



# VERA

## VERification Approchée

LRI-Orsay

Equipe de Logique Paris VII

<http://www.lri.fr/~mdr/vera.htm>

**Michel de Rougemont**  
**Université Paris II**



# Equipes

- LRI-Orsay
  - Sophie Laplante
  - Frédéric Magniez
  - Sylvain Peyronnet
  - Michel de Rougemont
  - Miklos Santha
- Paris VII
  - Richard Lassaigne

# Thèmes de Recherche

## 1. Logique, Testeurs et Correcteurs

- Testeurs et Correcteurs
- Arbres réguliers

## 2. Abstraction probabiliste de programmes

- Bornes inférieures sur OBDDs et automates
- Abstraction en Model Checking

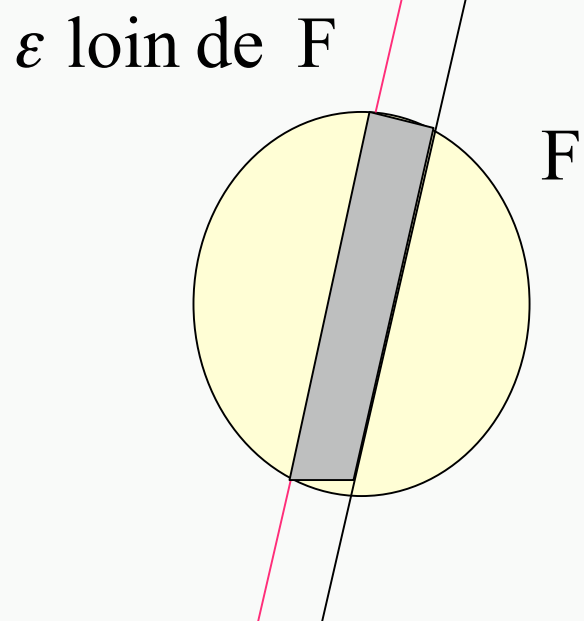
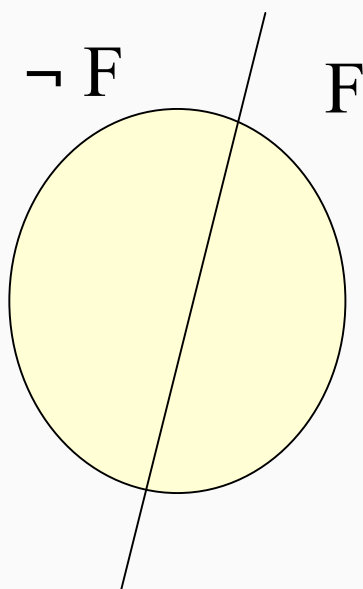
## 3. Mécanismes et Jeux

- Calcul d'équilibre

# Satisfiabilité Approchée

1. Satisfiabilité :  $T \models F$
2. Satisfiabilité approchée  
 $\text{Tree} \models_{\varepsilon} F$

Image sur une classe K d'arbres



# Logic, testeurs, correcteurs

Un **Testeur** decide  $\models_{\in}$  pour une formule  $F$ .

Un **Correcteur** prend une structure  $U$  proche de  $K$  en entrée et calcule  $U'$  dans  $K$ , proche de  $U$ .

**Problème:** Une classe  $K$  définissable dans une logique  $L$  admet-elle un testeur et un correcteur?

**Théorème.** (Alon and al. FOCS2000) Les mots réguliers sont testables pour la distance d'Édition.

Généralisation aux arbres réguliers.

Application au test de fichiers XML et à la correction XML.

# Vérification par Modèle

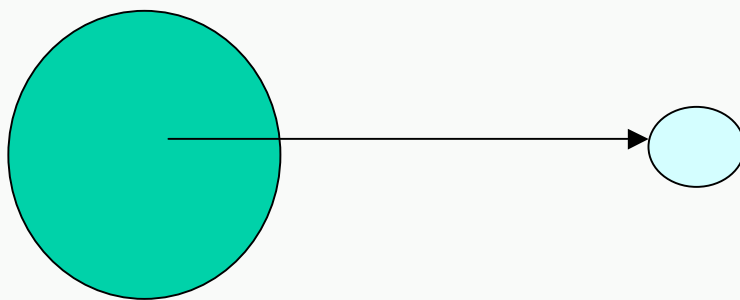
- Programme P



- Spécification F (X,Y)
- Structure de donnée OBDD
$$O_P = O_F$$
- Problème : taille des structures explose
- Complexité en Communication montre des bornes inférieures exponentielles.

# Testeurs et Vérification

- La spécification admet un testeur  
 $P(A) = 1$  ssi  $A$  est 3 coloriable



$A$  est 3-coloriable

$\hat{A}$  est 3-coloriable

- Problème : comment appliquer le test à un programme?

# Abstraction probabiliste

P(A). Enumerate C:  $D_n \rightarrow (1,2,3)$

While ( $x \leq n$ ) {

    While ( $y \leq n$ ) {

        If  $A(x,y)$  check  $C(x)=C(y)$  } }

- Define an abstraction  $\Delta \subseteq D_n$

P(A). Enumerate C:  $\Delta \rightarrow (1,2,3)$

While ( $x \leq m$ ) {

    While ( $y \leq m$ ) {

        If  $A(x,y)$  check  $C(x)=C(y)$  } }

# Peut-on trouver une abstraction?

- Toute propriété  $\forall x \exists y \exists z (P(x, y, z))$

Admet un testeur.

- Comment trouver une abstraction à partir d'un programme P?
- Difficulté algorithmique.

# Applications

## 1. Programmes XML

- Testeurs
- Correcteurs

## 2. Mécanismes et Jeux

- Calcul d'équilibre
- Comment vérifier qu'un programme distribué atteindra un équilibre satisfaisant une propriété P?
- Nash est approximable (2003)
- Mécanismes de sécurité et de régulation

# Conclusion

- Vérifier exactement peut être trop difficile.
  - Vérifier approximativement peut être réalisable.
1. Théorie des Testeurs et Correcteurs
  2. Vérification probabiliste de programmes.
  3. Vérification approchée d'équilibres de protocole.