INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

centre de recherche
SACLAY - ÎLE-DE-FRANCE

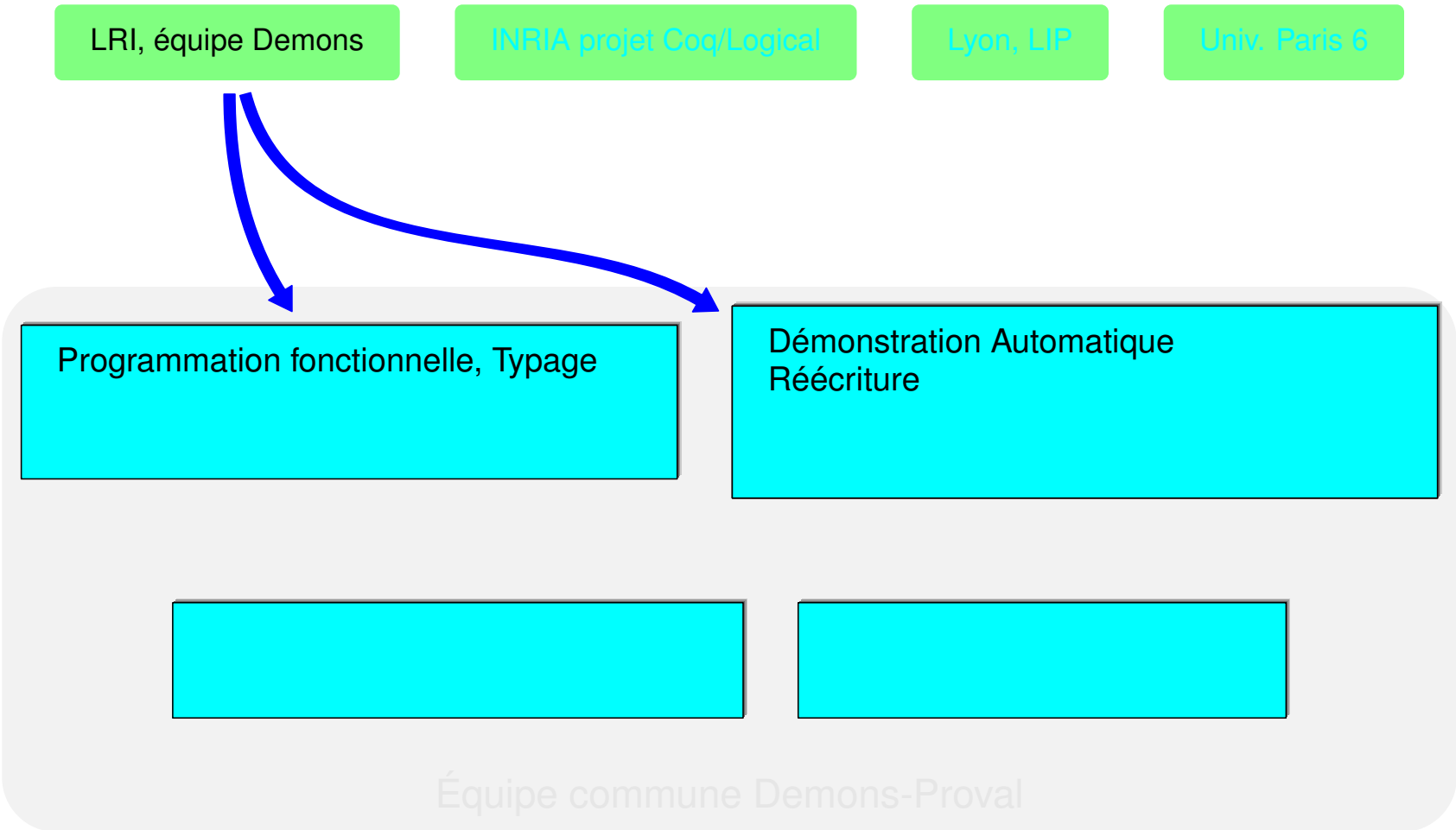
L'équipe ProVal
Responsable : Christine Paulin

un focus sur
la preuve de programmes en virgule flottante

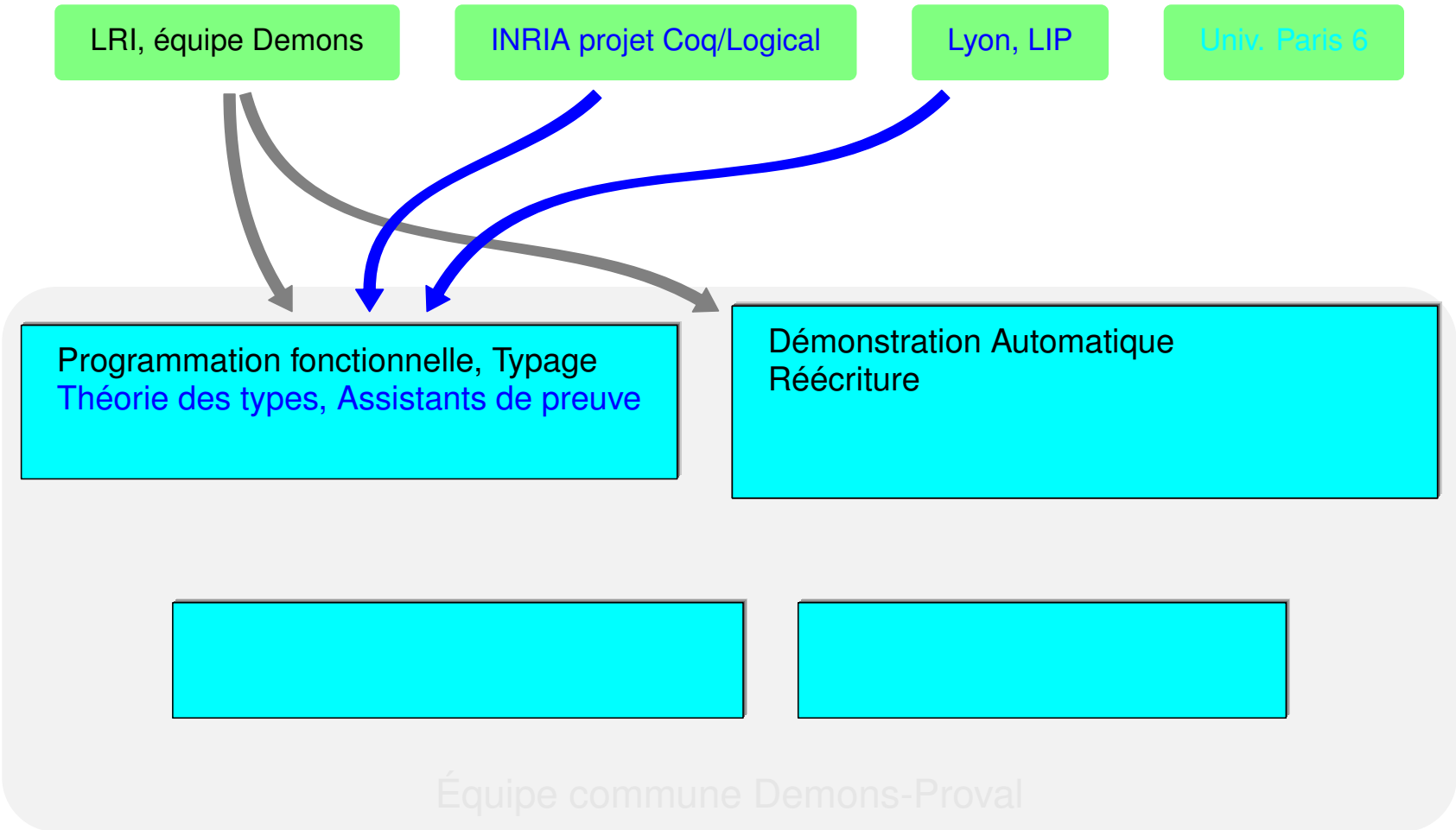
Claude Marché

Accord INRIA Université Paris-Sud 11

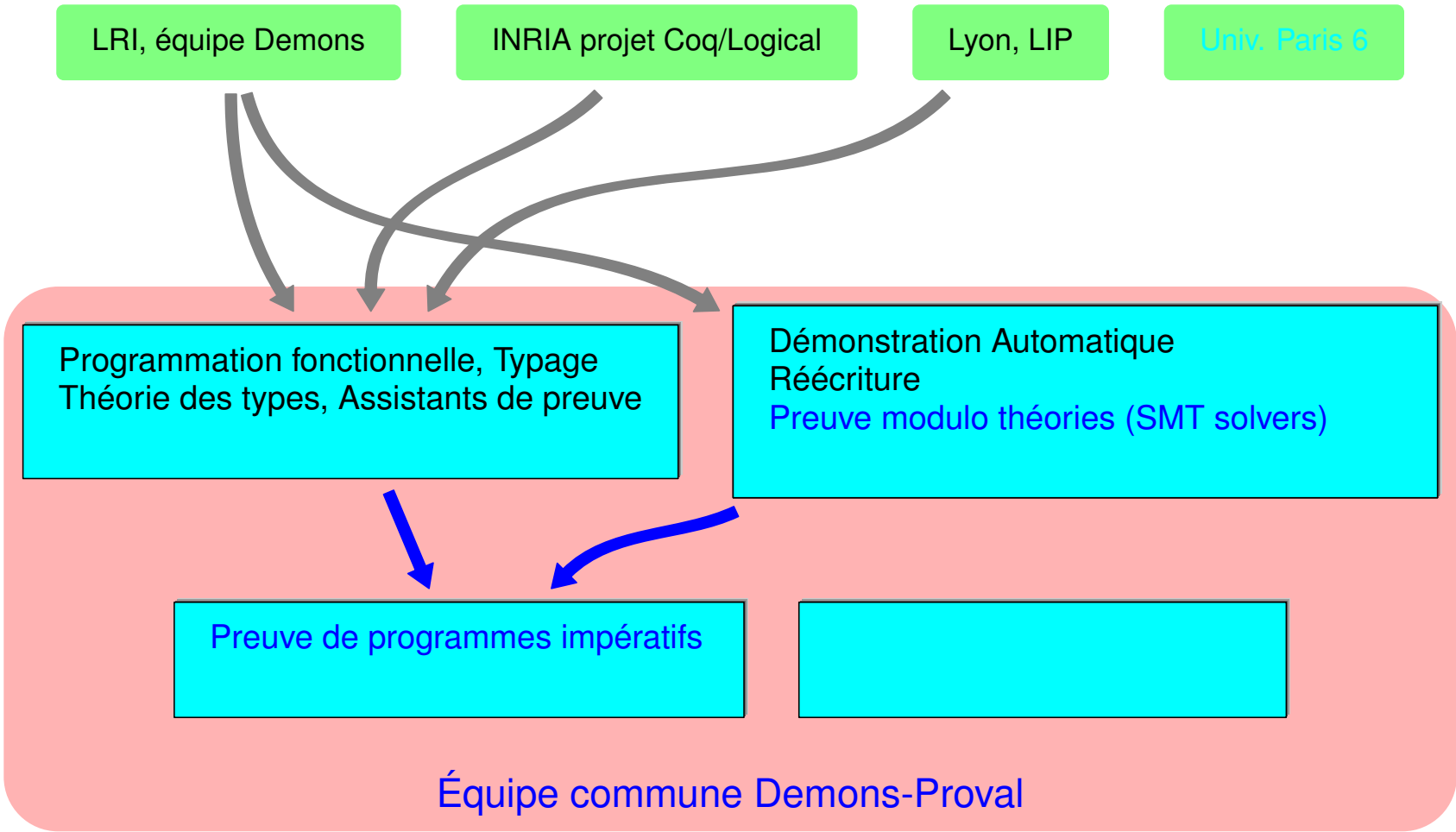
Petit historique, et thèmes de recherche



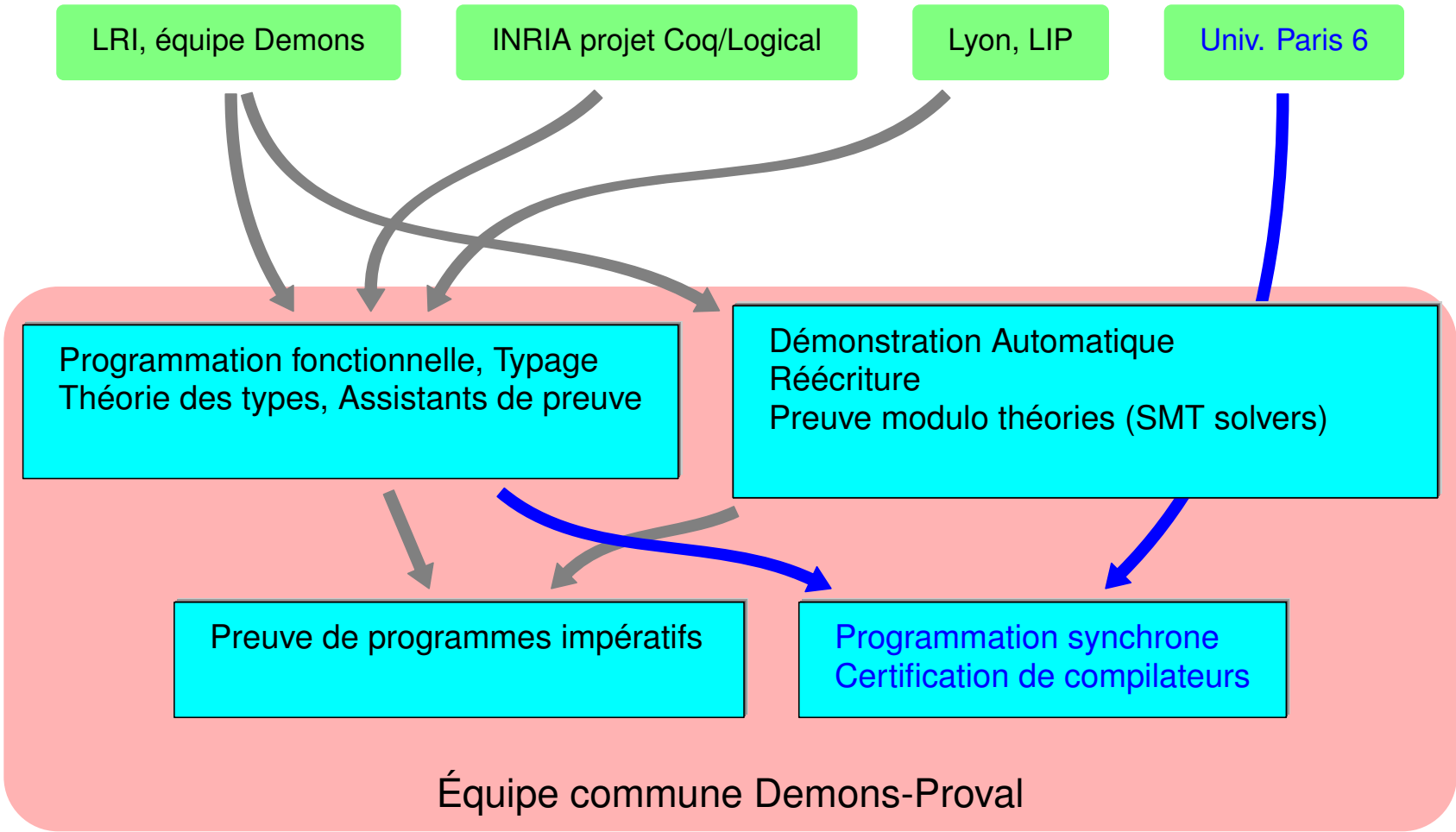
Petit historique, et thèmes de recherche



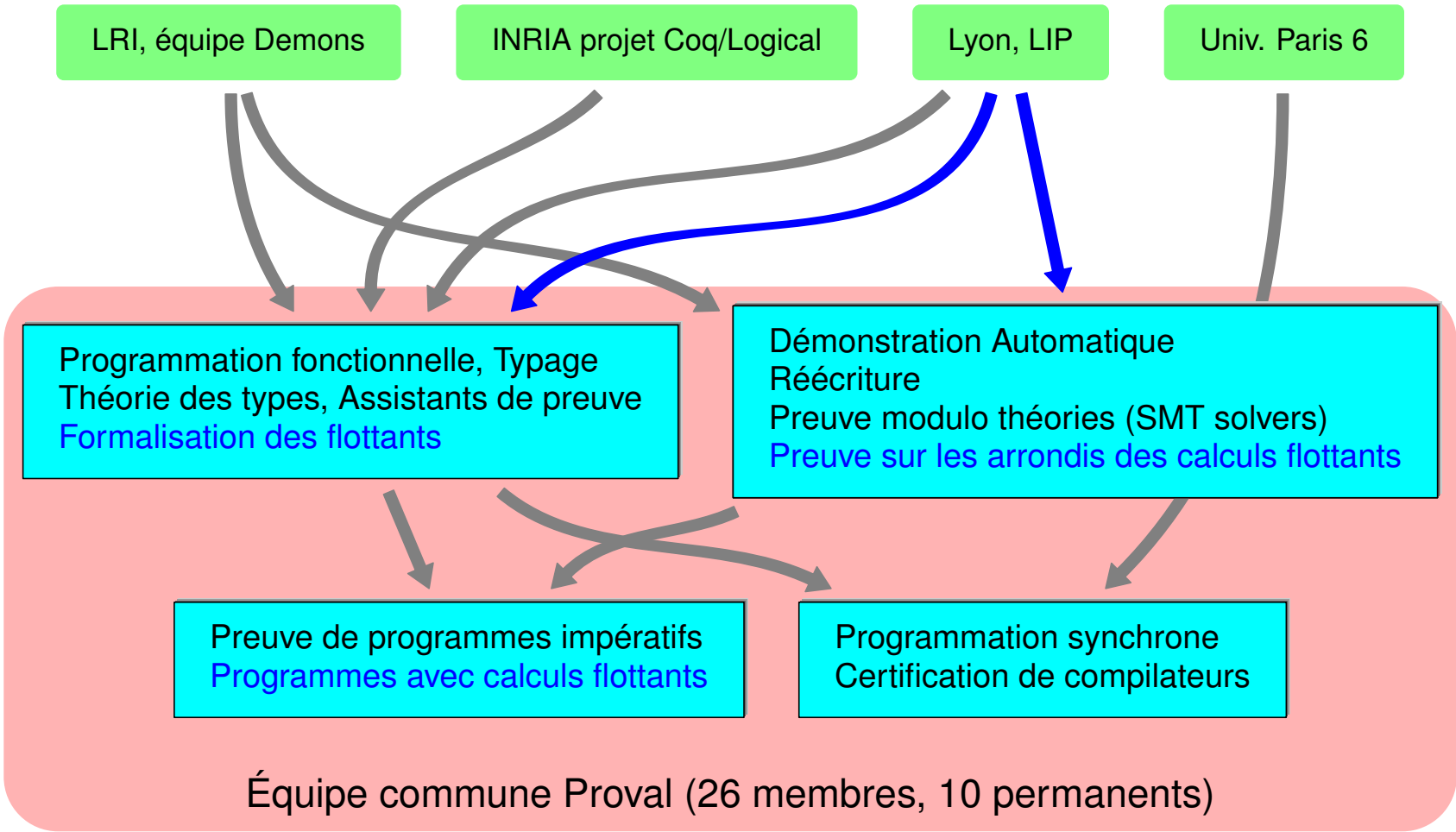
Petit historique, et thèmes de recherche



Petit historique, et thèmes de recherche



Petit historique, et thèmes de recherche



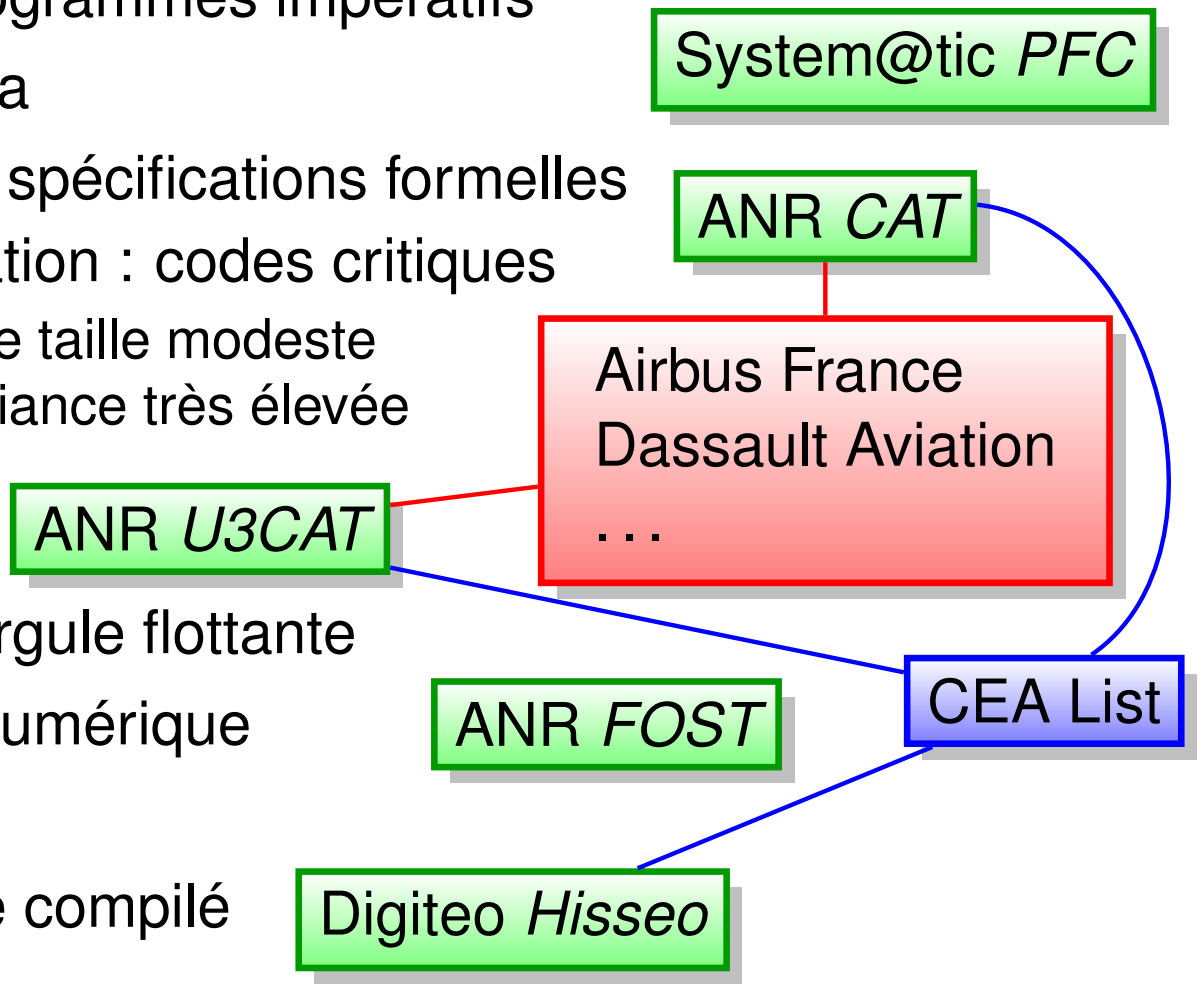
Motivations, contexte

Thème Preuves de programmes impératifs

- langages C et Java
- codes équipés de spécifications formelles
- domaine d'application : codes critiques
  - programmes de taille modeste
  - besoin de confiance très élevée

Thème transversal : virgule flottante

- codes d'analyse numérique
- codes embarqués
- analyse du binaire compilé



Exemple illustratif

Approximation polynomiale d'une function transcendante

- cosinus sur un intervalle autour de zéro

```
float my_cosine(float x) {
  return 1.0f - x * x * 0.5f;
}
```

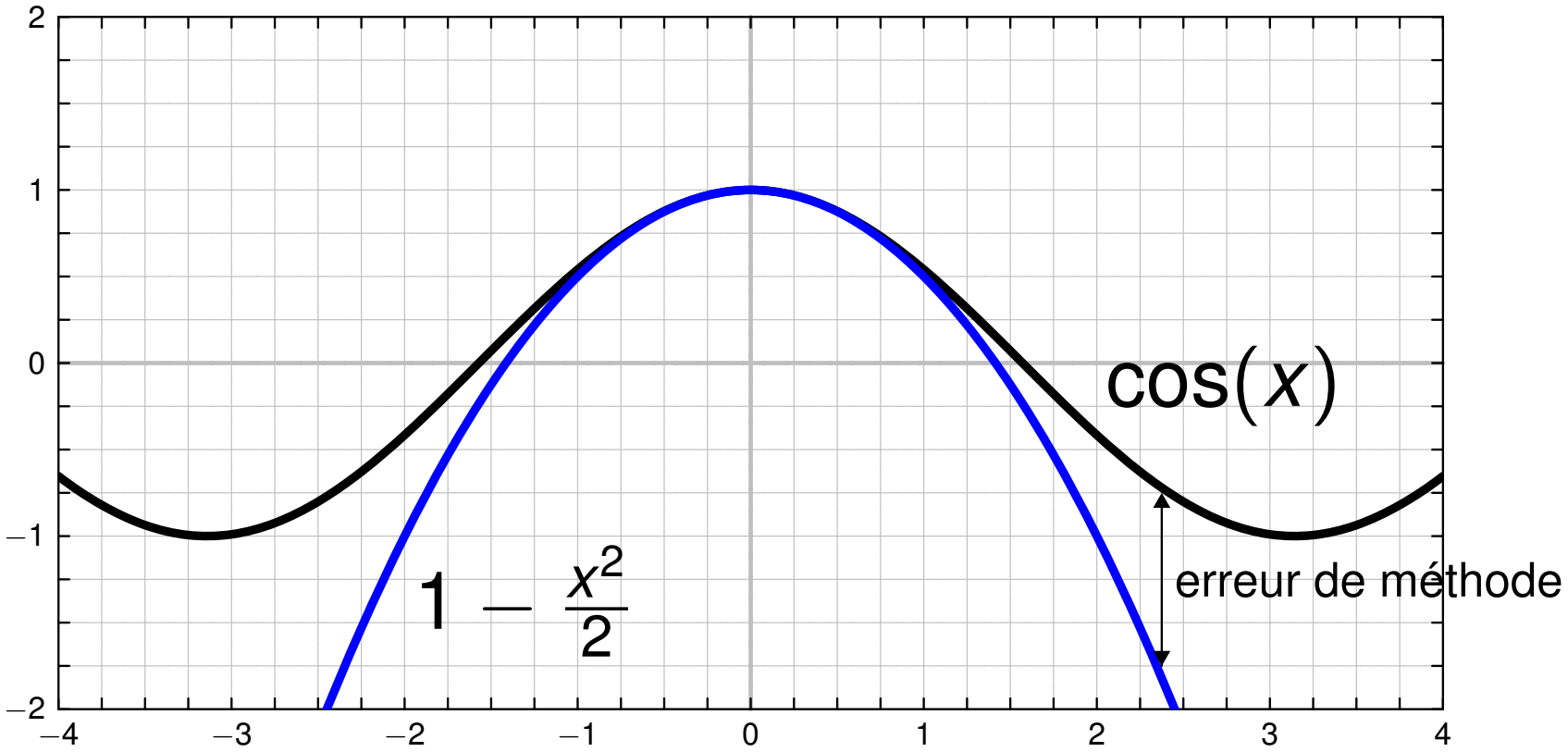
Exemple illustratif

Approximation polynomiale d'une function transcendante

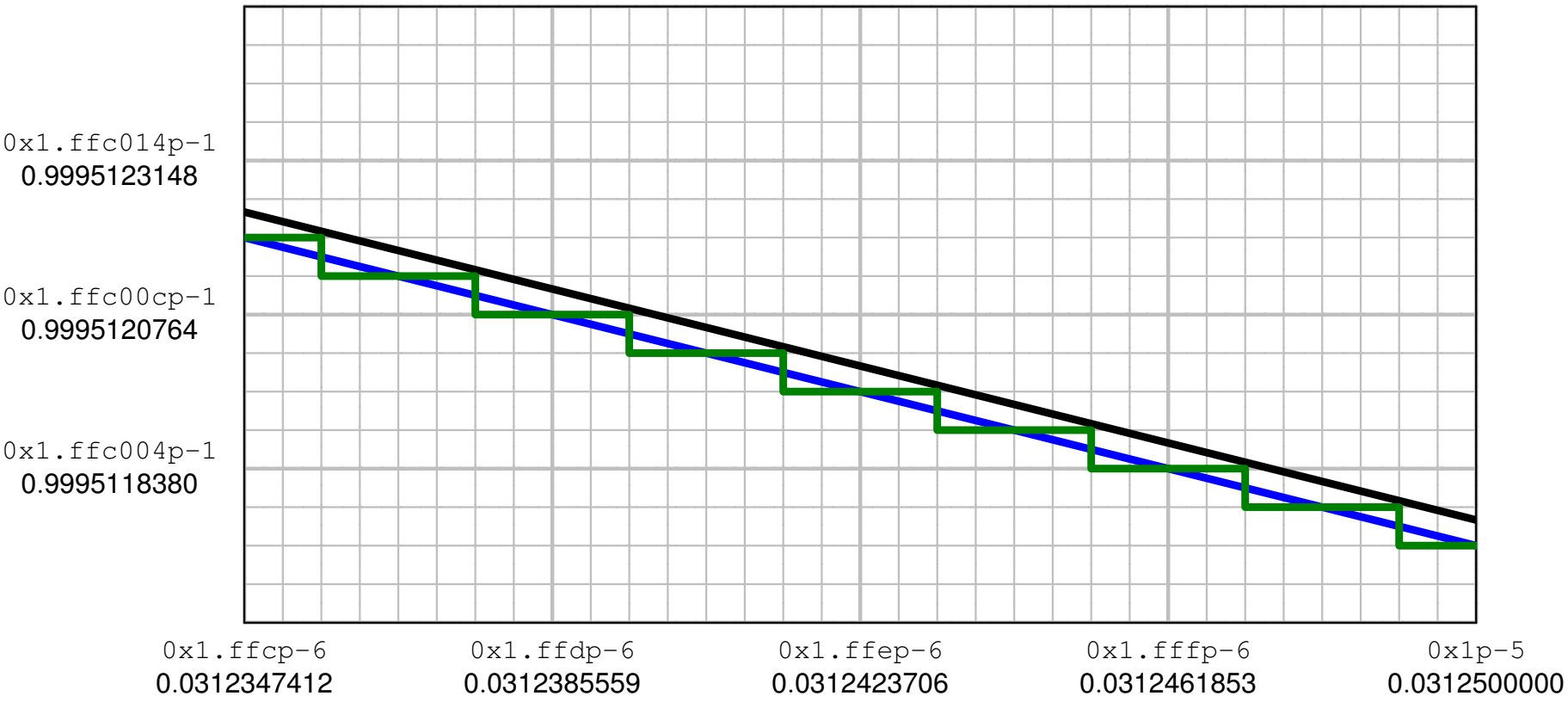
- cosinus sur un intervalle autour de zéro

```
/*@ requires \abs(x) <= ??? ;
  @ ensures \abs(\result - \cos(x)) <= ???;
  @*/
float my_cosine(float x) {
  return 1.0f - x * x * 0.5f;
}
```

Visualisation graphique



Visualisation : focus sur $[1/32 - \varepsilon, 1/32]$



bleu - noir : erreur de méthode

vert - bleu : erreur d'arrondi

vert - noir : erreur totale

Retour sur le code annoté

```
/*@ requires \abs(x) <= 0x1p-5; // x ∈ [-1/32, 1/32]
  @ ensures \abs(\result - \cos(x)) <= 0x1p-23;
  @                // erreur totale
  @*/
float my_cosine(float x) {


  return 1.0f - x * x * 0.5f;
}
```

Retour sur le code annoté

```
/*@ requires \abs(x) <= 0x1p-5; // x ∈ [-1/32, 1/32]
  @ ensures \abs(\result - \cos(x)) <= 0x1p-23;
  @                 // erreur totale
  @*/
float my_cosine(float x) {
  //@ assert \abs(1.0 - x*x*0.5 - \cos(x)) <= 0x1p-24;
                    // erreur de méthode
  return 1.0f - x * x * 0.5f;
}
```

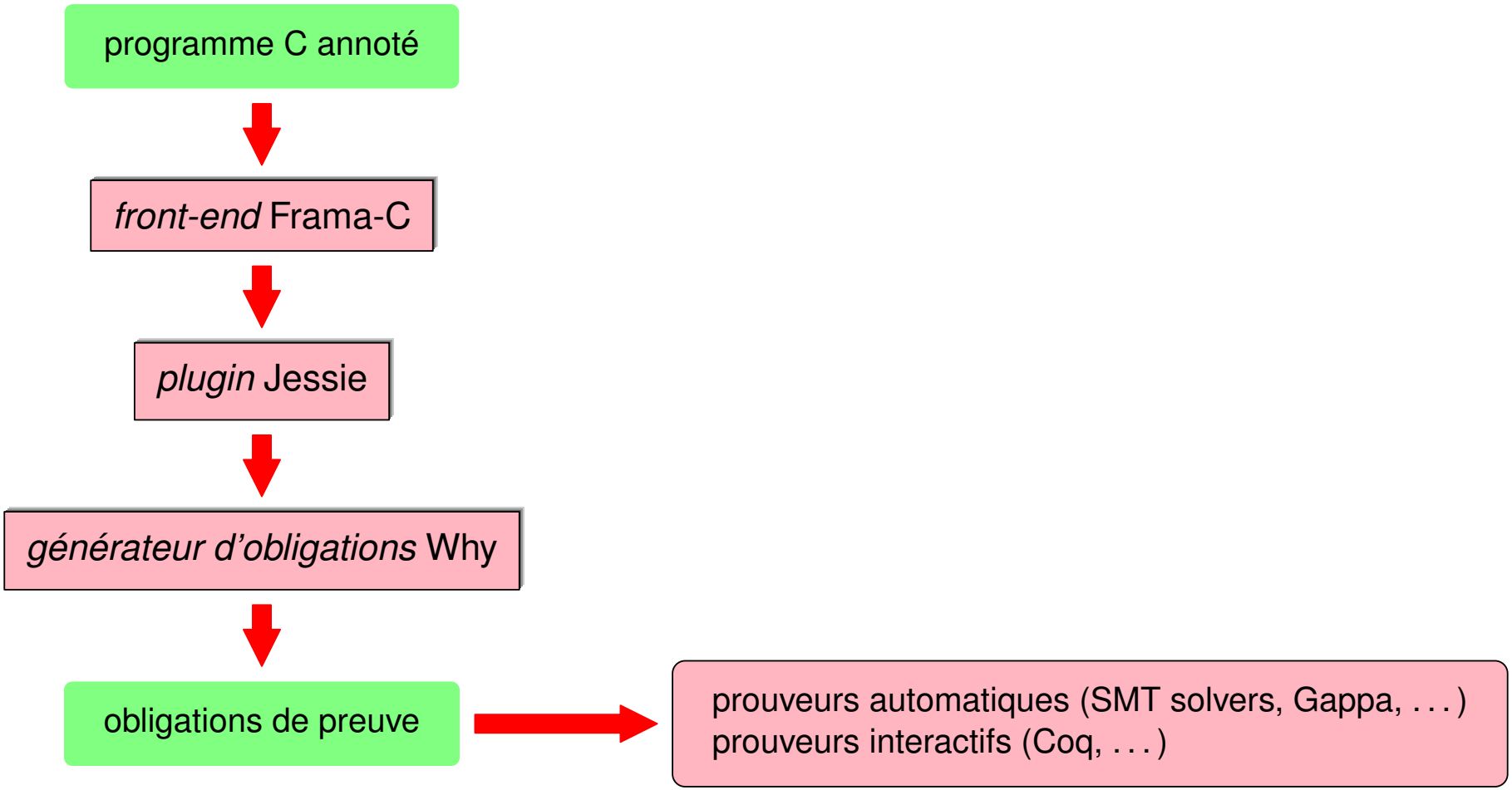
Retour sur le code annoté

```
/*@ requires \abs(x) <= 0x1p-5; // x ∈ [-1/32, 1/32]
  @ ensures \abs(\result - \cos(x)) <= 0x1p-23;
  @                // erreur totale
  @*/
float my_cosine(float x) {
  //@ assert \abs(1.0 - x*x*0.5 - \cos(x)) <= 0x1p-24;
                   // erreur de méthode
  return 1.0f - x * x * 0.5f;
}
```

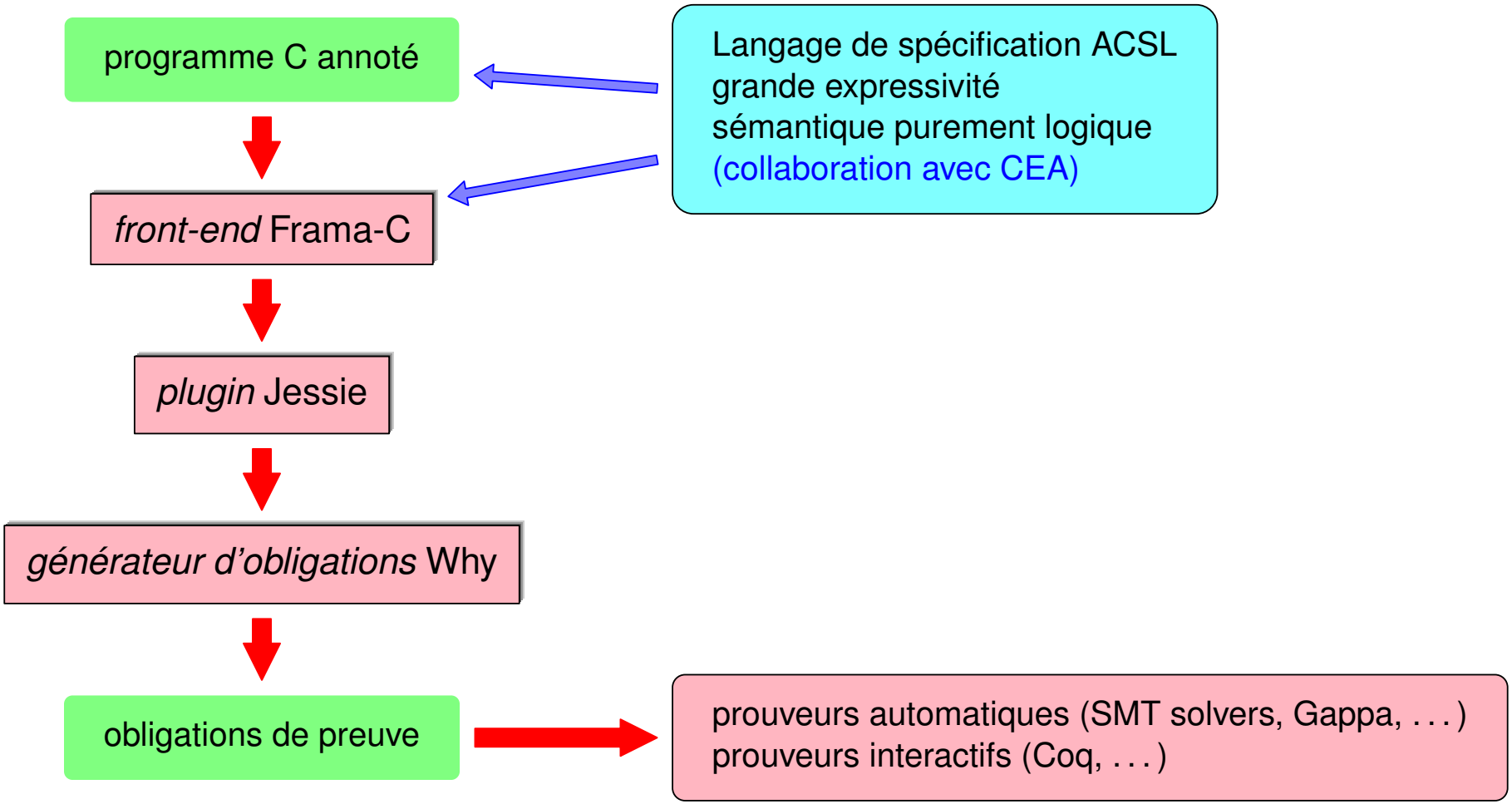
Dans les annotations :

- $+$, $-$, $*$ opèrent sur les nombres réels
- Pas d'arrondi, pas d'overflow dans les annotations

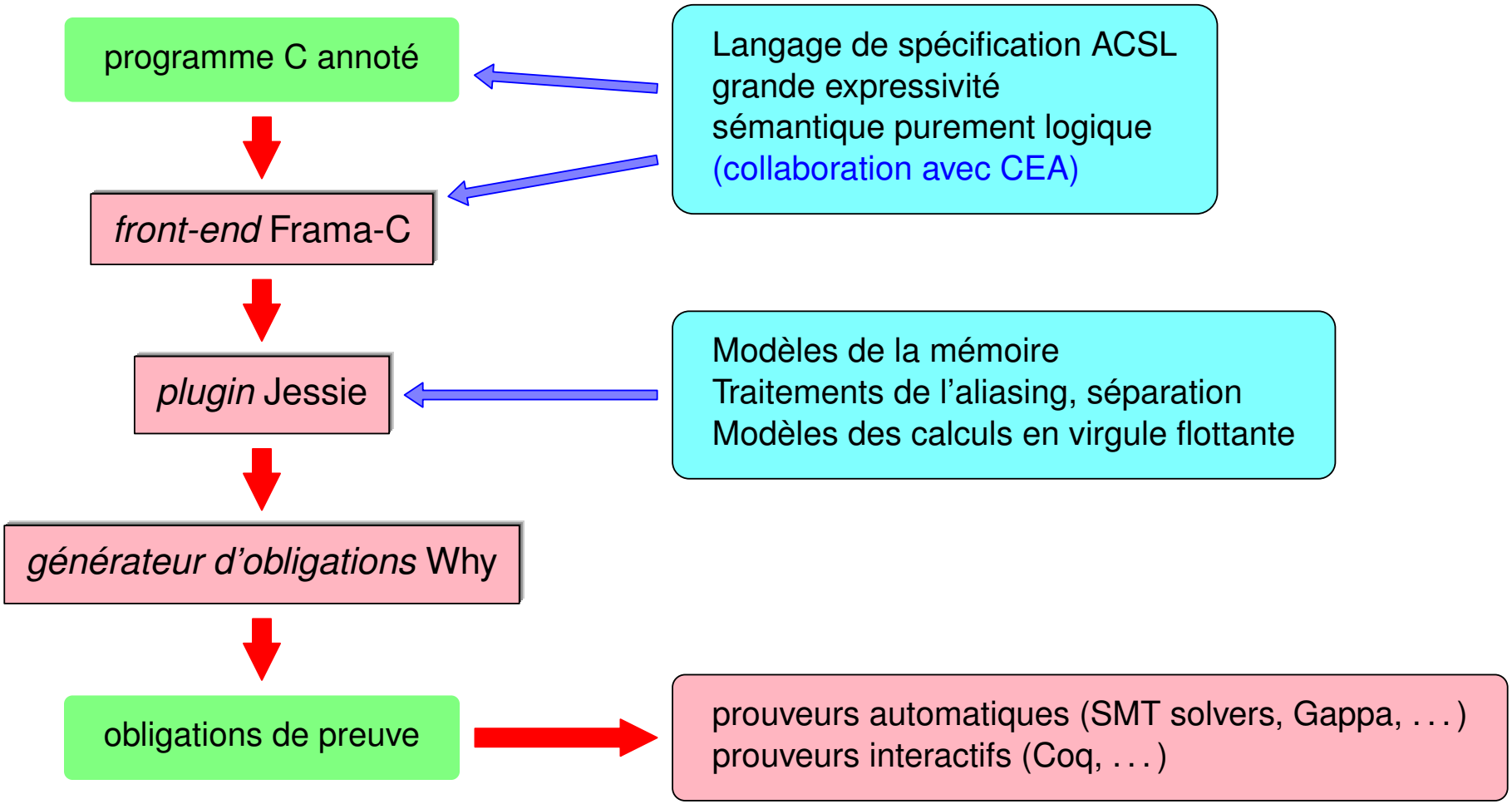
Démarche de preuve



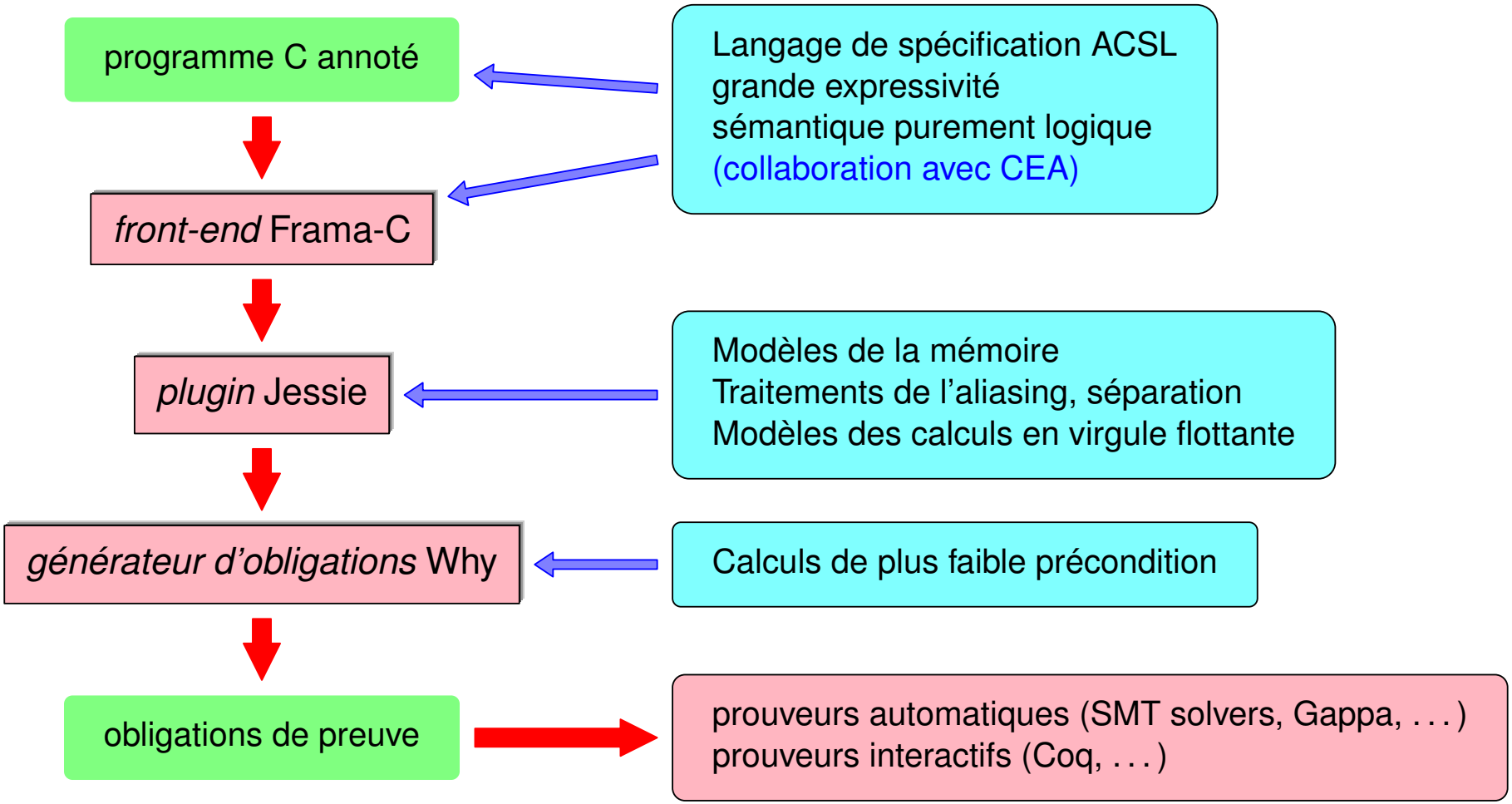
Démarche de preuve



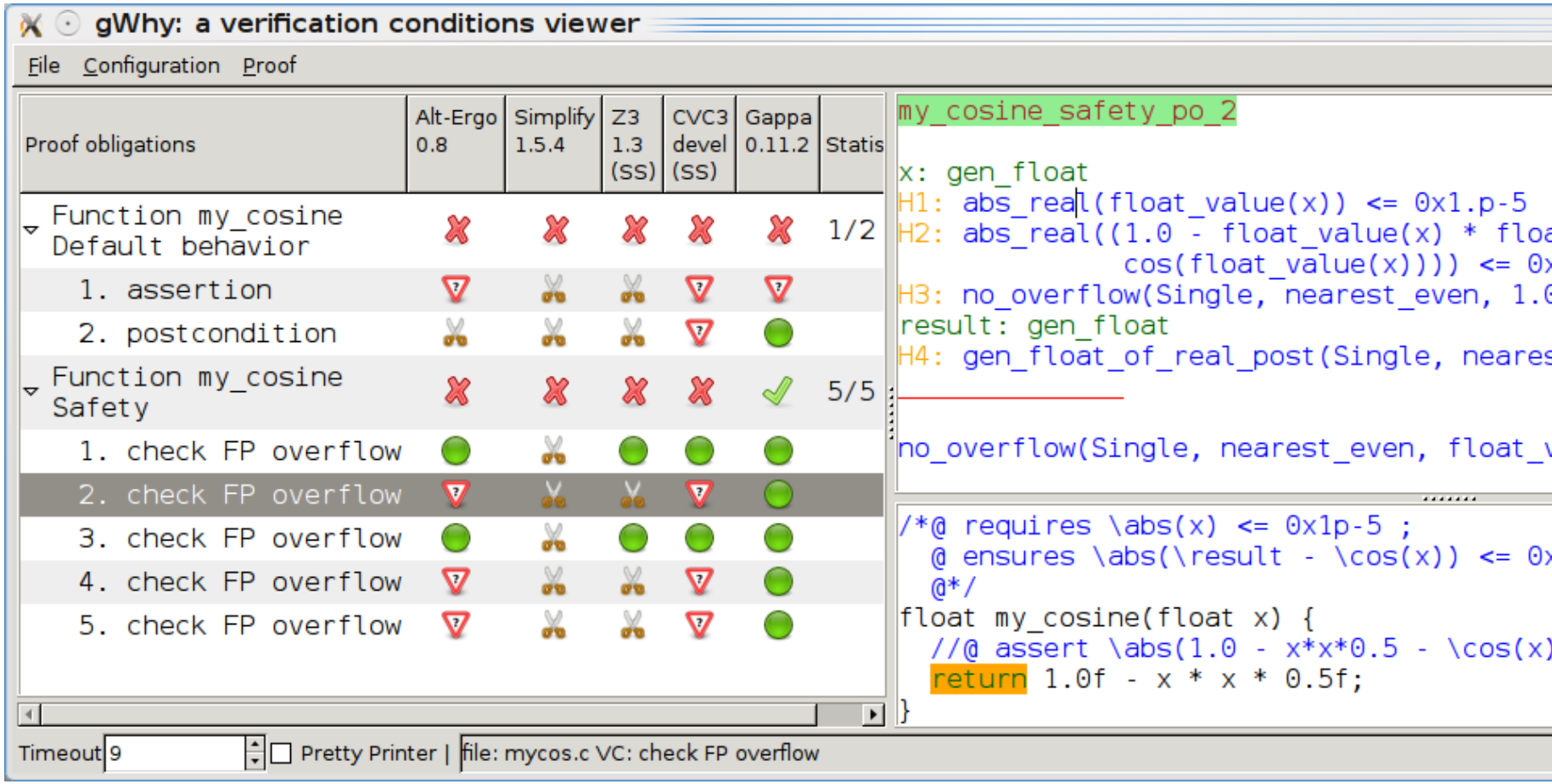
Démarche de preuve



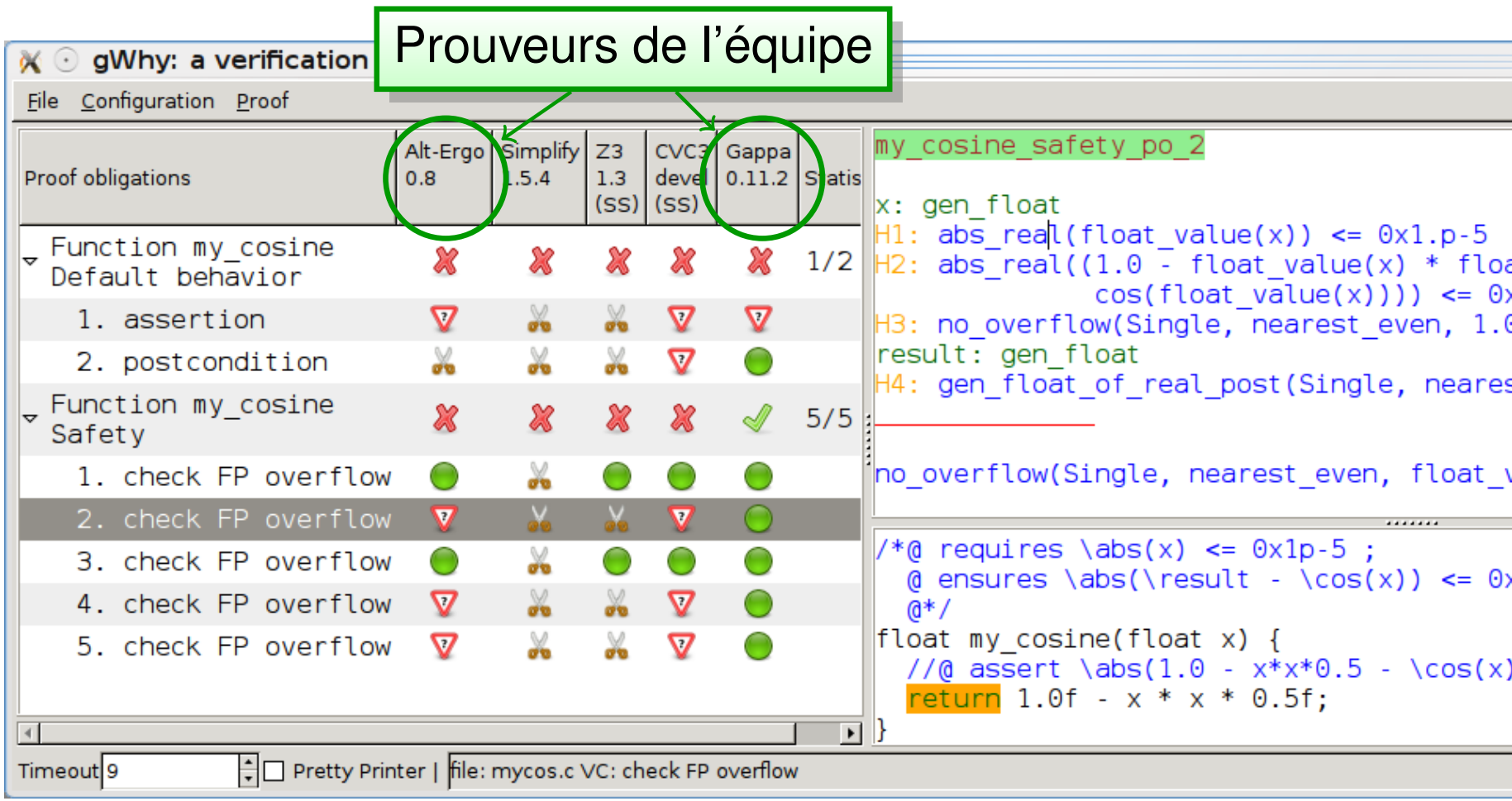
Démarche de preuve



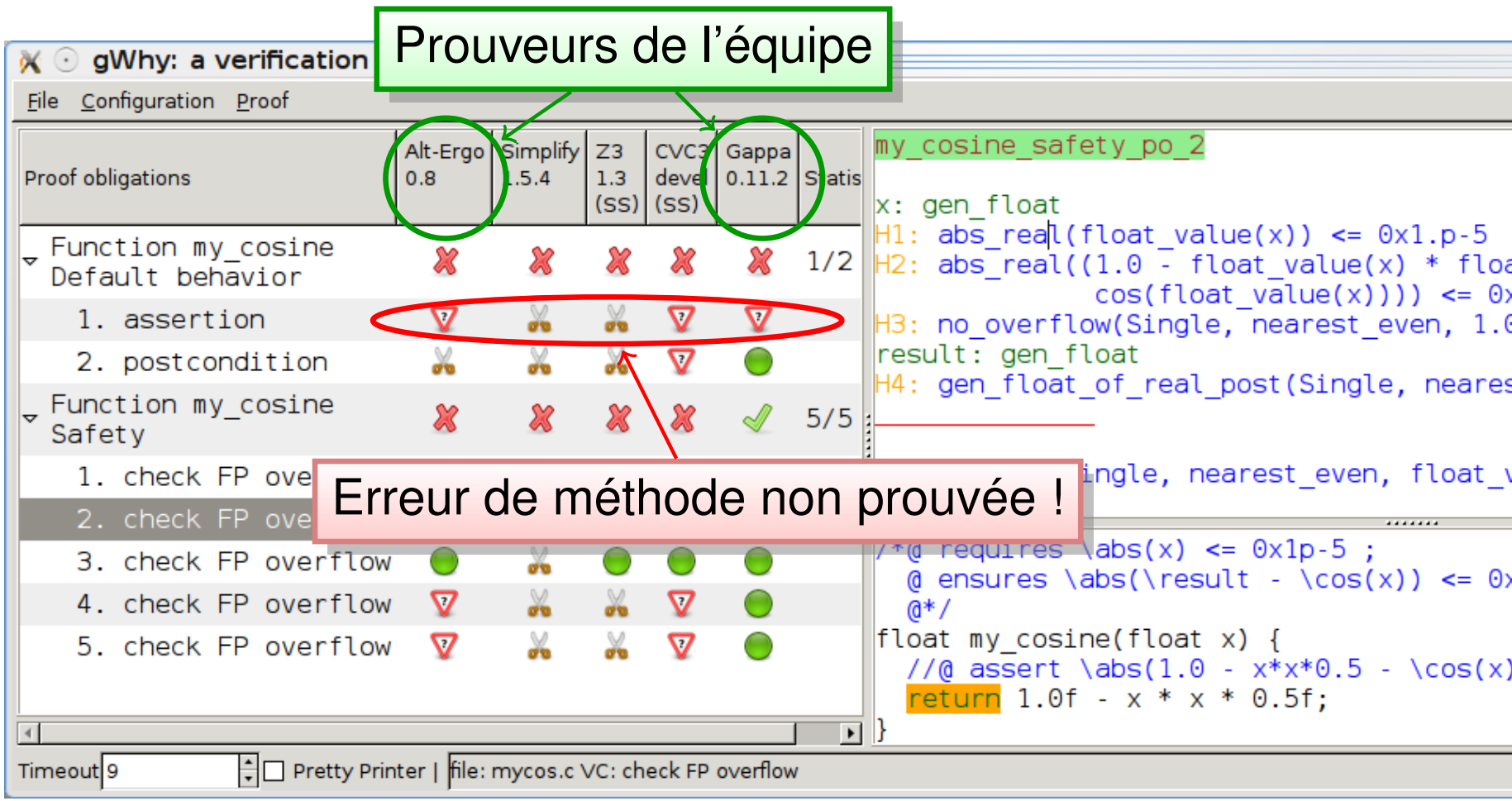
Preuve des obligations



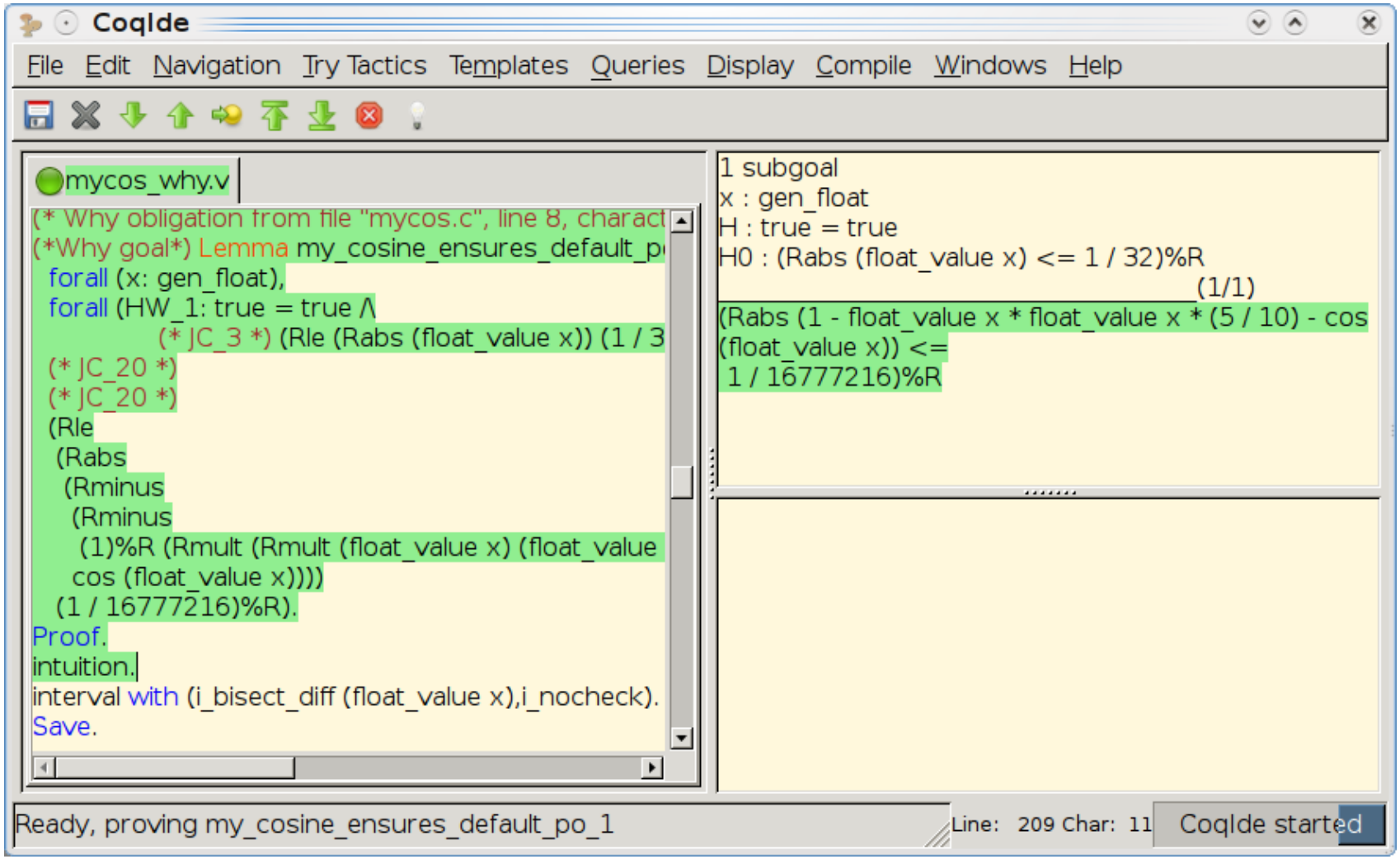
Preuve des obligations



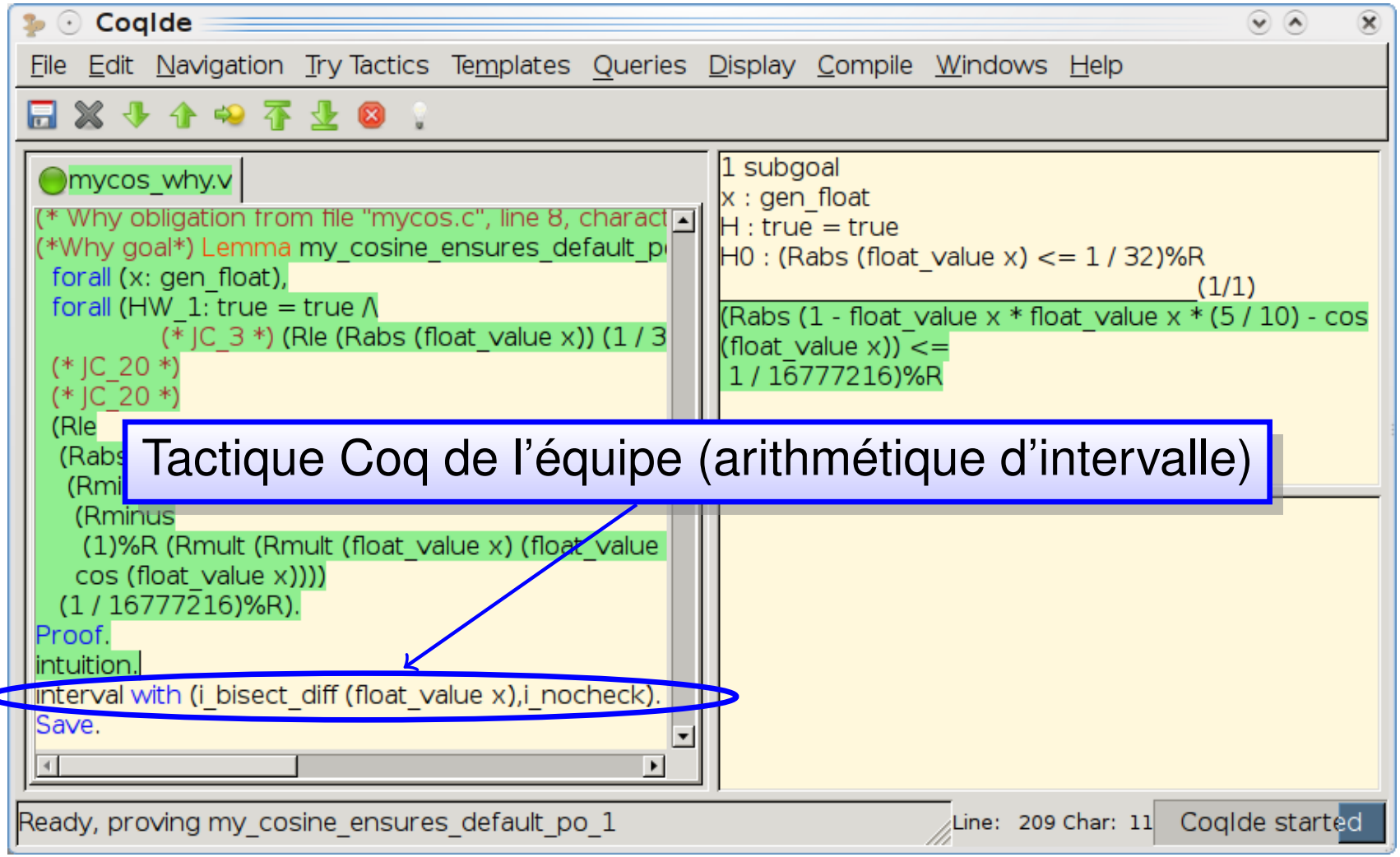
Preuve des obligations



Et l'obligation non prouvée ?



Et l'obligation non prouvée ?



Un élément de perspective

Un axe émergent : Certification des méthodes et des outils

- Qualification DO-178B de Alt-Ergo (Airbus France)
- Production de traces de preuves vérifiables
- Certification formelle
  - Compilateur SCADE pour le synchrone
  - Chaîne Frama-C/Jessie/Why