Introduction à la preuve de programme

Qu'est-ce que la preuve de programme ?

- **Problématique**
  - Programmer sans erreur est une tâche difficile
    - La taille des logiciels, du nombre de personnes impliquées dans leur confection et de leur historique.
    - Pourtant cela est un enjeu fondamental pour les systèmes critiques
      - Dans le domaine du médical, de l'aérospatial, des transports routiers et ferroviaires, du nucléaire...).
    - Elle est aussi utile dans d'autres domaines moins critiques car elle permet de formaliser le cahier des charges d'un programme et de faire une implémentation la plus adéquate que possible
    - La preuve de programme propose des outils semi-automatiques permettant de certifier la correction des programmes.
    - Exemple 1 (Bugs célèbres (http://www.cs.tau.ac.il/~nachumd/horror.html)).
      -

1. La sonde Mars Climate Orbiter s'est écrasée en septembre 1999 à cause d'une « erreur stupide » : des unités erronées dans un programme. Histoire Rapport d'histoire
2. L'Airbus 320 abattu en 1988 par le L'USS Vincennes a été attribué à la production énigmatique et trompeuse affiché par le logiciel de suivi. Histoire Plus

5. Des erreurs dans les logiciels médicaux ont causé des décès. Détails dans B.W. Boehm, « Software and its Impact : A Quantitative Assessment », *Datamation,* 19(5), 48-59 (1973).

Qu'est-ce que la preuve de programme ?

- **Problématique**
  - Programmer sans erreur est une tâche difficile

52. Une ligne de production s'est arrêtée lorsque le L'imprimante laser qui met des dates de péremption sur les produits ne pouvait pas Gérez la date de 2000. *Semaine de l'industrie*, 5 janvier 1998, p. 26.

9. Une erreur dans un programme de conception d'aéronef ont contribué à plusieurs accidents aériens graves. D'après P. Naur et B. Randell, eds., *Génie logiciel : Rapport sur une conférence Avec l'appui du Comité scientifique de l'OTAN,* Bruxelles, de l'OTAN Division des affaires étrangères, 1968, p. 121.

25. Une erreur dans une seule instruction FORTRAN s'est produite dans la perte de la première sonde américaine vers Vénus. De G. J. Myers, *Logiciel Fiabilité : principes et pratique,* p. 25.

- **Définitions**
  - Programme correcte

    Un programme est correct s'il effectue sans se tromper la tâche qui lui est con ée et ce dans tous les cas possibles
  - Spécification d'un programme

    La spécification d'un programme est la description sans ambiguïté de la tâche que doit effectuer un programme et des cas permis

    Dans le domaine du médical, de l'aérospatial, des transports routiers et ferroviaires, du nucléaire...).
- **Caractère de la preuve de logicielle**
  - La spécification et preuve de corrections d'un programme.
    - La spécification oblige à abstraire les propriétés d'un programme.
    - La preuve de la correction du programme vis-à-vis de sa spécification est un problème tout aussi difficile. En effet, une analyse exacte d'un programme est impossible comme le montre le théorème de Rice

Qu'est-ce que la preuve de programme ?

- **Caractère de la preuve de logicielle**
    - Elle est aussi utile dans d'autres domaines moins critiques car elle permet de formaliser le cahier des charges d'un programme et de faire une implémentation la plus adéquate que possible
  - Exemple
    - Le typage est un exemple de propriété approchée que l'on peut prouver sur les programmes. Il permet de prouver que les fonctions sont appliquées à des arguments compatibles.
- **Quelles propriétés prouver ?**
  - Il y a plusieurs types de propriété que l'on veut prouver.
    - Prouver que le programme résout le problème que l'on s'est posé.
    - Prouver qu'il termine sur toutes les entrées.
    - Une série d'erreurs que le programme ne doit pas produire à l'exécution :
      - pas de débordement arithmétique,
      - pas de débordement de tableau,
      - pas de débordement de pile,
      - pas de déréférencement de pointeur null, absence de deadlocks.
- **Quelles méthodes**
  - L'analyse dynamique. Elle est la plus répandue.
    - Elle consiste à exécuter le code ou à le simuler en vue de faire apparaître d'éventuels bugs.
    - La Méthode de tests consiste à comparer le résultat d'un programme avec le résultat attendu.

Qu'est-ce que la preuve de programme ?

- **Quelles méthodes**
  - **L'analyse dynamique**. Elle est la plus répandue.
    - Pour que cette méthode soit efficace, il faut tester les différentes situations possibles.
    - Il existe deux types de tests :
      - les tests fonctionnels qui considèrent le programme comme une boîte noire et ne sont établis qu'à partir de la connaissance de la spécification du programme;
      - les tests structurels qui, à partir de la connaissance du programme, cherchent à exécuter toutes les parties du code.
    - Pour établir un plan de test, il faut
      - énumérer les cas à tester et établir un test par cas.
      - Quand le programme est constitué de plusieurs modules, chacun doit être testé indépendamment avant de tester la globalité dans une série de tests dits d'intégration.
      - Elles ne constituent donc pas en général une preuve de la correction du programme.
  - **L'analyse statique**.
    - Utilisée surtout dans le développement de logiciels critiques (par exemple des systèmes embarqués).
    - Consiste à parcourir le texte du code sans l'exécuter a n de prouver certaines propriétés. Il existe différentes méthodes d'analyse statique
    - Elle consiste à parcourir le texte du code sans l'exécuter a n de prouver certaines propriétés.
    - Il existe différentes méthodes d'analyse statique :

Qu'est-ce que la preuve de programme ?

- **Quelles méthodes**
  - **L'analyse statique.**
    - Le Model Checking part d'une représentation nie du système, une abstraction, et s'en sert pour vérifier les propriétés voulues.
    - L'interprétation abstraite permet de calculer les intervalles dans lesquels les variables évolueront au cours de l'exécution du code. Elle est utilisée dans l'aéronautique.
    - Les méthodes par raffinement comme la méthode B partent de la spécification d'un problème et implémente de façon de plus en plus précise le programme jusqu'à
    - Obtenir un code exécutable. Chaque étape du raffinement est prouvée correcte. Elle est utilisée notamment dans le métro Météor (ligne 14).
    - En logique de Hoare, la spécification d'un programme est vue comme un théorème. Ce formalisme permet alors de prouver cette spécification à l'aide d'un système de déduction.
    - La programmation certifiée repose sur la correspondance entre preuves mathématiques et programmes. À la spécification d'un programme est associée une formule logique (un théorème). À partir d'une preuve de cette formule, on extrait un programme et un certificat de ce programme.

Qu'est-ce que la preuve de programme ?

- **Preuve sur papier ?**
  - Programmes impératifs (Invariants et terminaison)
    - La programmation impérative repose sur l'utilisation de boucles (for ou while) dont il faut démonter l'effet et la terminaison.
      - Pour montrer l'effet d'une boucle sur les variables d'un programme, on a recourt à un invariant de boucle. C'est-à-dire une expression mathématique reliant les variables du programme et qui est vérifiée avant l'entrée dans la boucle et à chaque passage dans celle-ci.
      - Pour démontrer qu'une boucle termine, on utilise la propriété suivante : toute suite décroissante d'entier est finie
  - **Applications**
    - **Exercice 1.** 1. Écrire un programme impératif prenant en entrée un entier n et permettant de calculer la somme des n premiers entiers. 2. Prouver la correction du programme et sa terminaison.
    - **Exercice 2.** On considère le programme Caml suivant :

```
let f n=
    let x= ref 0 and y = ref n in
    while (!y <> 0) do
       x := !x + 3;
       y := !y - 1;
       done;
    !x;;
```

1. Donner une spécification du problème
2. Prouver la correction et la terminaison du programme

Qu'est-ce que la preuve de programme ?

- **Preuve sur papier**
  - **Applications (suite)**
    - **Exercice 3**. On cherche à calculer la somme de deux polynômes représentés par des tableaux. Par exemple, $X^5 + 3X^5 + 5$ est représenté par le tableau 500031.
      1. Écrire une spécification du problème.
      2. Écrire un programme solution.
      3. Prouver la correction du programme par rapport à la spécification du problème.
    - **Exercice 4.** On cherche à déterminer l'élément minimum d'un tableau.
      1. Écrire une spécification du problème.
      2. Écrire un programme solution.
      3. Prouver la correction du programme par rapport à la spécification