Verification approchée

Michel de Rougemont,
University Paris II et LRI

Projet Vera

1. LRI: Algorithmes et Complexité: Sophie Laplante, Frédéric Magniez, Michel de Rougemont, Miklos Santha

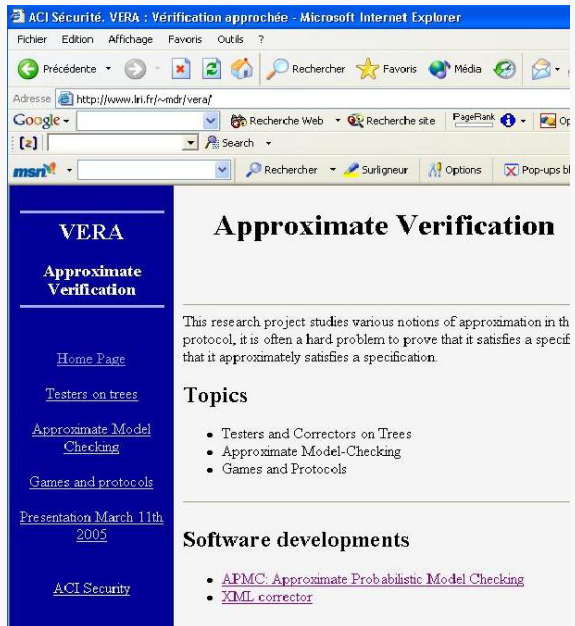
Génie Logiciel: Marie-Claude Gaudel

Parallélisme, Thomas Hérault

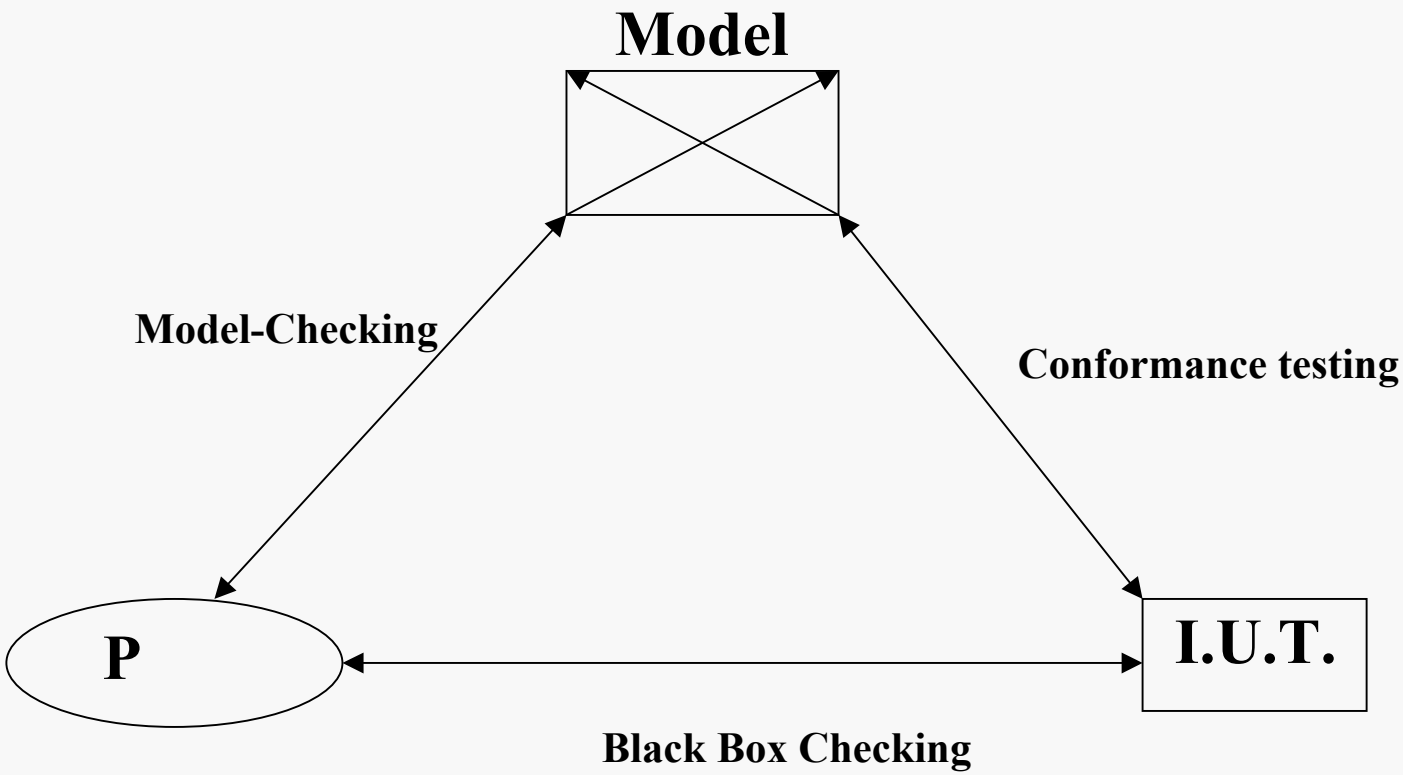
2. Equipe de Logique, Université Paris VII: R. Lassaigne, S. Peyronnet

Topics: http://www.lri.fr/~mdr/vera/

1. Testers and Correctors
2. Approximate Model-Checking
3. Games and Protocols



Model Checking et Test



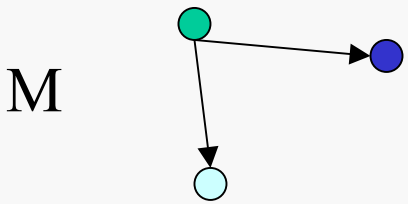
Exemple: automate fini, P: 0*1*

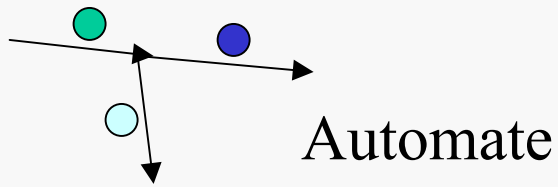
Vérification

1. Logique $M \models \Theta$ (M satisfies Θ)

$M = (S, R, P_1, \ldots P_k, s)$ Structure de Kripke, $R \subseteq S.S$ and $P_i \subseteq S$, $s \in S$

Θ: $E(p_1 U p_2)$ formule CTL





2. Complexité: $f(x)=y$

   Pour x,y donnés, vérifier $f(x)=y$

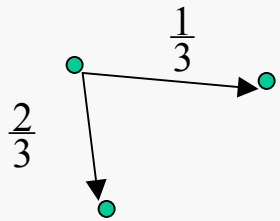
   Exemple: $Permanent\,(A)=y$

Vérification probabiliste

1. Logique $\quad M \models \text{Prob}[\Theta] > \frac{1}{2}$

$M = (S, R, P_1, \ldots P_k, s)$ où $R \subseteq S.S.[0,1]$ et $P_i \subseteq S$, $s \in S$

Θ: formule LTL (par exemple $(p_1 \mathrm{U} p_2)$)



2. Complexité

Pour x,y donnés, vérifier avec un protocole proba. $f(x)=y$

IP : Interactive proof

PCP: Probabilistic Checkable Proof

Test de Propriété (Property Testing)

Approximations en MC et test

1. Model-Checking: Automate, OBDD, SAT

   Problème: explosion combinatoire

   Approximations en Model-Checking:

   BMC: Bounded Model-Checking

   SAT solveurs

   Méthodes d'Abstraction

   Probabilistic MC $\quad \text{Prob}_{\Omega}\ [\ (p_1 \text{U} p_2)\ ]$

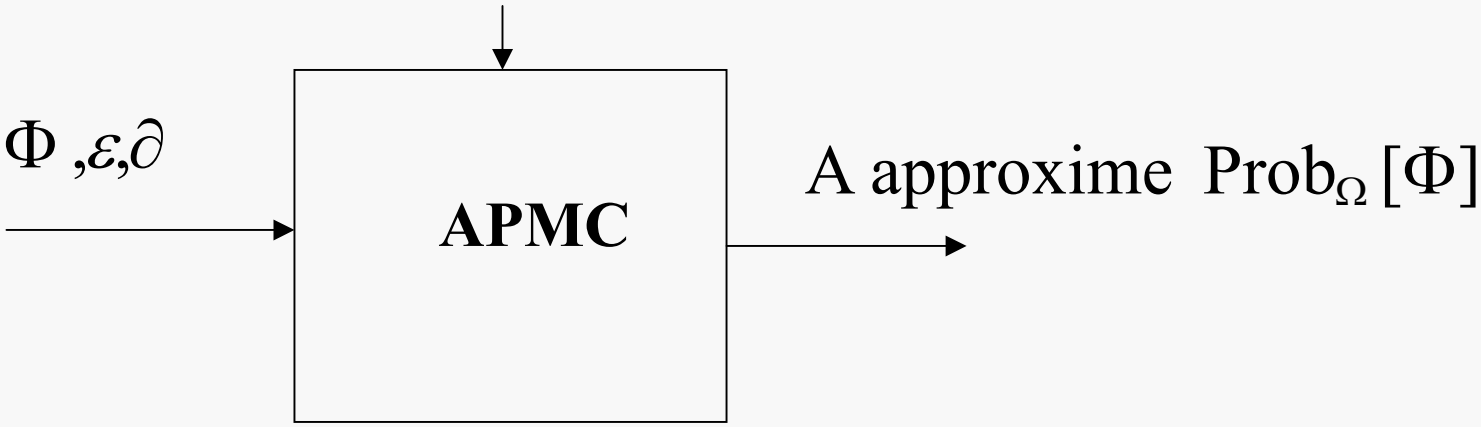
2. Test de propriété

   PAC Learning

   Tests Statistiques

APMC

Générateur aléatoire de chemins



$$Prob_\Omega [\ \Phi\ ]=p$$

$$Prob_\Omega [\ p-\varepsilon \leq\ A\ \leq p+\varepsilon\ ] \geq 1-\delta$$

ε=distance, δ= paramètre de confiance

Model Checking approché

Testeurs et correcteurs:

Une propriété est ε testable s'il existe un algorithme efficace qui distingue une structure ε loin d'une structure satisfaisable.

Satisfiabilité approchée:

U |= F se géneralise à U $|=_{\varepsilon}$ F

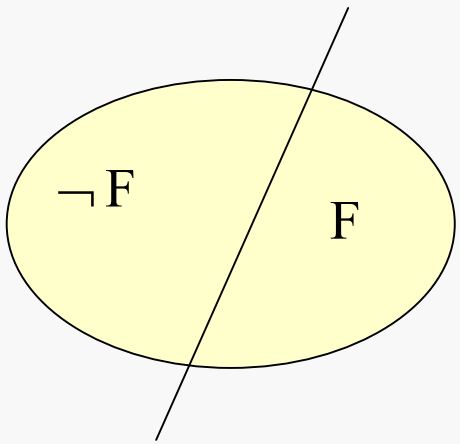
S'il existe U' tel que dist(U,U')< ε et U' |= F

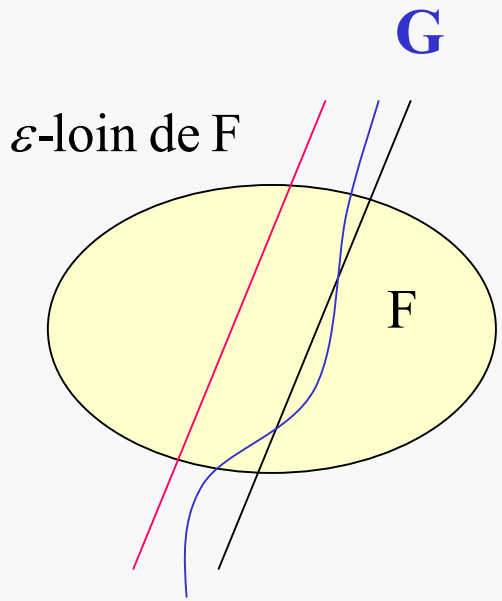
Application au Model-Checking: $M =_{\varepsilon} \Theta$

Satisfiabilité et Equivalence approchées

1. Satisfiabilité : $T \models F$
2. Satisfiabilité approchée $T \models_{\varepsilon} F$
3. Equivalence approchée $F \equiv_{\varepsilon} G$

Image sur une classs K d'arbres





Distance d'Edition avec déplacements

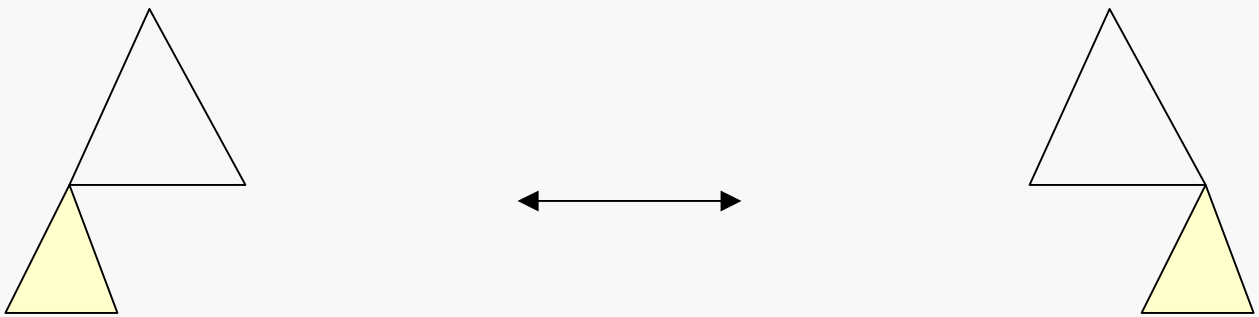
1. Distance d'Edition: Insertions, Effacements, Modifications
2. Distance Edition avec déplacements:

$$dist(W,W') \;\; ; \;\; dist(W,L) = Min_{W' \in L} \; \{dist(W,W')\}$$

01110000111100011001

01110111100000011001

3. Distance Edition avec déplacements se généralise aux arbres ordonnés



Statistiques uniformes d'un mot

W=001010101110 longueur n, n-k+1 blocs de longueur k=1/ε

$$u.stat(W)=\begin{pmatrix}\#n_1\\ ..\\ ..\\ \#n_{2^k}\end{pmatrix}.\frac{1}{n-k+1}$$

$\#n_1$ number of "00...0"
$\#n_2$ number of "00...1"
.. ..
.. ..
$\#n_{2^k}$ number of "11...1"

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

$$u.stat(W)=\begin{pmatrix}1\\ 4\\ 4\\ 2\end{pmatrix}.\frac{1}{11}\approx u.stat(W+\varepsilon)$$

$dist(W,W')\approx\left|u.stat(W)-u.stat(W')\right|$, lorsque les mots sont de longueur proche,

Distance de mots:

- NP-complet
- Testable, O(1): échantillonner N sous-mots de longueur k: Y(W) et Y(W')

Si |Y(w)-Y(w')| <ε. accepter, sinon rejeter

Testeur pour un langage régulier

Automate A définit L, et un polytope H dans l'espace des u.stats

$$u.stat(W) \approx \begin{pmatrix} 0.5-\varepsilon/2 \\ \varepsilon/2 \\ \varepsilon/2 \\ 0.5-\varepsilon/2 \end{pmatrix} \approx u.stat(Z) \approx u.stat(Y)$$

$$\begin{pmatrix} 0.25 \\ 0{,}25 \\ 0{,}25 \\ 0{,}25 \end{pmatrix} \approx u.stat(T)$$

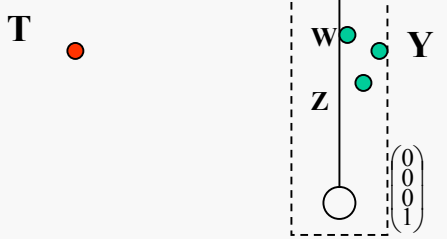
W: 00000000001111111111

Y: 00000100001111101111

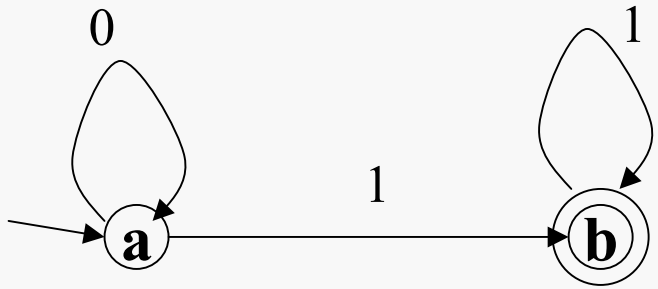
Z: 11111111111000000000

T: 01001010001011000111010101

H



A



Testeur x dans L:

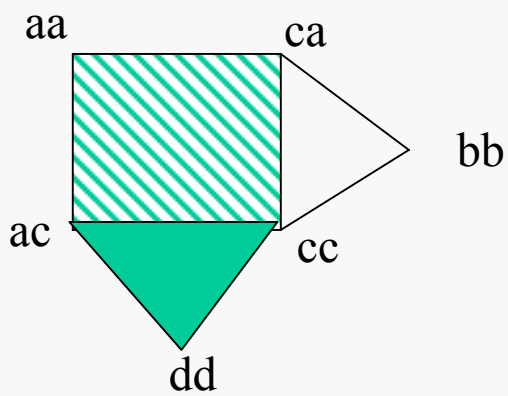
- Testable, O(1): calculer Y(W),
- Si dist(Y(w),H) <ε. accepter, sinon rejeter

Remarque: robustesse au bruit.

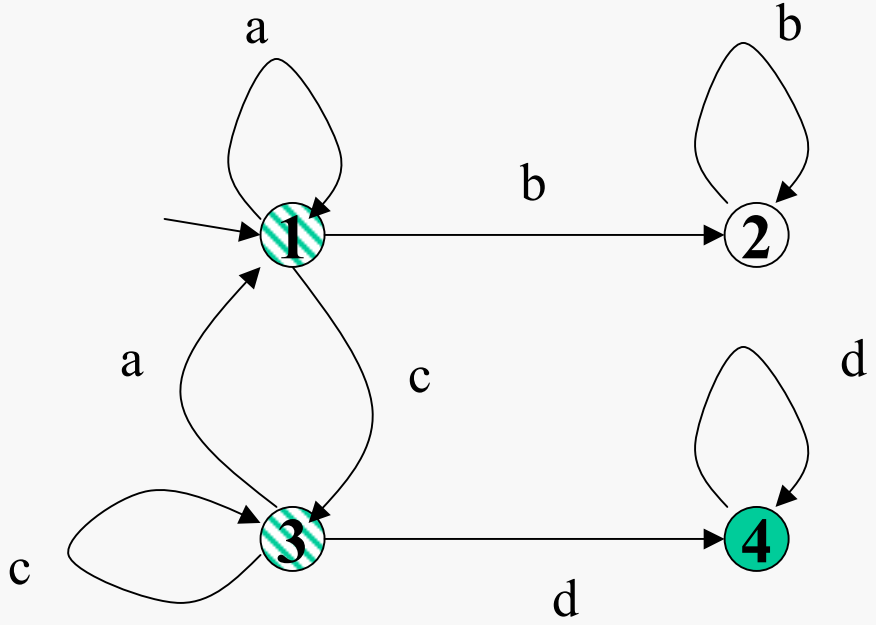
Exemple de couple (A,H)

Blocs, k=2, m=4, | Σ |=4, | Σ| ^{k} +1=17:

Boucles de taille 1 bloc: {(aa,ca:1),(bb,2),(cc,ac:3),(dd:4)}

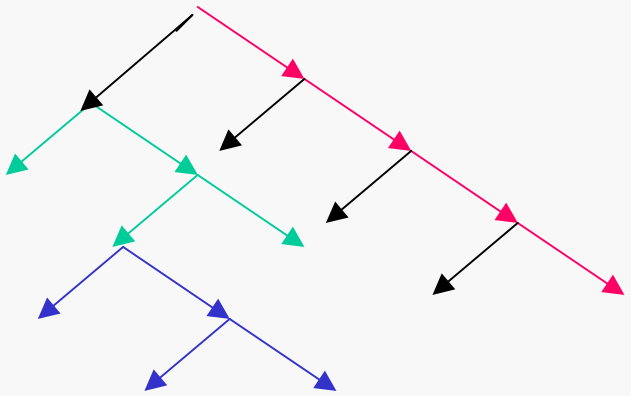


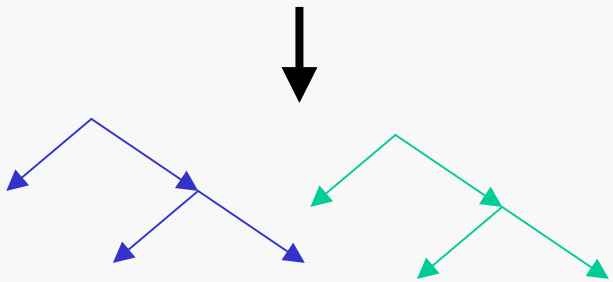
H

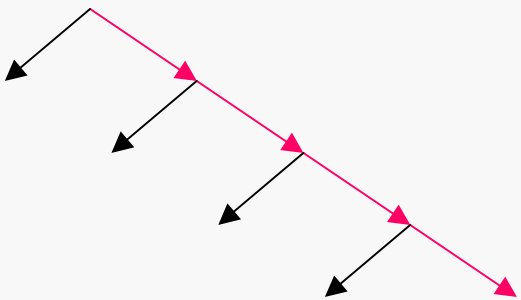


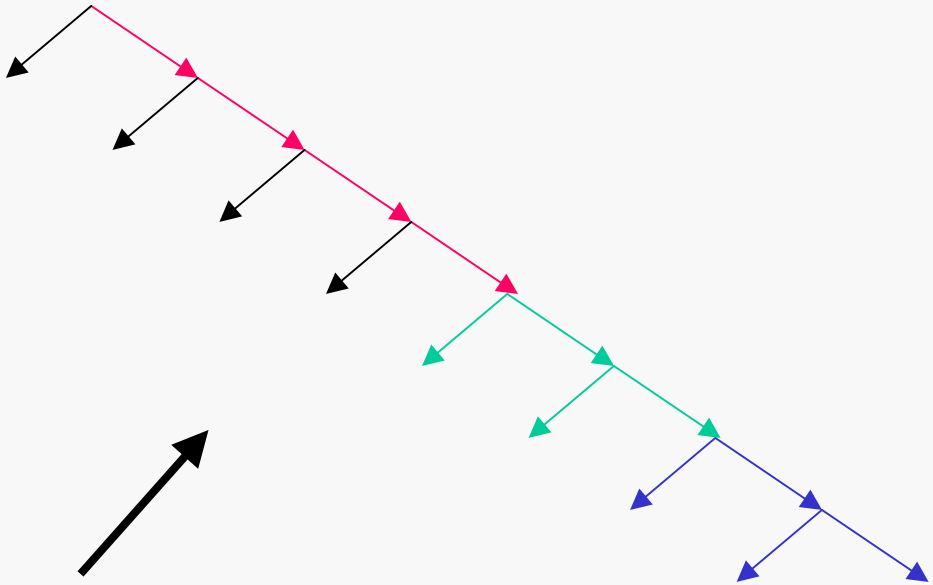
A

Correction d'un arbre ordonné





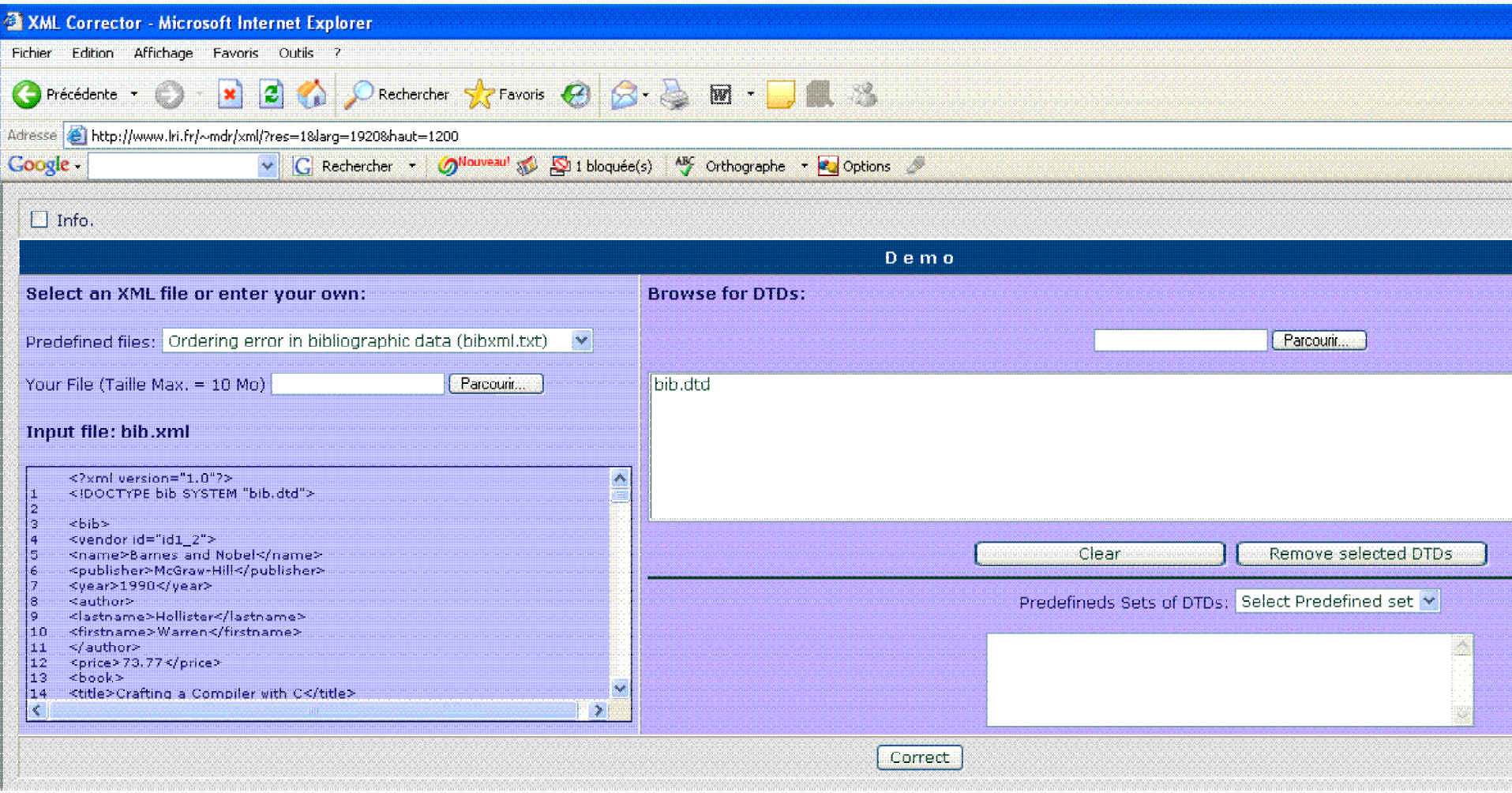




2 moves, dist=2

Automate d'arbre ou DTD:
t: l,r
r: l,r

Correcteur XML: http://www.lri.fr/~mdr/xml/



Applications

Testeur:

- Estimateur de la distance entre deux fichiers XML,
- Décide si un fichier XML F est ε-valide,
- Décide si deux DTDs sont proches.

Correcteur: Si un fichier XML F est ε-proche d'une DTD,

- Trouve F' valide ε-proche de F;
- Classe les fichiers XML du Web pour un ensemble de DTD's

Vérification de programmes:

- Décide si deux automates sont ε-proches en temps polynomial.
- Model-Checking approché: http://www.lri.fr/~mdr/vera/
  - Langage de spécification
  - Modèle
  - Distance

Conclusion

1. Testeurs et Correcteurs:
   - Techniques statistiques
   - Logique et hasard
2. Généralisation aux modèles probabilistes
3. Application aux jeux et protocoles

Références:

1. Robust characterizations of polynomials, R. Rubinfeld, M. Sudan, 1994
2. O. Goldreich, S. Goldwasser and D. Ron, Property Testing and its connection to Learning and Approximation, 1996.
3. Property testing for regular tree languages, Mdr, F. Magniez (Icalp 2004) (.pdf)
4. Correctors for XML data, U. Boobna, M. de Rougemont (XSym 2004) (.pdf)
5. Property and Equivalence Testing on strings, E. Fischer, F. Magniez, M. de Rougemont (ECCC 2005)(.pdf)
6. http://www.lri.fr/~mdr/xml/ et http://www.lri.fr/~mdr/vera/