

# Exploration de traces à l'aide de fouille de données

Peggy Cellier, Mireille Ducassé, Sébastien Ferré

ÉQUIPE LIS – IRISA  
INSA DE RENNES - UNIVERSITÉ DE RENNES



16 mai 2011

Atelier IC "Traces numériques, connaissances et cognition"

# Plan

- 1 Introduction
  - Trace
  - Fouille de données
- 2 Extraction d'information dans les traces
- 3 Perspectives/Discussion

# Trace

- Trace = ens. d'événements associés à un objet ou à un individu pendant une tâche
  - Exemples : traces d'exécution d'un programme, traces d'interaction avec un humain
- Traitement des traces en 3 étapes :
  - Acquisition
  - Formalisation
  - *Exploitation*
    - Exemples : caractériser et expliquer des comportements anormaux, définir des profils de tâches

# Exemple : Trace d'exécution d'un programme

- Rapport sur une exécution :
  - Verdict : **Fail** ou **Pass**
  - Appels de méthodes
  - Lignes exécutées
  - Valeurs des variables
  - ...
- Comment exploiter les indices contenus dans les traces d'exécution pour comprendre les défaillances (Fail) d'un programme ?

# Fouille de données

- Extraction automatique d'information “pertinente” dans de gros volumes de données
  - Trouver des régularités
  - Regrouper des données
  - ...
- Plusieurs techniques :
  - Recherche de règles d'association [Agrawal et al. 93]
  - Extraction de motifs séquentiels fréquents [Agrawal et al. 95]
  - ...
- Applications : marketing, bioinformatique, ...

# Idée

Utiliser la fouille de données pour analyser des traces d'exécution

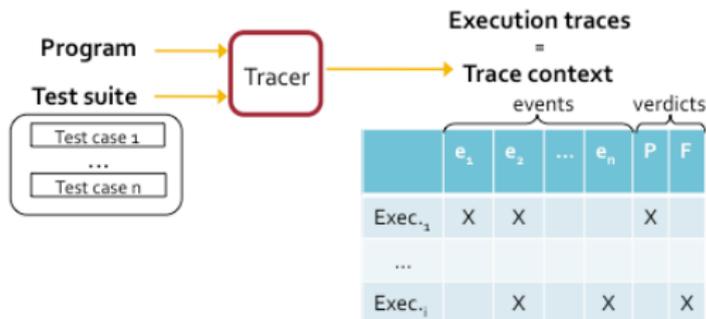
# Plan

- 1 Introduction
- 2 Extraction d'information dans les traces
  - Principe général
  - Application : débogage
- 3 Perspectives/Discussion

# Principe général [Cellier et al. 11]

- Les traces = indices pour une tâche (débogage, analyse du comportement, ...)
- Application de techniques de fouille de données sur l'ens. des traces
  - Extraction de “profils” sous forme de motifs
  - Problème : Beaucoup de motifs extraits
- Proposition : organisation de ces motifs dans un ordre partiel
  - Permet de compacter et structurer les motifs
  - Permet de naviguer à travers eux

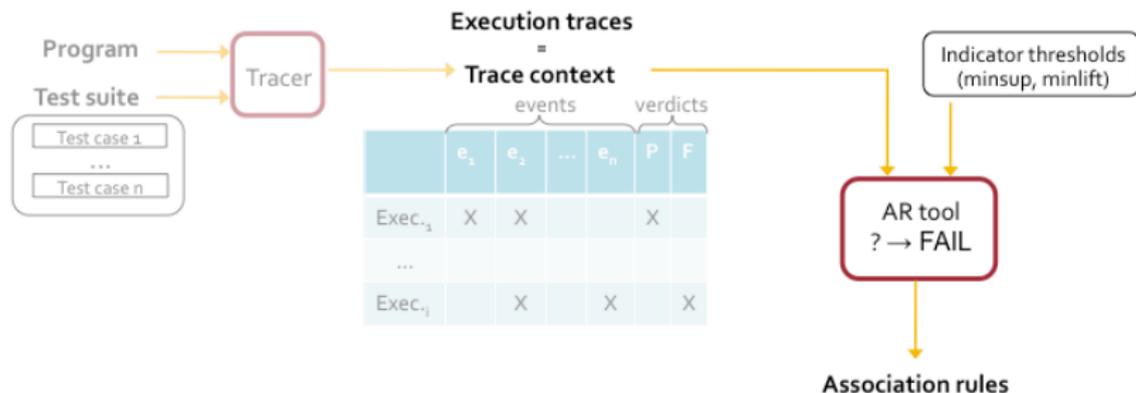
# Application : débogage (1/8) [Cellier et al. 08]



- Contexte de traces
  - Toutes les exécutions du programme
- Description d'une exécution
  - Tous les événements appartenant à la trace (lignes exécutées, valeur des variables, ...)
  - Verdict de l'exécution (Fail ou Pass)

Note : Dans les exemples, les événements sont les lignes exécutées

# Application : débogage (2/8) [Cellier et al. 08]



- Extraction des règles d'association (=motifs)

# Application : débogage (3/8) [Cellier et al. 08]

- **Règles d'association** :  $e_1, \dots, e_n \rightarrow FAIL$

Quand tous les événements  $e_1, \dots, e_n$  appartiennent à une trace, cela implique “la plupart du temps” une défaillance

- **“La plupart du temps” mesuré par des indices statistiques**

- **Support** : fréquence de la règle

Nombre d'exécutions en échec qui contiennent dans leur trace  $e_1, \dots, e_n$

- **Lift** : pertinence de la règle

Comment l'observation de  $e_1, \dots, e_n$  dans la trace augmente la probabilité que le programme échoue ?

# Application : débogage (4/8) [Cellier et al. 08]

- Extrait des règles d'association générées à partir d'un contexte de traces

**R1:**line\_81,line\_84,line\_87,line\_90,line\_105,line\_66,line\_78,line\_62,line\_63,line\_65,line\_67,line\_69,line\_97,line\_112,line\_113,line\_115,line\_116,line\_117,line\_124,line\_126,line\_127,line\_128,line\_129,line\_14,line\_15,line\_16,line\_17,line\_58,line\_93 → **FAIL**

**R2:**line\_78,line\_84,line\_81,line\_90,line\_87,line\_62,line\_63,line\_65,line\_67,line\_69,line\_105,line\_97,line\_112,line\_113,line\_115,line\_116,line\_117,line\_124,line\_126,line\_127,line\_128,line\_129,line\_14,line\_15,line\_16,line\_17,line\_58,line\_93 → **FAIL**

**R3:**line\_66,line\_78,line\_62,line\_63,line\_65,line\_67,line\_69,line\_97,line\_112,line\_113,line\_115,line\_116,line\_117,line\_124,line\_126,line\_127,line\_128,line\_129,line\_14,line\_15,line\_16,line\_17,line\_58,line\_93 → **FAIL**

**R4:**line\_101,line\_84,line\_85,line\_68,line\_81,line\_78,line\_62,line\_63,line\_65,line\_67,line\_69,line\_97,line\_112,line\_113,line\_115,line\_116,line\_117,line\_124,line\_126,line\_127,line\_128,line\_129,line\_14,line\_15,line\_16,line\_17,line\_58,line\_93 → **FAIL**

**R5:**line\_101,line\_81,line\_78,line\_62,line\_63,line\_65,line\_67,line\_69,line\_97,line\_112,line\_113,line\_115,line\_116,line\_117,line\_124,line\_126,line\_127,line\_128,line\_129,line\_14,line\_15,line\_16,line\_17,line\_58,line\_93 → **FAIL**

**R6:**line\_68,line\_78,line\_62,line\_63,line\_65,line\_67,line\_69,line\_97,line\_112,line\_113,line\_115,line\_116,line\_117,line\_124,line\_126,line\_127,line\_128,line\_129,line\_14,line\_15,line\_16,line\_17,line\_58,line\_93 → **FAIL**

**R7:**line\_84,line\_81,line\_78,line\_62,line\_63,line\_65,line\_67,line\_97,line\_112,line\_113,line\_115,line\_116,line\_117,line\_124,line\_126,line\_127,line\_128,line\_129,line\_14,line\_15,line\_16,line\_17,line\_58,line\_93 → **FAIL**

**R8:**line\_81,line\_78,line\_62,line\_63,line\_65,line\_67,line\_69,line\_97,line\_112,line\_113,line\_115,line\_116,line\_117,line\_124,line\_126,line\_127,line\_128,line\_129,line\_14,line\_15,line\_16,line\_17,line\_58,line\_93 → **FAIL**

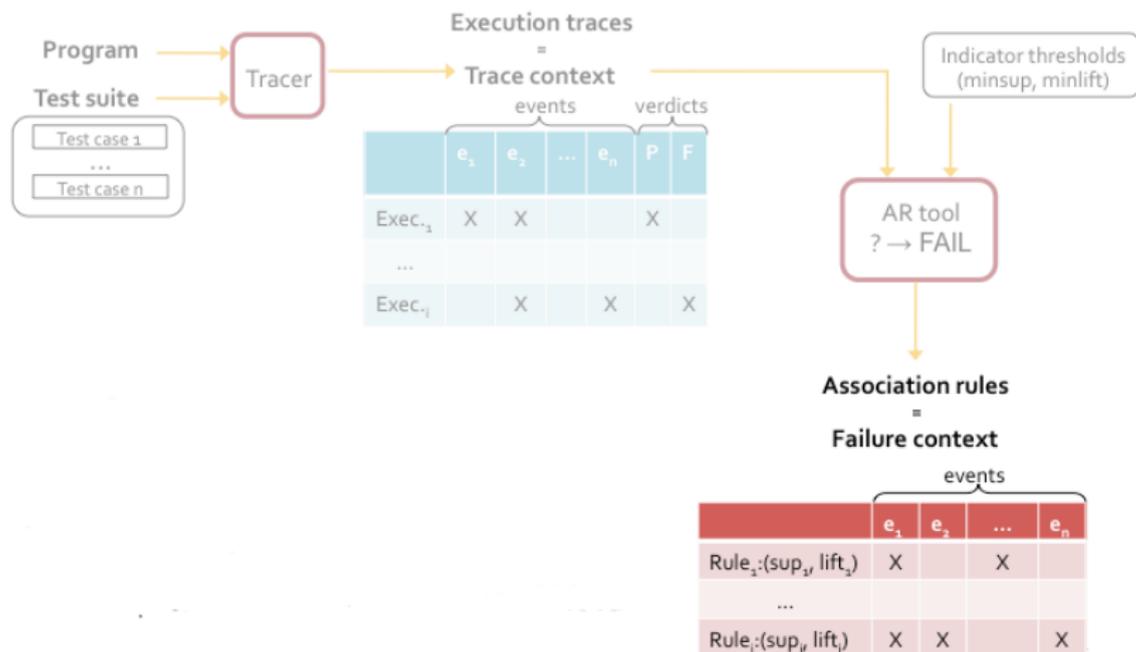
**R9:**line\_78,line\_62,line\_63,line\_65,line\_67,line\_69,line\_97,line\_112,line\_113,line\_115,line\_116,line\_117,line\_124,line\_126,line\_127,line\_128,line\_129,line\_14,line\_15,line\_16,line\_17,line\_58,line\_93 → **FAIL**

- Problème : beaucoup de règles avec de grosses prémisses

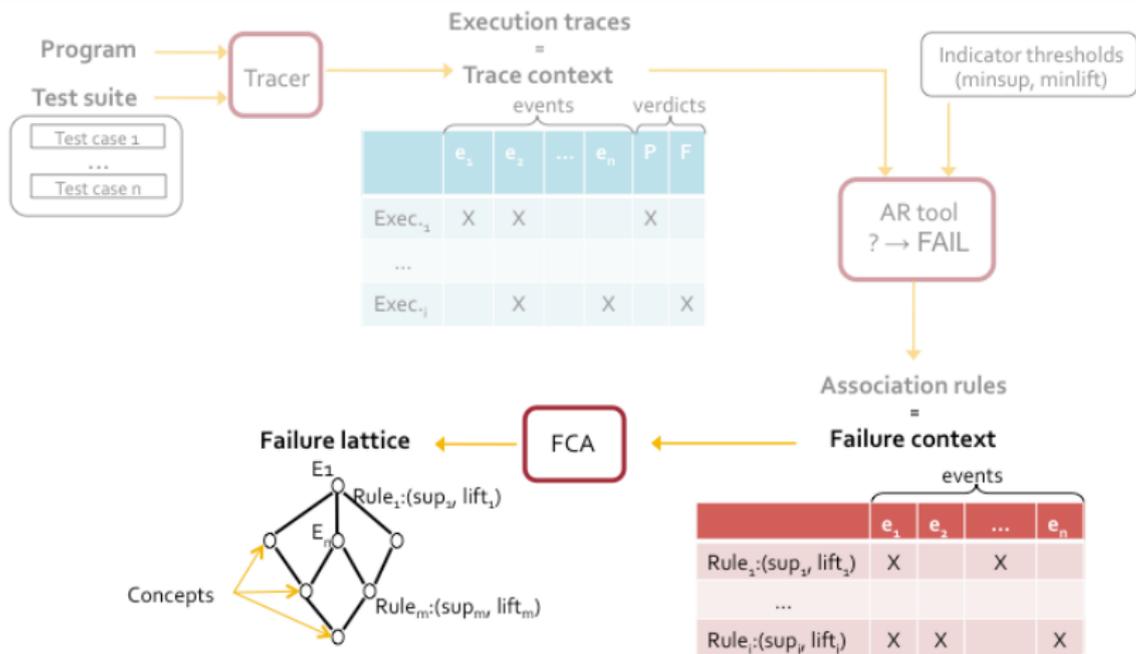
# Application : débogage (5/8) [Cellier et al. 08]

- Certaines règles sont plus spécifiques que d'autres
  - R4 : line 84, line 81, line 93, line 105, line 90, line 87 → Fail  
est plus spécifique que
  - R5 : line 84, line 81, line 93 → Fail
- Règles partiellement ordonnées  $\Rightarrow$  Contexte des défaillances

# Application : débogage (6/8) [Cellier et al. 08]



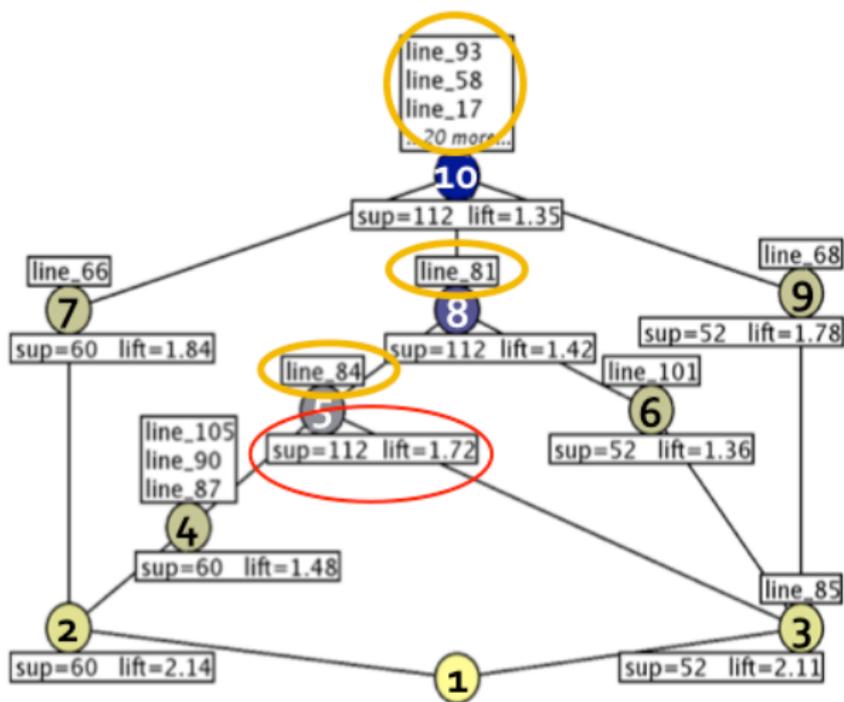
# Application : débogage (6/8) [Cellier et al. 08]





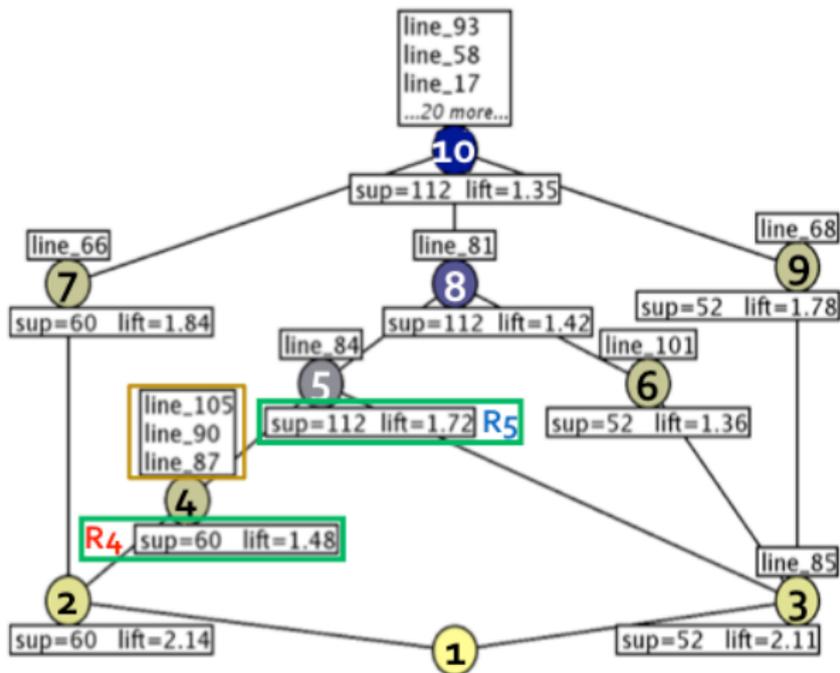
# Application : débogage (7/8) [Cellier et al. 08]

Lecture de la prémisse de la règle R5



# Application : débogage (7/8) [Cellier et al. 08]

R4 plus spécifique que R5



# Application : débogage (8/8) [Cellier et al. 08]

- Débogage
  - Parcours du treillis du bas vers le haut
  - càd des indices les plus spécifiques aux plus généraux
- Avantage par rapport aux méthodes existantes
  - Structure de données : ordre partiel
    - Mise en avant des relations dynamiques qui existent entre les différents événements
    - Propriétés intéressantes pour gérer des programmes contenant plusieurs fautes

# Plan

- 1 Introduction
- 2 Extraction d'information dans les traces
- 3 Perspectives/Discussion

# Perspectives/Discussion

- Formalisation et définition de la trace
  - Impact important sur le résultat de la fouille
    - Mauvaise modélisation  $\Rightarrow$  pas de résultat intéressant
- Enrichissement des informations contenues dans les traces
  - Prise en compte de l'ordre des événements
    - Idée : Motifs séquentiels
  - Prise en compte de la répétition
    - Exemple : 1 fois événement1 puis 3 fois événement2 puis 1 fois événement3 ...
  - Prise en compte d'informations statiques
    - Exemples : arbre de syntaxe abstraite, relations entre les attributs
    - Idée : Taxonomies, Camelis 2 (démo IC jeudi 19 mai)
- Quelles tâches/questions sur les traces ?
  - Autres types de données
    - Exemple : Traces d'interaction avec un humain
  - ...