



*Cycle Ingénieur – 2<sup>ème</sup> année*  
*Département Informatique*

# Verification and Validation

## Part IV : Proof-based Verification

Burkhart Wolff

Département Informatique  
Université Paris-Sud / Orsay

# Test et/ou Preuve (1)

---

## Le test

- Complicé : chemins infaisables; choix des valeurs significatives; validité du résultat obtenu
- Laborieux: volume des tests; choix des valeurs de test; dépouillement
- Nécessite des outils
- Nécessite d'avoir précisément décrit le résultat attendu
- Quand arrête-t-on de tester : sur quels critères ?

*Testing shows the presence of errors, not their absence...*

# Test et/ou Preuve (2)

---

## La preuve

- Complicé (choix des invariants; précision de la formulation; choix des stratégies de preuve)
- Laborieux (taille des formules; niveau de détail)
- Nécessite des outils (simplification, procédures de décision, stratégies de haut niveau)
- Ce n'est qu'une étape dans le processus de vérification:  
Ce qui compte c'est l'ensemble

logiciel + matériel + support d'exécution

*Prend mal en compte les performances, l'utilisabilité, ...*

# Test et/ou Preuve (3)

---

- Est-on sûr que le système formel est consistant ?
- Correction: vis-à-vis d'une spécification formelle de départ
  - ☞ cette dernière peut être inconsistante
  - ☞ elle peut ne pas représenter ce qu'on voulait exprimer
- Est-on sûr d'avoir prouvé les "bons" théorèmes ? Une formule, c'est dur à lire et à écrire...

Correction ≠ Robustesse

# Test et/ou Preuve (fin)

---

Deux techniques complémentaires !

- ❑ Un continuum : analyse statique / preuve formelle ?
- ❑ Ne pas oublier les techniques de validation
- ❑ Concevoir en fonction de la vérification
  - programmes "faciles" à tester
  - programmes "faciles" à prouver
  - documentation explicite et précise de la conception
    - ☞ garder les choix de conception
    - ☞ spécifier les choix de représentation