

Vérification Probabiliste et Approximation

Richard Lassaigne

Logique mathématique,
CNRS-Université Paris 7

ACI Sécurité Informatique/Projet VERA

[http ://www.lri.fr/~mdr/vera](http://www.lri.fr/~mdr/vera)

Contexte : Approximation de la vérification

Test de propriétés et d'équivalence

Frédéric Magniez, Michel de Rougemont (LRI, Orsay)

Complexité de la vérification probabiliste Schémas probabilistes d'approximation

Richard Lassaigne, Sylvain Peyronnet (Paris 7, LIX)

Outil de vérification approchée : APMC

Conclusion

Approximation de la vérification

Motivation :

Complexité de la vérification par model checking

Approximation sur les données/Approximation sur le calcul

Objectifs :

- Approximation de la satisfaction d'une propriété logique
- Algorithmes d'approximation efficaces et robustes

Méthodes :

- Test de propriétés (Approximation sur les données)
- Schémas probabilistes d'approximation
(Approximation sur le calcul).

Approximations efficaces de la satisfaction logique :

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

Approximation sur les données : Modèle = **Automate** \mathcal{A}

Propriété = un mot $w \in_{\varepsilon} \mathcal{L}(\mathcal{A})$

Test de propriétés

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

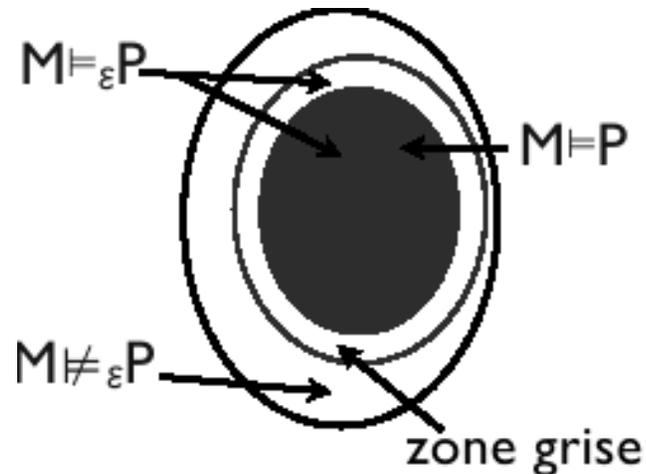
Approximation sur le calcul : Modèle = **Système de transition probabiliste**

Propriété = la **mesure de probabilité** de ... égale $_{\varepsilon}$ p

Schémas probabilistes d'approximation

(R. Lassaigne, S. Peyronnet)

- Satisfaction classique : $\mathcal{M} \models \mathbf{P}$
- Décider si \mathcal{M} satisfait la propriété \mathbf{P}



- Satisfaction **approchée** : $\mathcal{M} \models_\epsilon \mathbf{P}$
 $\mathcal{M} \models_\epsilon \mathbf{P}$ si \mathcal{M} est ϵ -proche de \mathcal{M}' tel que $\mathcal{M}' \models \mathbf{P}$

Testeur : Algorithme probabiliste A

- si $\mathcal{M} \models \mathbf{P}$, alors A accepte
- si \mathcal{M} est ϵ -loin de \mathbf{P} , alors A rejette avec grande probabilité
 Le temps de calcul est indépendant de $|\mathcal{M}|$
 (mais dépend de $1/\epsilon$)

Distance d'édition avec déplacements

- Distance d'édition avec déplacements : $d(w, w')$
 - Insertion, suppression, modification
 - Déplacement de blocs

$w = 011100011110011001$

$w' = 011101111000011000$

$$d(w, w') = 2$$

Si L est un langage, $d(w, L) = \min_{w' \in L} \{d(w, w')\}$

- Calcul exact : NP-difficile
- Calcul approché : $\log|w|$ -approximation, temps quasi-linéaire
- ε -test de l'égalité :
 - accepte si $w = w'$
 - rejette si la distance normalisée $\frac{d(w, w')}{\max(|w|, |w'|)} > \varepsilon$

Statistiques uniformes d'un mot

Mots de longueur n ,

$$n = 12$$

$n - k + 1$ blocs de longueur $k = 1/\varepsilon$

$$w = 001010101110$$

$$u.stat(w) = 1/(n - k + 1) \cdot \begin{pmatrix} \#n_1 \\ \dots \\ \dots \\ \#n_{2^k} \end{pmatrix} \quad u.stat(w) = 1/11 \cdot \begin{pmatrix} 1 \\ 4 \\ 4 \\ 2 \end{pmatrix}$$

n_1 nombre de 00...0

Pour $k = 2$, $n - k + 1 = 11$

n_2 nombre de 00...1

...

n_{2^k} nombre de 11...1

La distance normalisée $\frac{d(w,w')}{n} \approx |u.stat(w) - u.stat(w')|$

lorsque les mots sont de longueur proche

- **ε -Testeur de l'égalité**

- échantillonner N sous-mots de longueur k ,
- estimer les **statistiques uniformes** de w, w' par $Y(w)$ et $Y(w')$
- si $|Y(w) - Y(w')| < \varepsilon$, accepter, sinon refuser

- **Appartenance à un langage régulier**

- précalculer le **polyèdre** H_ε associé à l'automate, composé des polygones des statistiques des boucles de longueur k
- calculer l'approximation $Y(w)$ de la **statistique uniforme** du mot w
- si $dist_1(Y(w), H_\varepsilon) < \varepsilon$, accepter, sinon refuser

Complexité des Automates finis non déterministes

- **Reconnaissance d'une entrée**
Temps linéaire \longrightarrow Temps constant
- **Equivalence de 2 automates**
PSPACE-complet \longrightarrow Temps polynomial
- **Equivalence de 2 automates à pile**
Indécidable \longrightarrow Temps exponentiel

Résultats :[ICALP04] [ECCC05] [LICS06]

- Une **théorie de l'approximation** pour la satisfaction et l'équivalence logiques
- **Testeur** pour la distance sur les **mots**
- Testeur pour l'appartenance à un langage **régulier**
- Extensions aux langages **context-free** et aux langages d'arbres
- Algorithmes probabilistes **simples**, mais preuves **difficiles**

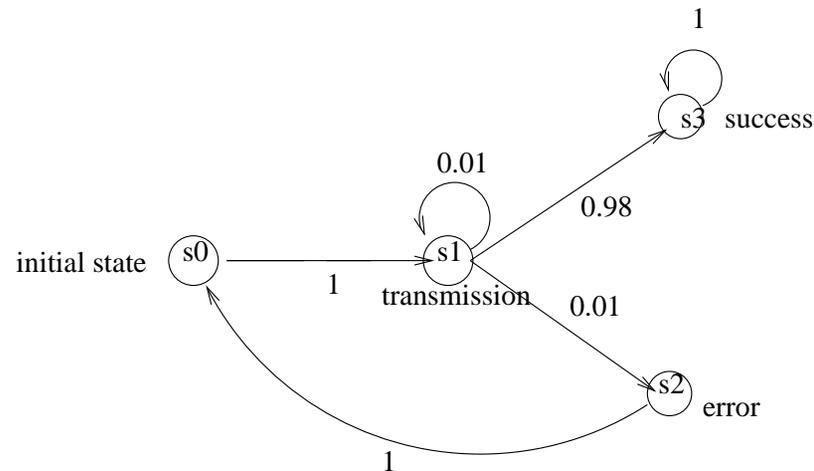
Applications :[XSym04] [ISIP05] [ICDT07]

- Notion de testeur **tolérant** \implies Possibilité de **correction**
- Testeur pour la distance entre deux **fichiers XML**
- Testeur pour la distance entre deux **DTD**
- **Correcteur** pour un fichier proche d'une DTD

Systemes de Transition Probabilistes

Entrée :

- Modèle $\mathcal{M} = (S, P, L)$ et état initial s_0
- $P : S^2 \rightarrow [0, 1]$ Fonction de probabilité
- $L : S \rightarrow 2^{AP}$ (étiquetage des états)
- Formula ψ (**LTL**)



Sortie : $Prob_{\Omega}[\psi]$

Exemple : $\psi \equiv transmission \text{ Until } success$

(Ω **espace probabiliste** des chemins d'exécution d'origine s_0)

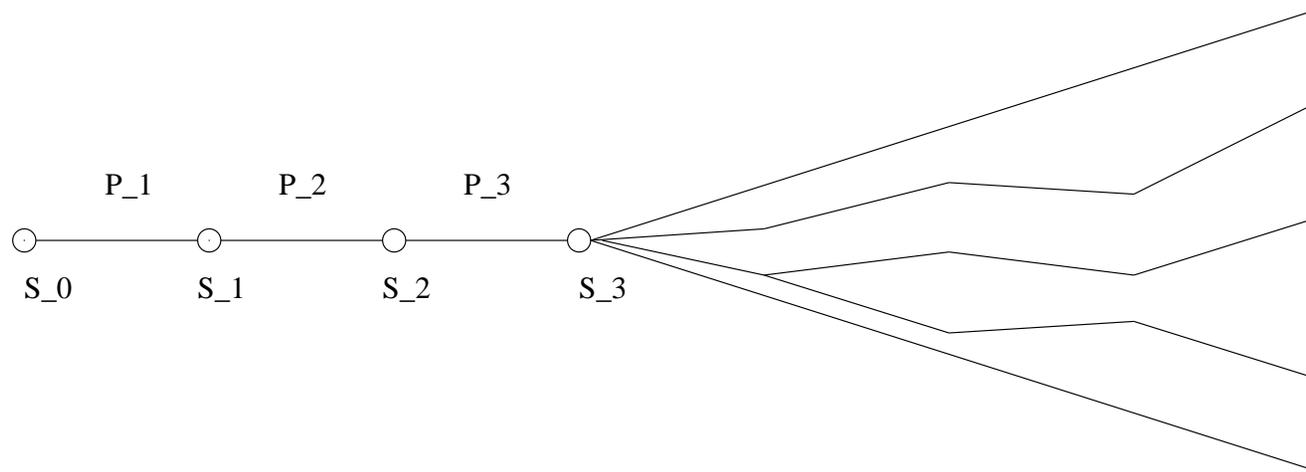
Espace probabiliste :

Cône des extensions d'un chemin fini $\rho = (s_0, s_1, \dots, s_n)$:

$Prob(\{\sigma / \sigma \text{ est un chemin et } (s_0, s_1, \dots, s_n) \text{ est un préfixe de } \sigma\}) =$

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

La mesure peut être définie sur la famille borélienne engendrée par les ensembles $\{\sigma / \rho \text{ préfixe de } \sigma\}$ où ρ est un chemin fini.



L'ensemble des chemins $\{\sigma / \sigma(0) = s_0 \text{ et } \mathcal{M}, \sigma \models \psi\}$ est mesurable.

Complexité : (Coucourbetis and Yannakakis) [CY95]

Vérification qualitative (i.e. prob ≥ 0 ?)

Même **complexité** que **model checking pour LTL**

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

Vérification Quantitative (i.e. prob = ?)

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

Méthode : Calculer $Prob_{\Omega}[\psi]$

- Transformer étape par étape la formule et la chaîne de Markov \mathcal{M}
- Eliminer les connecteurs temporels un par un
- En préservant la probabilité de satisfaction
- En résolvant un système d'équations linéaires de taille $|M|$.

Problèmes d'**énumération** : (L. Valiant 79)

- $\#P$ classe des problèmes d'**énumération** associés aux problèmes de décision NP
- $\#SAT$ est un problème $\#P$ -complet

Schéma probabiliste d'approximation :

(R. Karp and M. Luby 85)

Algorithme probabiliste A

- Entrée : instance x d'un problème d'énumération, $\varepsilon, \delta > 0$
- Sortie : valeur $A(x, \varepsilon, \delta)$ telle que

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

Schéma probabiliste d'approximation pleinement polynomial :

Le temps de calcul est $poly(|x|, (1/\varepsilon), \log(1/\delta))$ **FPRAS**

Schémas probabilistes d'approximation classiques :

- Approximation de $\#DNF$ (Karp, Luby, Madras 89)

Entrée : Formule propositionnelle sous forme normale disjonctive Φ

Sortie : Nombre de valuations satisfaisant Φ

- Approximation de la **fiabilité d'un graphe** (Karger 99)

Entrée : un graphe dont les arêtes ont une probabilité de disparition

Sortie : la probabilité que le graphe reste connexe

Peut-on approximer efficacement $Prob_{\Omega}(\psi)$?

Cas général : (R. Lasseigne and S. Peyronnet 05)

L'existence d'un schéma probabiliste d'approximation pleinement polynomial pour calculer $Prob_{\Omega}(\psi)$ ($\psi \in LTL$) entrainerait que $BPP = NP$.

BPP : Classe de complexité des problèmes décidables par un algorithme probabiliste de Monte-Carlo (avec erreur des 2 côtés).

Bounded-error **P**robabilistic **P**olynomial time : classe des langages L tels que

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

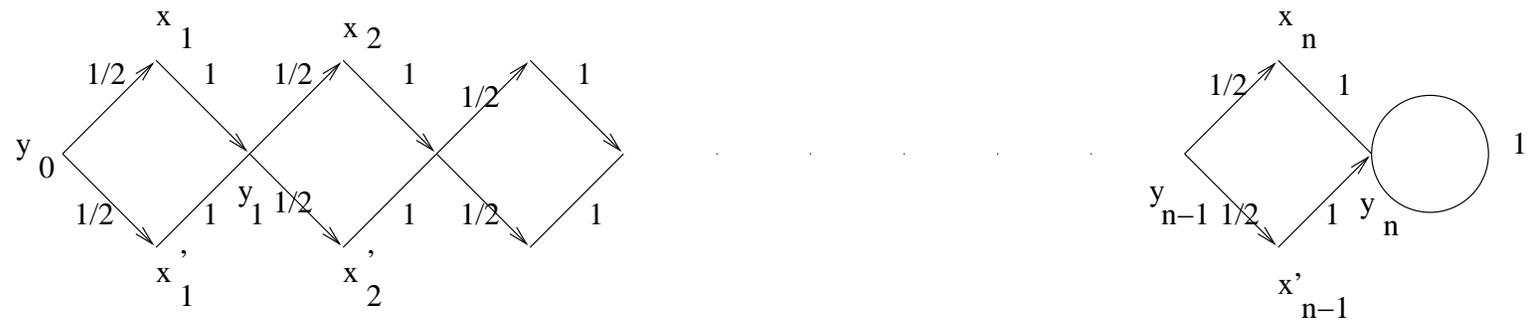
$$x \notin L : Prob[\text{acceptation de } x] \leq 1/4$$

#SAT peut se réduire à l'énumération des chemins de longueur $2n$, dont les extensions infinies satisfont une certaine formule LTL.

Instance de *#SAT* :

- variables propositionnelles $\{x_i \mid 1 \leq i \leq n\}$
- clauses propositionnelles : c_1, \dots, c_m

Formule LTL $\psi : \bigwedge_{i=1}^n Fc_j$



Etiquettes des états :

$$L(x_i) = \{c_j \mid x_i \text{ apparaît dans } c_j\} \quad (i = 1, \dots, n)$$

$$L(x'_i) = \{c_j \mid \neg x_i \text{ apparaît dans } c_j\} \quad (i = 1, \dots, n)$$

Idée de la preuve :

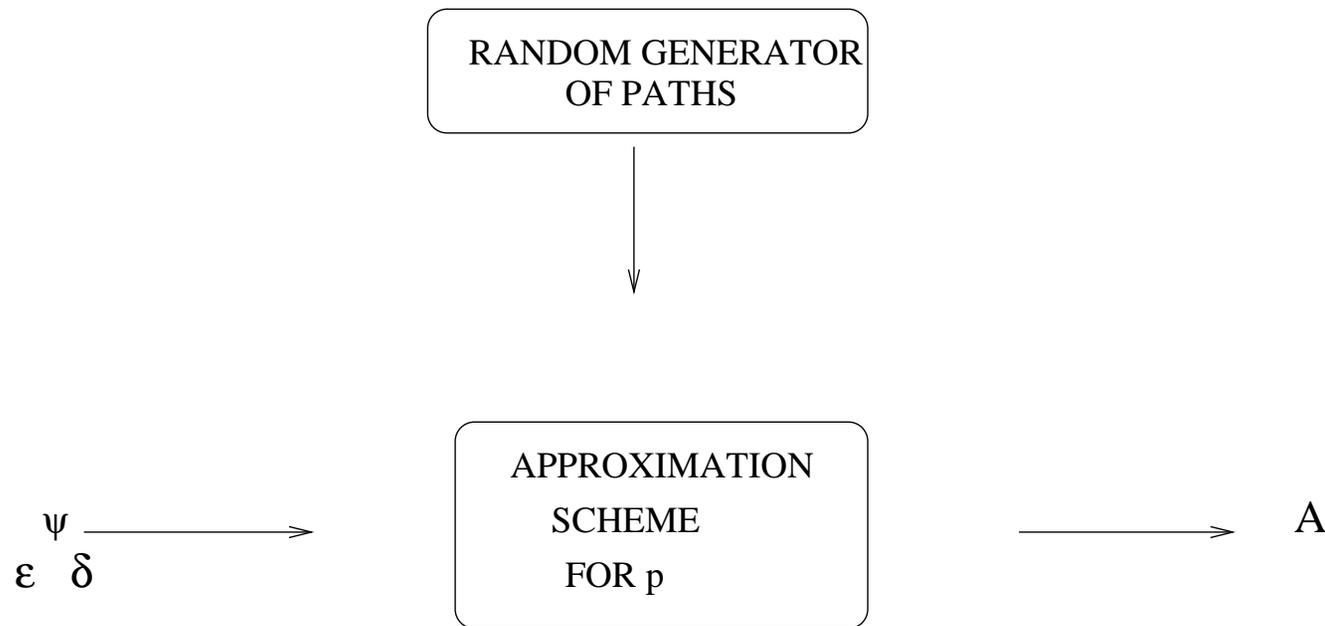
- Compter le nombre de ces chemins fournit $Prob_{\Omega}(\psi)$
- S'il existait un **FPRAS** pour calculer $Prob_{\Omega}(\psi)$, alors il existerait un algorithme probabiliste d'approximation pour $\#SAT$ en temps polynomial
- Un **FPRAS** pour $\#SAT$ permettrait de distinguer, en temps polynomial, pour une entrée x , entre le cas $\#(x) = 0$ et le cas $\#(x) > 0$
- Alors il existerait un algorithme probabiliste en temps polynomial pour décider SAT et $BPP = NP$

D'autre part

- Il n'existe pas d'algorithme déterministe d'approximation en temps polynomial ni pour $\#SAT$ ni pour calculer $Prob_{\Omega}(\psi)$ (Jerrum et Sinclair : les problèmes $\#P$ -complets soit admettent un **FPRAS**, soit ne sont pas approximables du tout)

Schéma probabiliste d'approximation

On désire approximer une probabilité p .



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

ε : paramètre d'approximation (additive)

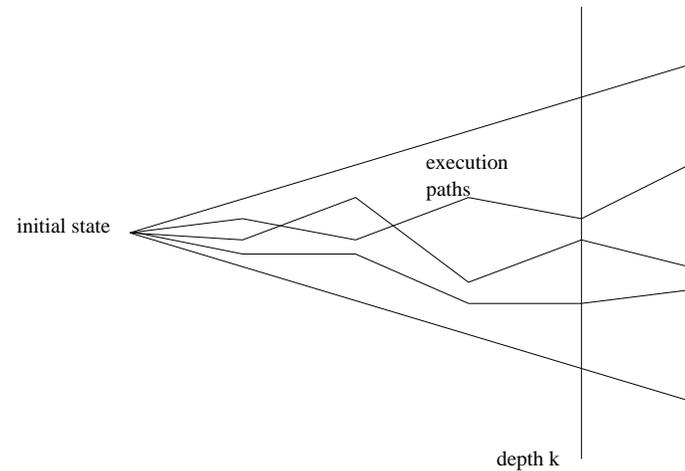
δ : paramètre de confiance (algorithme probabiliste)

Le schéma est dit pleinement polynomial si le temps est

$poly(|\text{entrée}|, (1/\varepsilon), \log(1/\delta))$

On considère $Prob_k(\psi)$ où :

- l'espace probabiliste est l'espace des chemins de longueur $\leq k$



- ψ exprime une propriété **monotone**

$$\lim_{k \rightarrow \infty} Prob_k(\psi) = Prob_{\Omega}(\psi)$$

Algorithme générique d'approximation \mathcal{GAA}

input : $\phi, \text{diagramme}, \varepsilon, \delta$

$A := 0$

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

Pour $i := 1$ à N

- Engendrer de manière aléatoire un chemin σ de longueur k
- Si ψ est vraie sur σ alors $A := A + 1$

Retourner (A/N)

Algorithme basé sur une estimation de type Monte-Carlo et la borne de Chernoff-Hoeffding

Diagramme : représentation succincte du système
(par exemple en Reactive Modules)

Méthode :

Estimation (Monte-Carlo) + borne de Chernoff-Hoeffding

X variable de Bernoulli $(0, 1)$ avec probabilité de succès p

- Faire N tirages aléatoires indépendants X_1, X_2, \dots, X_N
- Estimer p par $\mu = \sum_{i=1}^N X_i / N$ avec erreur ε
- La taille de l'échantillon N est telle que la probabilité d'erreur (de l'algorithme) $< \delta$

Borne de Chernoff-Hoeffding :

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

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

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

Théorème :

\mathcal{GAA} est un FPRAS pour $Prob_k(\psi)$

Méthodologie : Pour approximer $Prob_\Omega(\psi)$

- Choisir $k \approx \log|M| \cdot \ln(1/\varepsilon)$
- Itérer l'approximation de $Prob_k(\psi)$

Corollaire :

L'algorithme de point fixe obtenu en itérant \mathcal{GAA} est un schéma probabiliste d'approximation en **espace logarithmique** pour $Prob(\psi)$

Remarque :

- La longueur des chemins nécessaires peut être le **diamètre** du système
- La vitesse de **convergence** peut être lente, mais la **complexité en espace** est **logarithmique**...

Résultats : [VMCAI04] [WoLLIC05] [QEST06]

- Approximation efficace de la vérification de propriétés quantitatives d'**accessibilité** et de **suret **
- Non existence d'un algorithme g n ral d'approximation (relative) en temps polynomial
- R duction exponentielle de la **complexit  en espace**
- Extension aux cha nes de Markov en temps continu
- Un **outil** de v rification approch e **efficace**.

- Implémentation **distribuée** de la méthode d'approximation [PCMC05]
- Extension (XRM) du langage standard de modélisation des systèmes probabilistes en temps discret et en temps continu (**APMC 3.0**) [QEST06]
- Nombreux **cas d'étude** (algorithmes distribués, protocoles de communication, réseaux de capteurs, modèles biologiques,...) [AVoCS04] [AVoCS06] [ISoLA06]
- Outil de **vérification** et de **simulation** pour des valeurs réelles des paramètres du modèle (pas d'explosion en espace).
- **Intégration** avec le model checker probabiliste **PRISM** (Birmingham)
- Collaboration avec le projet Contraintes (INRIA) pour la simulation de **modèles biochimiques**

- Théorie de l'**approximation** pour la **vérification**
- Validité et **robustesse** de la méthode d'approximation sur les données pour la vérification classique
- Nouveau type d'applications
(données **gigantesques** et **bruitées**)
- Efficacité théorique des **schémas d'approximation** pour la vérification probabiliste
- Efficacité pratique de l'outil **APMC**
- Nouvelles collaborations (Technion, Birmingham, New York)

Perspectives

- Approximation du test de propriété pour une “boîte noire” (**Black Box Checking**) à l’aide de techniques d’apprentissage (**PAC Learning**)
- Comparaison avec les approches étendant la **bisimulation** aux systèmes probabilistes et utilisant d’autres types de **métriques**
- Approximation à partir d’une **représentation succincte** du système pour la vérification classique
- Intégration des deux approches (approximation sur les **données** et approximation **probabiliste**)

- [CY95] C. Courcoubetis and M. Yannakakis. *The complexity of probabilistic verification*. JACM, 24(4), p. 857-907, 1995.
- [LICS06] E. Fisher, F. Magniez and M. de Rougemont. *Approximate Satisfiability and Equivalence*. Proc. 21st IEEE Symposium on Logic In Computer Science, 2006.
- [VMCAI04] T. Héroult, R. Lassaigne, F. Magniette and S. Peyronnet. *Approximate Probabilistic Model Checking*. Proc. 5th Int. Conf. on Verification, Model Checking and Abstraction, LNCS n° 2937, p. 73-84, 2004.
- [QEST06] T. Héroult, R. Lassaigne, and S. Peyronnet. *APMC 3.0 : Approximate verification of Discrete and Continuous Time markov Chains*. Proc. 3rd Int. Conf. on Quantitative Evaluation of Systems, 2006.

- [KLM89] R. Karp, M. Luby and N. Madras. *Monte-Carlo Approximation Algorithms for Enumeration Problems*. Journal of Algorithms 10, 429-448, 1989.
- [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, 360 p., 2003.
- [ICALP04] F. Magniez and M. de Rougemont. *Property testing of regular tree languages*. Proc. 31st ICALP, p. 932-944, 2004.
- [Var85] *Automatic verification of probabilistic concurrent finite-state programs* Proc. 26th IEEE FOCS, p. 327-338, 1985.