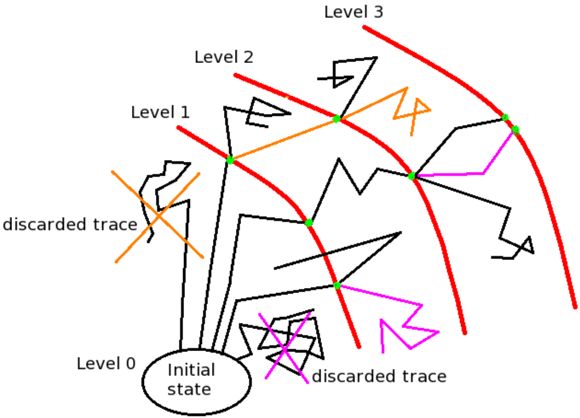
Example: Reaching Level 3 within T



Example: Reaching Level 3 within T



Example: Reaching Level 3 within T



P(reaching Level 3)=3/5*2/5*2/5

Confidence Interval based on relative error

- $(1-\alpha)$ CI based on relative variance σ^2:

$$\left[\tilde{\gamma}\left(\frac{1}{1+\frac{z_\alpha \sigma}{\sqrt{N}}}\right); \tilde{\gamma}\left(\frac{1}{1-\frac{z_\alpha \sigma}{\sqrt{N}}}\right)\right] \text{ with } \sigma^2 \geq \sum_{k=1}^{n} \frac{1-\gamma_k}{\gamma_k}$$

- σ^2 is minimize when all the γ_k have same probability.

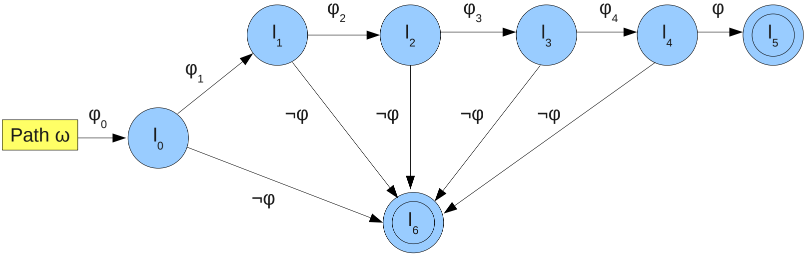
Importance Splitting in a Model Checking Context

Idea: given a rare property ϕ, define a set of levels based on a sequence of temporal properties such that:

$$(\phi_k)_{0 \leq k \leq n} \; : \; \phi_0 \Leftarrow \phi_1 \Leftarrow \ldots \Leftarrow \phi_n = \phi$$

Thus,

$$\gamma = P(\omega \models \phi_0) \prod_{k=1}^{n} P(\omega \models \phi_k \mid \omega \models \phi_{k-1})$$



Formula decomposition

Simple decomposition:

$$\phi = \bigwedge_{j=1}^{n} \psi_j \quad \longrightarrow \quad \forall i \in \{1, \ldots, n\}, \quad \phi_i = \bigwedge_{j=1}^{i} \psi_j$$

Natural decomposition. Given x a state variable,

$$\phi = (x \geq \tau) \quad \longrightarrow \quad \forall \tau_0 \leq \cdots \leq \tau_n = \tau, \quad \phi_i = (x \geq \tau_i)$$

Temporal decomposition. Make use of propositions:

- $(\phi_n \Rightarrow \phi_{n-1}) \Longrightarrow (\triangle\phi_n \Rightarrow \triangle\phi_{n-1})$ with $\triangle \in \{\Diamond^{\leq t}, \Box^{\leq t}, \bigcirc, \Diamond^{\leq t}\Box^{\leq s}\}$
- $(\phi_n \Rightarrow \phi_{n-1} \wedge \psi_m \Rightarrow \psi_{m-1}) \Longrightarrow (\phi_n \, U \, \psi_m \Rightarrow \phi_{n-1} \, U \, \psi_{m-1})$

Time decomposition

$$\phi = (\Box^{\leq t}\psi) \quad \longrightarrow \quad \forall t_0 \leq \cdots \leq t_n = t, \quad \phi_i = (\Box^{\leq t_i}\psi)$$

Formula decomposition

Simple decomposition:

$$\phi = \bigwedge_{j=1}^{n} \psi_j \quad \longrightarrow \quad \forall i \in \{1, \ldots, n\}, \quad \phi_i = \bigwedge_{j=1}^{i} \psi_j$$

Natural decomposition. Given x a state variable,

$$\phi = (x \geq \tau) \quad \longrightarrow \quad \forall \tau_0 \leq \cdots \leq \tau_n = \tau, \quad \phi_i = (x \geq \tau_i)$$

Temporal decomposition. Make use of propositions:

- $(\phi_n \Rightarrow \phi_{n-1}) \Longrightarrow (\triangle\phi_n \Rightarrow \triangle\phi_{n-1})$ with $\triangle \in \{\Diamond^{\leq t}, \Box^{\leq t}, \bigcirc, \Diamond^{\leq t}\Box^{\leq s}\}$
- $(\phi_n \Rightarrow \phi_{n-1} \wedge \psi_m \Rightarrow \psi_{m-1}) \Longrightarrow (\phi_n \, U \, \psi_m \Rightarrow \phi_{n-1} \, U \, \psi_{m-1})$

Time decomposition

$$\phi = (\Box^{\leq t}\psi) \quad \longrightarrow \quad \forall t_0 \leq \cdots \leq t_n = t, \quad \phi_i = (\Box^{\leq t_i}\psi)$$

Formula decomposition

Simple decomposition:

$$\phi = \bigwedge_{j=1}^{n} \psi_j \quad \longrightarrow \quad \forall i \in \{1, \ldots, n\}, \quad \phi_i = \bigwedge_{j=1}^{i} \psi_j$$

Natural decomposition. Given x a state variable,

$$\phi = (x \geq \tau) \quad \longrightarrow \quad \forall \tau_0 \leq \cdots \leq \tau_n = \tau, \quad \phi_i = (x \geq \tau_i)$$

Temporal decomposition. Make use of propositions:

- $(\phi_n \Rightarrow \phi_{n-1}) \Longrightarrow (\triangle\phi_n \Rightarrow \triangle\phi_{n-1})$ with $\triangle \in \{\Diamond^{\leq t}, \Box^{\leq t}, \bigcirc, \Diamond^{\leq t}\Box^{\leq s}\}$
- $(\phi_n \Rightarrow \phi_{n-1} \wedge \psi_m \Rightarrow \psi_{m-1}) \Longrightarrow (\phi_n \, U \, \psi_m \Rightarrow \phi_{n-1} \, U \, \psi_{m-1})$

Time decomposition

$$\phi = (\Box^{\leq t}\psi) \quad \longrightarrow \quad \forall t_0 \leq \cdots \leq t_n = t, \quad \phi_i = (\Box^{\leq t_i}\psi)$$

Formula decomposition

Simple decomposition:

$$\phi = \bigwedge_{j=1}^{n} \psi_j \quad \longrightarrow \quad \forall i \in \{1, \ldots, n\}, \quad \phi_i = \bigwedge_{j=1}^{i} \psi_j$$

Natural decomposition. Given x a state variable,

$$\phi = (x \geq \tau) \quad \longrightarrow \quad \forall \tau_0 \leq \cdots \leq \tau_n = \tau, \quad \phi_i = (x \geq \tau_i)$$

Temporal decomposition. Make use of propositions:

- $(\phi_n \Rightarrow \phi_{n-1}) \Longrightarrow (\triangle\phi_n \Rightarrow \triangle\phi_{n-1})$ with $\triangle \in \{\Diamond^{\leq t}, \Box^{\leq t}, \bigcirc, \Diamond^{\leq t}\Box^{\leq s}\}$
- $(\phi_n \Rightarrow \phi_{n-1} \wedge \psi_m \Rightarrow \psi_{m-1}) \Longrightarrow (\phi_n \, U \, \psi_m \Rightarrow \phi_{n-1} \, U \, \psi_{m-1})$

Time decomposition

$$\phi = (\Box^{\leq t}\psi) \quad \longrightarrow \quad \forall t_0 \leq \cdots \leq t_n = t, \quad \phi_i = (\Box^{\leq t_i}\psi)$$

Dining Philosophers Problem

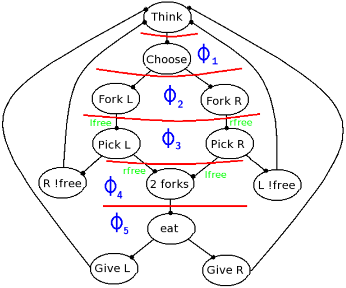


Figure : Automata modelling a philosopher

- property of interest:
  $\phi = \phi_5 = \mathbf{F}^{30}$ (Phil i eat)
- $\phi_4 = \mathbf{F}^{30}$ (Phil i picks 2 forks)
- $\phi_3 = \mathbf{F}^{30}$ (Phil i picks 1 fork)
- $\phi_2 = \mathbf{F}^{30}$ (Phil i intends to take a fork)
- $\phi_1 = \mathbf{F}^{30}$ (Phil i chooses)
- $\phi_0 = \mathbf{F}^{30}$ (Phil i thinks)
- $\phi_5 \Rightarrow \phi_4 \Rightarrow \cdots \Rightarrow \phi_0$

Dining Philosophers Problem

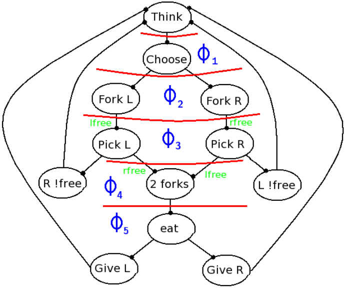
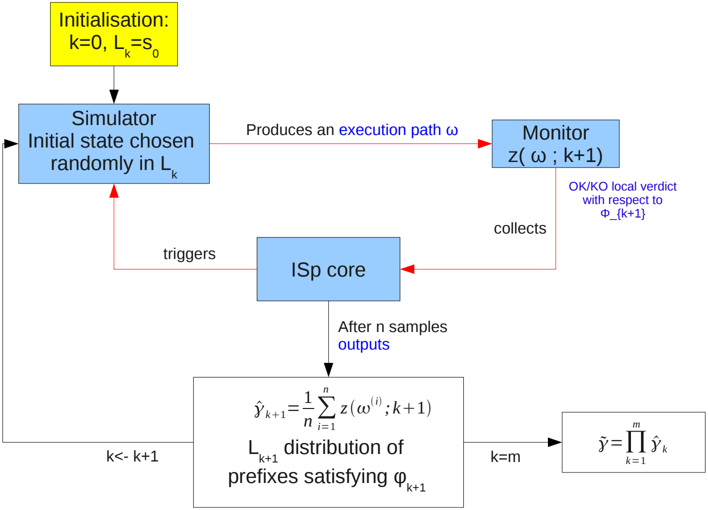


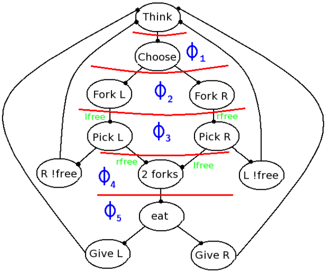
Figure : Automata modelling a philosopher

- property of interest: $\phi = \phi_5 = \mathbf{F}^{30}$ (Phil i eat)
- $\phi_4 = \mathbf{F}^{30}$ (Phil i picks 2 forks)
- $\phi_3 = \mathbf{F}^{30}$ (Phil i picks 1 fork)
- $\phi_2 = \mathbf{F}^{30}$ (Phil i intends to take a fork)
- $\phi_1 = \mathbf{F}^{30}$ (Phil i chooses)
- $\phi_0 = \mathbf{F}^{30}$ (Phil i thinks)
- $\phi_5 \Rightarrow \phi_4 \Rightarrow \cdots \Rightarrow \phi_0$

Importance Splitting in a Model Checking Context



A naive decomposition



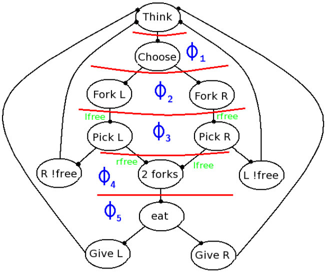
- 150 philosophers
- property of interest:
  $\phi = \phi_5 = \mathbf{F}^{30}$ (Phil i eat)
- $\gamma \approx 1.59 \times 10^{-6}$

Results:

- Time (with 1000 paths per iteration): 6.95 seconds in average
- $\tilde{\gamma}_{1:5} \in \{0.158, 0.088, 0.027, 0.008, 0.003\}$
- $\tilde{\gamma} = 10^{-8}$

=> Need to increase the decomposition to make use of Cérou-Guyader's Adaptive Important Splitting algorithms

A naive decomposition



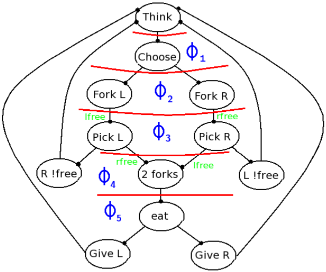
- 150 philosophers
- property of interest:
  $\phi = \phi_5 = \mathbf{F}^{30}$ (Phil i eat)
- $\gamma \approx 1.59 \times 10^{-6}$

Results:

- Time (with 1000 paths per iteration): 6.95 seconds in average
- $\tilde{\gamma}_{1:5} \in \{0.158, 0.088, 0.027, 0.008, 0.003\}$
- $\tilde{\gamma} = 10^{-8}$

=> Need to increase the decomposition to make use of Cérou-Guyader's Adaptive Important Splitting algorithms

A naive decomposition



- 150 philosophers
- property of interest:
  $\phi = \phi_5 = \mathbf{F}^{30}$ (Phil i eat)
- $\gamma \approx 1.59 \times 10^{-6}$

Results:

- Time (with 1000 paths per iteration): 6.95 seconds in average
- $\tilde{\gamma}_{1:5} \in \{0.158, 0.088, 0.027, 0.008, 0.003\}$
- $\tilde{\gamma} = 10^{-8}$

=> Need to increase the decomposition to make use of Cérou-Guyader's Adaptive Important Splitting algorithms

Idealized Version (Cérou-Guyader)

- Relative variance of the estimator: $\sigma^2 = \sum_{k=1}^{n} \frac{1-\gamma_k}{\gamma_k}$
- For a fixed number of levels, minimal variance if all the conditional probabilities are equal ($= \gamma_0$).
- Variance minimised when γ_0 is close to 1.
- Problem: levels might be too coarse.

Idealized Version (Cérou-Guyader)

- Relative variance of the estimator: $\sigma^2 = \sum_{k=1}^{n} \frac{1-\gamma_k}{\gamma_k}$
- For a fixed number of levels, minimal variance if all the conditional probabilities are equal ($= \gamma_0$).
- Variance minimised when γ_0 is close to 1.
- Problem: levels might be too coarse.

Idealized Version (Cérou-Guyader)

- Relative variance of the estimator: $\sigma^2 = \sum_{k=1}^{n} \frac{1-\gamma_k}{\gamma_k}$
- For a fixed number of levels, minimal variance if all the conditional probabilities are equal ($= \gamma_0$).
- Variance minimised when γ_0 is close to 1.
- Problem: levels might be too coarse.

Idealized Version (Cérou-Guyader)

- Relative variance of the estimator: $\sigma^2 = \sum_{k=1}^{n} \frac{1-\gamma_k}{\gamma_k}$
- For a fixed number of levels, minimal variance if all the conditional probabilities are equal ($= \gamma_0$).
- Variance minimised when γ_0 is close to 1.
- Problem: levels might be too coarse.

Use of heuristics

- Assign a finer score $S(\omega)$ to path ω
- How to increase the granularity of the score function S?

- Time-bounded reachability problem:

$$S(\omega) = k^* - \epsilon(t_{k^*})$$

with:
  - $k^* = \max_k \{k \mid \omega \models \phi_k\}$
  - $t_{k^*} = \min_t \{t \in [0, T] \mid \omega \models \phi_{k^*}\}$
  - $\epsilon(\cdot) \in]0; 1[$ an increasing time function.

Use of heuristics

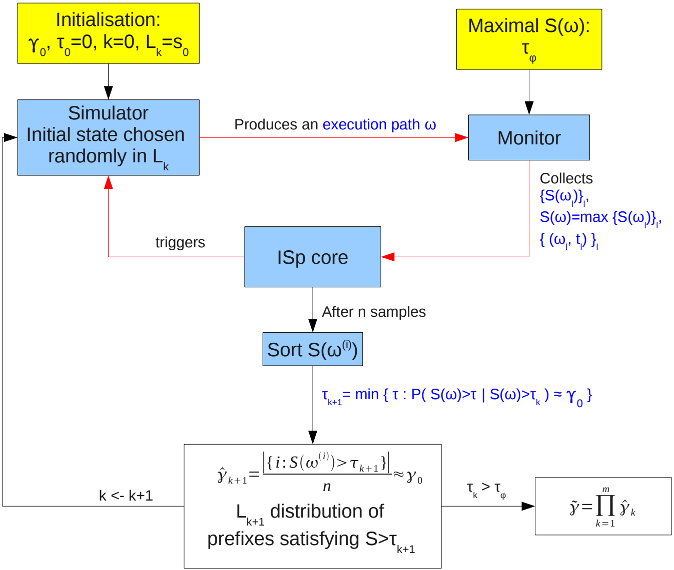
- Assign a finer score $S(\omega)$ to path ω
- How to increase the granularity of the score function S?
- Time-bounded reachability problem:

$$S(\omega) = k^* - \epsilon(t_{k^*})$$

with:

- $k^* = \max_k\{k \mid \omega \models \phi_k\}$
- $t_{k^*} = \min_t\{t \in [0, T] \mid \omega \models \phi_{k^*}\}$
- $\epsilon(\cdot) \in]0; 1[$ an increasing time function.

Adaptive important sampling implementation



Experimental Results given by an optimised algorithm

Stat.	MC	Importance splitting			
nb exp	1	100	100	100	100
nb path	10^7	100	200	500	1000
$\bar{t}$ in sec.	$>$ 5 h	1.73	4.08	11.64	23.77
$\bar{\tilde{\gamma}}$	1.5	1.52	1.59	1.58	1.65
$\sigma(\tilde{\gamma})$	0.39	1.02	0.87	0.5	0.38
95%-CI	[0.74; 2.26]	[1.34; 1.74]	[1.48; 1.72]	[1.54; 1.63]	[1.63; 1.67]

95%-CI based on a 3×10^8 sample: $\left[1.44 \times 10^{-6}; 1.72 \times 10^{-6}\right]$