Probabilistic verification and approximation schemes

Richard Lassaigne
Equipe de Logique mathématique,
CNRS-Université Paris 7

Joint work with Sylvain Peyronnet
(LRDE/EPITA & Equipe de Logique)

Motivation

Probabilistic verification

Randomised approximation schemes

Approximate Probabilistic Model Checker

Conclusion

Efficient approximations of logical satisfiability :

$$\text{Model } \mathcal{M} \models_\varepsilon \text{ Property}$$

1st case (difficult) : Model = **Automaton** $\mathcal{A}$

Property = a word $w \in_\varepsilon \mathcal{L}(\mathcal{A})$

Property testing

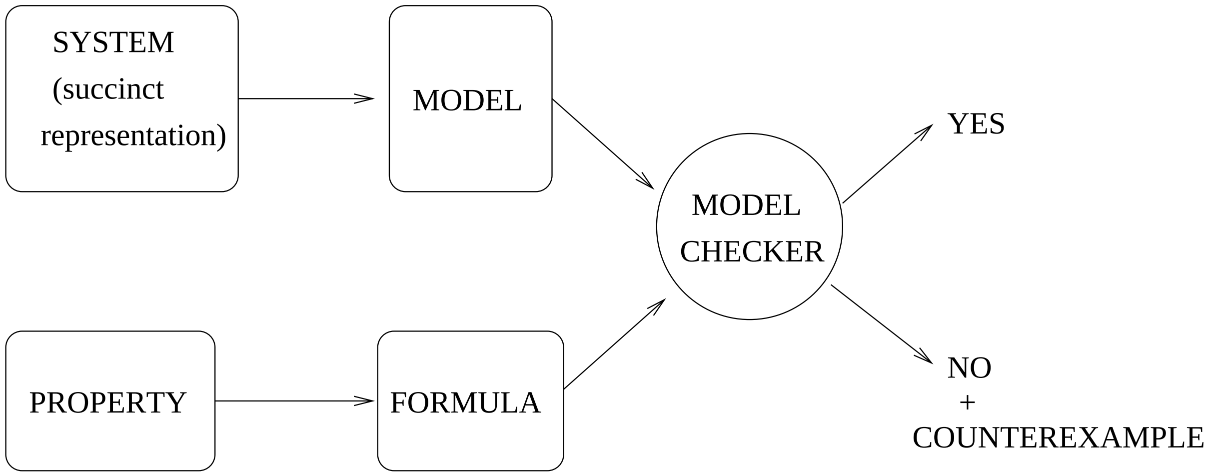
(E. Fisher, F. Magniez and M. de Rougemont)

2st case (easy) : Model = **Probabilistic transition system**

Property = the **probability measure** of ... **equals**$_\varepsilon$ **p**

Randomised Approximation Scheme

(this talk)



Input :

- Model $\mathcal{M} = (S, R)$ $R \subseteq S^2$ (transition relation)
- Initial state s_0
- Formula φ

Output :

- YES if $(\mathcal{M}, s_0) \models \varphi$
- NO with a counterexample if $(\mathcal{M}, s_0) \not\models \varphi$

Complexity

$O(|M|.|\varphi|)$ (Branching Time Temporal Logic **CTL**)

or

$O(|M|.2^{|\varphi|})$ (Linear Time Temporal Logic **LTL**)

Problem :

State space explosion phenomenon

(the problem is not the time but the space)

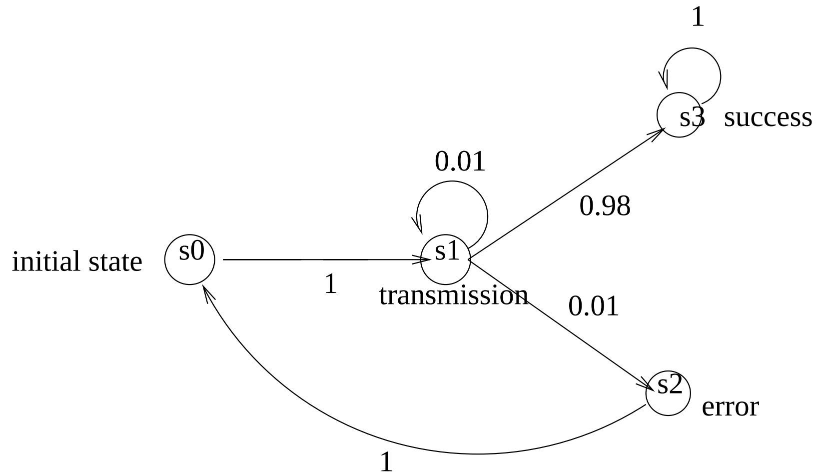
Classical methods :

- Symbolic representation (OBDD)
- SAT-based methods (Bounded moded checking)
- Abstraction

Probabilistic Transition Systems

Input :

- Model $\mathcal{M} = (S, \pi, L)$ and initial state s_0
- $\pi : S^2 \longrightarrow [0,1]$ Probability function
- $L : S \longrightarrow 2^{AP}$ (state labelling)
- Formula ψ (**LTL**)



Output : $Prob_\Omega[\psi]$

where (for example) $\psi \equiv transmission\,\mathbf{U}\,success$

(Ω probabilistic space of execution paths starting at s_0)

Probability space (and measure) :

Finite paths $\rho = (s_0, s_1, \ldots, s_n)$:

$Prob(\{\sigma/\sigma$ is a path and $(s_0, s_1, \ldots, s_n)$ is a prefix of $\sigma\}) =$

$$\prod_{i=1}^{n} P(s_{i-1}, s_i)$$

Measure extended to the Borel family of sets generated by the sets $\{\sigma/\rho$ is a prefix of $\sigma\}$ where ρ is a finite path.

The set of paths $\{\sigma/\sigma(0) = s$ and $\mathcal{M}, \sigma \models \psi\}$ is measurable (Vardi).

Complexity : (Coucourbetis and Yannakakis) [CY95]

Qualitative verification (i.e. prob=1 ?)

Same **complexity** as **LTL model checking**

$$O(|M|.2^{|\psi|})$$

Quantitative verification (i.e. prob= ?)

$$O(|M|^3.2^{|\psi|})$$

Method : Computing $Prob_\Omega[\psi]$

- Transforming step by step the formula and the Markov chain $\mathcal{M}$
- Eliminating one by one the temporal connectives
- Preserving the satisfaction probability
- Solving system of linear equations of size $|M|$.

Counting problems : (L. Valiant 79)

- $\sharp P$ class of counting problems associated with NP decision problems
- $\sharp SAT$ is a $\sharp P$-complete problem

Randomised Approximation Scheme :

(R. Karp and M. Luby 85)

Randomised Algorithm A

- Input : instance x of a counting problem, $\varepsilon, \delta > 0$
- Output : value $A(x, \varepsilon, \delta)$ such that

$$Pr[(1-\varepsilon)\#(x) \leq A(x,\varepsilon,\delta) \leq (1+\varepsilon)\#(x)] \geq 1-\delta$$

Fully Polynomial Randomised Approximation Scheme (FPRAS) :

Running time is $poly(|x|, (1/\varepsilon), \log(1/\delta))$

Classical Randomised Approximation Schemes :

- Approximation of $\sharp DNF$ (Karp, Luby, Madras 89)

  Input : Disjunctive Normal Form formula Φ
  Output : number of assignments satisfying Φ

- Approximation of graph reliability (Karger 99)

  Input : a graph whose edges can disappear with some probability
  Output : the probability that the graph remains connected

Can we efficiently approximate $Prob_{\Omega}(\psi)$?

General case : (R. Lassaigne and S. Peyronnet 05)

There is **no** probabilistic approximation algorithm with polynomial time complexity for computing $Prob_{\Omega}(\psi)$ ($\psi \in LTL$) unless $BPP = NP$.

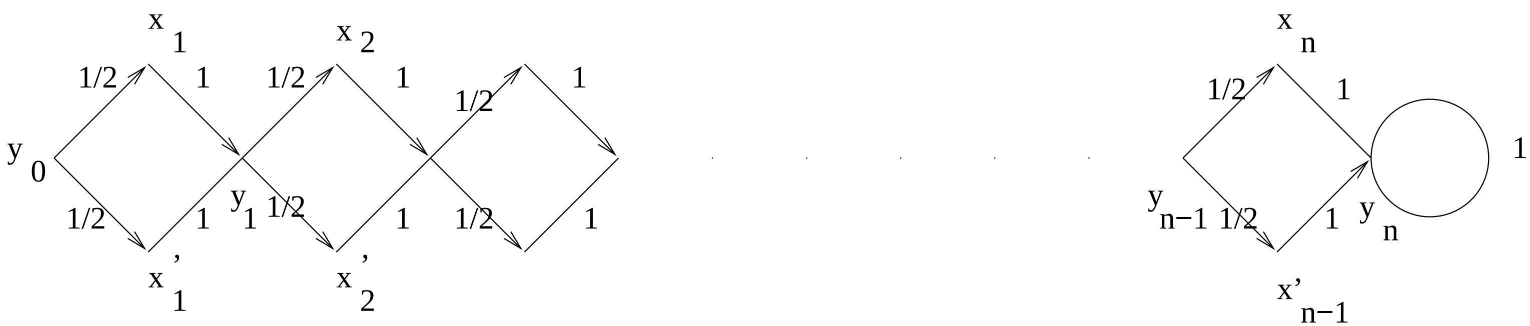
BPP **:** Complexity class of problems decidable by a Monte-Carlo randomized algorithm (with two-sided error).

Bounded-error, **P**robabilistic, **P**olynomial time : class of languages L s.t.

$$x \in L \ : \ Prob[\text{ acceptance of } x] \geq 3/4$$

$$x \notin L \ : \ Prob[\text{ acceptance of } x] \leq 1/4$$

$\#SAT$ can be reduced to counting the number of paths of length $2n$, whose infinite extensions satisfy ψ.



Propositional clauses : $c_1, \ldots, c_m$

Labelling of states :

$L(x_i) = \{c_j \ / \ x_i$ appears in $c_j\}$ $(i = 1, \ldots, n)$

$L(x'_i) = \{c_j \ / \ \neg x_i$ appears in $c_j\}$ $(i = 1, \ldots, n)$

LTL formula ψ : $\bigwedge_{i=1}^{n} F c_j$

Sketch of the proof

• Counting this number of paths gives $Prob_\Omega(\psi)$

• If there was a FPRAS for computing $Prob_\Omega(\psi)$, then we could randomly approximate $\#SAT$

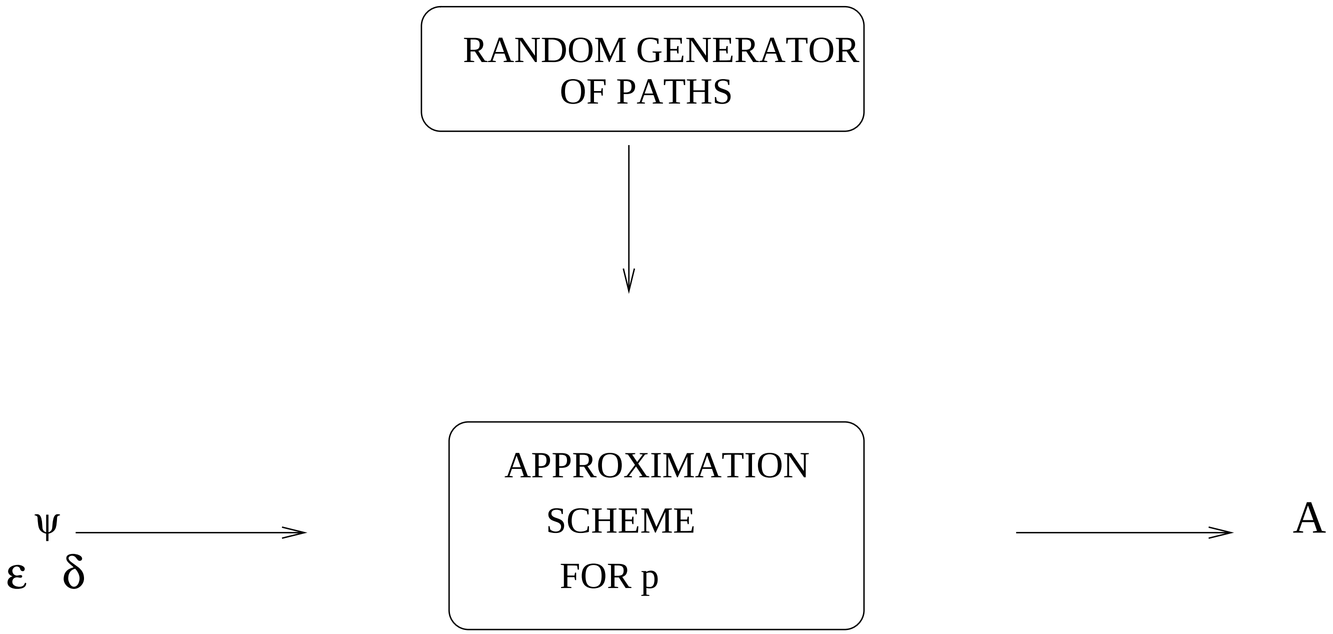
• A FPRAS allows to distinguish, in polynomial time, for input x, between the case $\#(x) = 0$ and the case $\#(x) > 0$

• Then we would have a polynomial time randomised algorithm to decide SAT and $BPP = NP$

Moreover

• There is no deterministic polynomial time approximation algorithm neither for $\#SAT$ nor for computing $Prob_\Omega(\psi)$ (Jerrum and Sinclair :$\#P$-complete problems either admit a FPRAS or are not approximable at all)

We want to approximate a probability p.



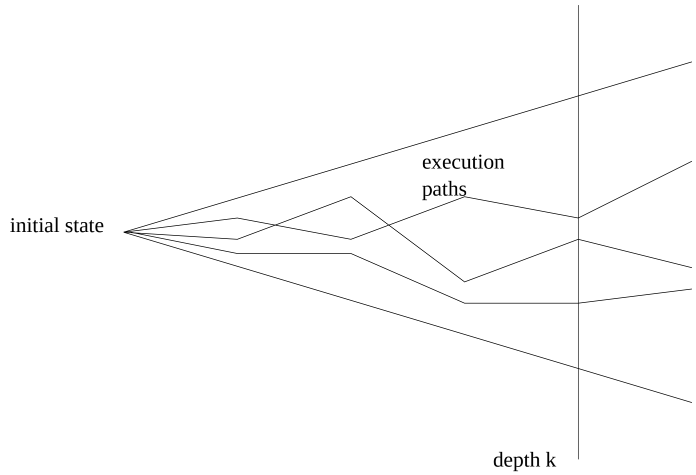
$$Pr[(p - \varepsilon) \leq A \leq (p + \varepsilon)] \geq 1 - \delta$$

ε : *error parameter (additive approximation)*

δ : *confidence parameter (randomised algorithm)*

We consider $Prob_k(\phi)$ with :

- the probability space is the space over paths of length $\leq k$



- ψ express a monotone property

$$\lim_{k \to \infty} Prob_k(\phi) = Prob_\Omega(\phi)$$

Generic approximation algorithm $\mathcal{GAA}$

input : $\phi, diagram, \varepsilon, \delta$

Let $A := 0$

Let $N := \log(\frac{2}{\delta})/2\varepsilon^2$

For i from 1 to N do

1. Generate a random path σ of depth k
2. If ϕ is true on σ then $A := A + 1$

Return (A/N)

Algorithm based on Monte-Carlo estimation and Chernoff-Hoeffding bound

Diagram : succinct representation of the system (for example in Reactive Modules)

Method : Estimation (Monte-Carlo) + Chernoff-Hoeffding bound

X Bernoulli $(0,1)$ random variable with success probability p

- Do N independent Bernoulli trials $X_1, X_2, \ldots, X_N$
- Estimate p by $\mu = \sum_{i=1}^{N} X_i/N$ with error ε
- Sample size N is such that the error probability $< \delta$

Chernoff-Hoeffding bound :

$$Pr[\mu < p - \varepsilon] + Pr[\mu > p + \varepsilon] < 2e^{-2N\varepsilon^2}$$

If $N \geq \ln(\frac{2}{\delta})/2\varepsilon^2$, then

$$Pr[p - \varepsilon \leq \mu \leq p + \varepsilon)] \geq 1 - \delta$$

Theorem :

$\mathcal{GAA}$ is a FPRAS for $Prob_k(\psi)$

Methodology : To approximate $Prob_\Omega[\psi]$

- Choose $k \approx log|M| \cdot ln(1/\varepsilon)$
- Iterate approximation of $Prob_k[\psi]$

Remark :

- Length of needed paths can be the diameter of the system
- Convergence time may be long, but space is saved...

Improvement :

Optimal Approximation Algorithm (Dagum, Karp, Luby and Madras) with multiplicative error.

Improvement : Randomised approximation scheme with multiplicative error

Idea : Use the optimal approximation algorithm ($\mathcal{OAA}$) [DKLR00]

- The first step outputs an (ε, δ) -approximation $\hat{p}$ of p after expected number of experiments proportional to Γ/p where $\Gamma = 4(e-2).\ln(\frac{2}{\delta})/\varepsilon^2$
- The second step uses the value of $\hat{p}$ to produce an estimate $\hat{\rho}$ de $\rho = max(\sigma^2, \varepsilon p)$ (σ^2 is the variance)
- The third step uses the values of $\hat{p}$ and $\hat{\rho}$ to set the number of experiments and runs the experiments to produce an (ε, δ)-approximation of p

Remark : It's not a **FPRAS**

Continuous Time Markov Chains

$\mathcal{M} = (S, R)$

S is the set of states

$R : S^2 :\longrightarrow \mathbb{R}_+$ Rate matrix

$s \in S, \lambda(s) = \sum_{s' \in S} R(s, s')$ Total rate of transition from s

Delay of transition from s to s' governed by an exponential distribution with rate $R(s, s')$.

Probability to move from s to s' within t time units :

$$P(s, s') = \frac{R(s, s')}{\lambda(s)}(1 - e^{-\lambda(s) \times t})$$

Random generator of paths

Execution path : $s_0(t_0) \rightarrow s_1(t_1) \rightarrow \ldots s_i(t_i) \ldots$

$t := 0$

Initialize at state s

Repeat

$i := s$

Choose state j with proba $P(i,j) = R(i,j)/\lambda(i)$

$s := j$

$t := t - \ln(random_{[0,1])})/R(i,j)$

Until $t \geq T$

Simulation by inversion of uniform distribution over $[0,1]$

APMC : Approximate Probabilistic Model Checker

- Freely available GPL software
- Developped at LRDE/EPITA, Paris VII and Paris XI (T. Hérault)
- Use **randomised approximation algorithm**
- **Distributed** computation
- Integrated in the probabilistic model checker **PRISM**
- Case studies : CSMA/CD, 2PCP, Sensor Networks...

The dining philosophers problem

$$Prob[\bigvee_{i=1}^{n} hungry(i) \implies \mathrm{F}(\bigvee_{i=1}^{n} eat(i))] \geq 1 - \varepsilon$$

# phil.	depth	time	PRISM (time)	PRISM (states)
5	23	24.67	0.615	64858
10	33	70.32	13.059	4.21×10^9
15	42	146.22	68.926	2.73×10^{14}
20	51	261.43	167.201	1.77×10^{19}
25	58	412.06	3237,143	1.14×10^{24}
30	66	614.49	out of mem.	-
50	95	2020.79	out of mem.	-
100	148	11475.28	out of mem	-

- Memory used : 2 MB
- k determined experimentally

The dining philosophers problem strike back

Cluster of 20 Athlon XP1800+ sous Linux

# phil.	depth	APMC (time : sec.)	(memory : kbytes)
15	38	11	324
25	55	25	340
50	130	104	388
100	145	418	484
200	230	1399	676
300	295	4071	1012

Conclusion

- Efficiency of randomised approximation schemes (exponential reduction of space complexity)
- Quantitative verification of monotone (reachability) and anti-monotone (safety) properties
- Extension to an approximation with multiplicative error (*optimal approximation algorithm*)
- Continuous time Markov chains CSL (*Continuous Stochastic Logic*)

Ongoing work

- Continuous time Markov chains : APMC 3.0
- New case studies :

  CSMA/CA with cheater

  WLAN sensor networks

  Biological processes
- Practical verification of C programs
- Black Box verification (via learning)

- [CY95] C. Courcoubetis and M. Yannakakis. *The complexity of probabilistic verification*. Journal of the ACM, 24(4), 857-907, 1995.
- [DKLR00] P. Dagum, R. Karp, M. Luby, and S. Ross. *An optimal algorithm for Monte-Carlo estimation*. SIAM Journal on Computing, 29(5), 1484-1496, 2000.
- [HLMP04] T. Hérault, R. Lassaigne, F. Magniette and S. Peyronnet. *Approximate Probabilistic Model Checking*. Int. Conf. on Verification, Model Checking and Abstraction, LNCS n° 2937.
- [KL83] R. Karp and M. Luby. *Monte-Carlo Algorithms for Enumeration and Reliability Problems.*, 24th IEEE FOCS, 56-64, 1983.
- [KLM89] R. Karp, M. Luby and N. Madras. *Monte-Carlo Approximation Algorithms for Enumeration Problems.* Journal of Algorithms 10, 429-448, 1989.

- [KNP02] M. Kwiatkowska, G. Norman and D. Parker. *Probabilistic symbolic model checking with PRISM : A hybrid approach.* Proc. of 8th Int. Conf. TACAS, LNCS n° 2280, p.52-66, 2002.

- [LP05] R. Lassaigne et S. Peyronnet. *Probabilistic Verification and Approximation*. WoLLIC 2005. Electronic Notes in Theoretical Computer Science, vol. 143, p. 101-114.

- [LR03] R. Lassaigne et M. de Rougemont : *Logic and Complexity.* Springer-Verlag, 350 p. (nov. 2003).

- [Val79] L.G. Valiant *The complexity of enumeration and reliability problems.* SIAM Journal on Computing, 8, 410-421, 1979.

- [Var85] *Automatic verification of probabilistic concurrent finite-state programs* 26th IEEE FOCS, 327-338, 1985.