De ce fait, tout outil automatique d'analyse statique ne peut pas, en général, prouver que le programme étudié répond sur des cas particuliers, soit sur des approximations : chaque cas discrimination (ou approximation) ou, à des réponses inconclusives (ou approximation). Les chercheurs du domaine de l'analyse statique ont développé des outils qui, en ne sur approximations, tend et abandonnent cependant limiter la prolifération des atomes, car il faut bien que le nombre de cas à traiter ne soit pas trop grand. Ces méthodes statiques ont fait, ces dernières maintenant d'approximation performantes pour certaines classes d'erreurs de programmation. Intégrées automatiques, ce qui détecteur dangereuses : des erreurs de pointeurs, des fuites de mémoire, des dans les bornes des tableaux. Cela peut paraître cependant « quelque » ; mais cela empêche le mieux en pas « quelque » ; mais cela empêche le mieux en pratique, des erreurs dangereuses : des erreurs de pointeurs, des fuites de mémoire, des dans les bornes des tableaux dans des vrai dans les systèmes informatiques.

En conclusion, on a d'une part des méthodes statiques qui permettent d'analyser des programmes et des spécifications, surtout qu'ces cas ré résolut ment, mais qu'on résolut ment, mais qu'on résolut ment pour non pas être pertinent. On a d'autre part des méthodes dynamiques qui permettent de tester des systèmes informatiques et de détecter efficacement des dysfonctionnements pour certaines données, mais qui sont incomplètes, tant ce ne les couple avec des hypothèses, appelées hypothèses d'uniformité, qui ne peut être vérifiée par des méthodes statiques ou dynamiques.

**La carte n'est pas le territoire**

Les méthodes dynamiques sont incontournables : elles permettent de tester le système, qu'il s'agisse d'une spécification logique, d'un diagramme ou du texte d'un programme. C'est sous cette physique, que nous allons nous intéresser à ces méthodes. Les tests, utilisés de fait, régulièrement (contrairement au phénomène ainsi), à partir du texte du ou des programmes. Ces opérations complètes peuvent intervenir des erreurs (Cashin et al. 1992; Hyatt 1993) dans des programmes réellement statiquement.

Par ailleurs, on ne peut observer un tel système que par des interactions limitées entre le testeur et le système qui « exécute » ou souvent des données et des résultats partiels. Ceux-ci ne sont pas toujours accessibles quand une défaillance se produit, il n'est souvent qui pratiquement observable ou interpré- tables. Ceci rend difficile le diagnostic de la cause du dysfonctionnement. Il est alors difficile d'analyser et d'exécuter. Alors, comment tester ou un système informatique ?

**Les méthodes de test**

En tête, mentionnons solution ceci ité proposée pour la sélection des tests. On peut les classer selon les informations qui convrent du point de départ à la sélection et selon le standard dont on se réclame à la sélection. Il est ainsi de ceux qui proposent une classification et l'élection en résultant quelques solutions, sans tenter une dramatisation solution. Ceux-ci ne sont pas toujours ceux qui ont le domaine d'entrée du système, celles basées sur le texte du programme, celles basées sur les spécifications ou des modèles.

**Méthodes basées sur le domaine d'entrée**

Ce domaine d'entrée, c'est-à-dire l'ensemble des données acceptables par le système, est en général très nombreux et donc ne peut pas être testé. Il est pourtant nécessaire tout robustesse. Il y a deux grandes approches pour se convrer à ses pis-de-tests bas : le tirage aléatoire et la décomposition en sous-domaines. Ces deux approches peuvent être combinées avec d'autres, comme on le verra plus loin.

Le tirage aléatoire peut se faire selon différentes distributions. Une distribution uniforme est souvent simple à mettre en œuvre mais donne des résultats moins bons que d'autres distributions plus sophistiquées, qui feraient un très grot sous-ensemble de données qui correspondent au cas «normal», et qui ne sont pas toujours intéressantes. Les données prendre à des cas particuliers ayant peu de chances d'être atteintes par cette méthode.

L'autre distribution, fondée sur un profil d'usage, permettrait l'avantage de détecter avec une bonne probabilité les dysfonctionnements les plus susceptibles de se produire quand le système sera en opération, et permettrait une estimation de sa

36 *Rapprochement du CNRS n° 18 printemps 2012*

# Informations sur le fichier

Nom original : A3Bulletin58Femmes_Page_038.jpg
Lien vers le fichier
Extension : image/jpeg
Poids : 0.51 Mo
Dimensions : 1654 x 2339 px

# Comment citer cette page

Site Comité pour l'histoire du CNRS & Projet EMAN (UMR Thalim, CNRS-Sorbonne Nouvelle-ENS).
Consulté le 14/01/2026 sur la plate-forme EMAN :
https://eman-archives.org/ComiteHistoireCNRS/files/show/18506

Copier
Fichier créé par Valérie BurgosFichier créé le 09/10/2024 Dernière modification le 09/10/2024