



# Vérification formelle en Isabelle(HOL) d'un algorithme calculant les composantes fortement connexes d'un graphe

Vincent Trélat, Stephan Merz

École Nationale Supérieure des Mines de Nancy  
Département Informatique

June 9, 2022

## ① Présentation et intérêt des méthodes formelles

Programmation défensive

Les méthodes formelles

## ② Vérification formelle en Isabelle

Comment garantir le fonctionnement continu d'un logiciel dans des conditions imprévues ?

## Comment garantir le fonctionnement continu d'un logiciel dans des conditions imprévues ?

- Concepts de Software Engineering (design documents, design patterns, etc.)
- Review de code en équipe (GitHub, etc.)
- Séries de tests (unitaires, d'intégration, fonctionnels, etc.)

## Comment garantir le fonctionnement continu d'un logiciel dans des conditions imprévues ?

- Concepts de Software Engineering (design documents, design patterns, etc.)
- Review de code en équipe (GitHub, etc.)
- Séries de tests (unitaires, d'intégration, fonctionnels, etc.)

⇒ Responsabilité du programmeur !

*Probability of human error is considerably higher than that of machine error.*

*Kenneth Appel*

*Probability of human error is considerably higher than that of machine error.*

*Kenneth Appel*

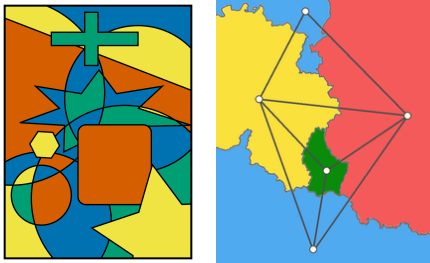


Figure: Illustrations du problème 4-COL

En méthodes formelles, on considère un programme comme une structure mathématique, ce qui permet de raisonner dessus de manière formelle.

- Rigueur
- Concepts de logique
- Sémantique
- Validité par rapport à des spécifications



## Preuve papier vs. preuve formelle

- L'utilisateur fournit la structure de la preuve
- Automatisation des preuves
- Différents outils informatiques : assistants à la preuve (Isabelle(HOL), Coq, Why3, B, etc.) et model checkers (TLC, etc.)

...