

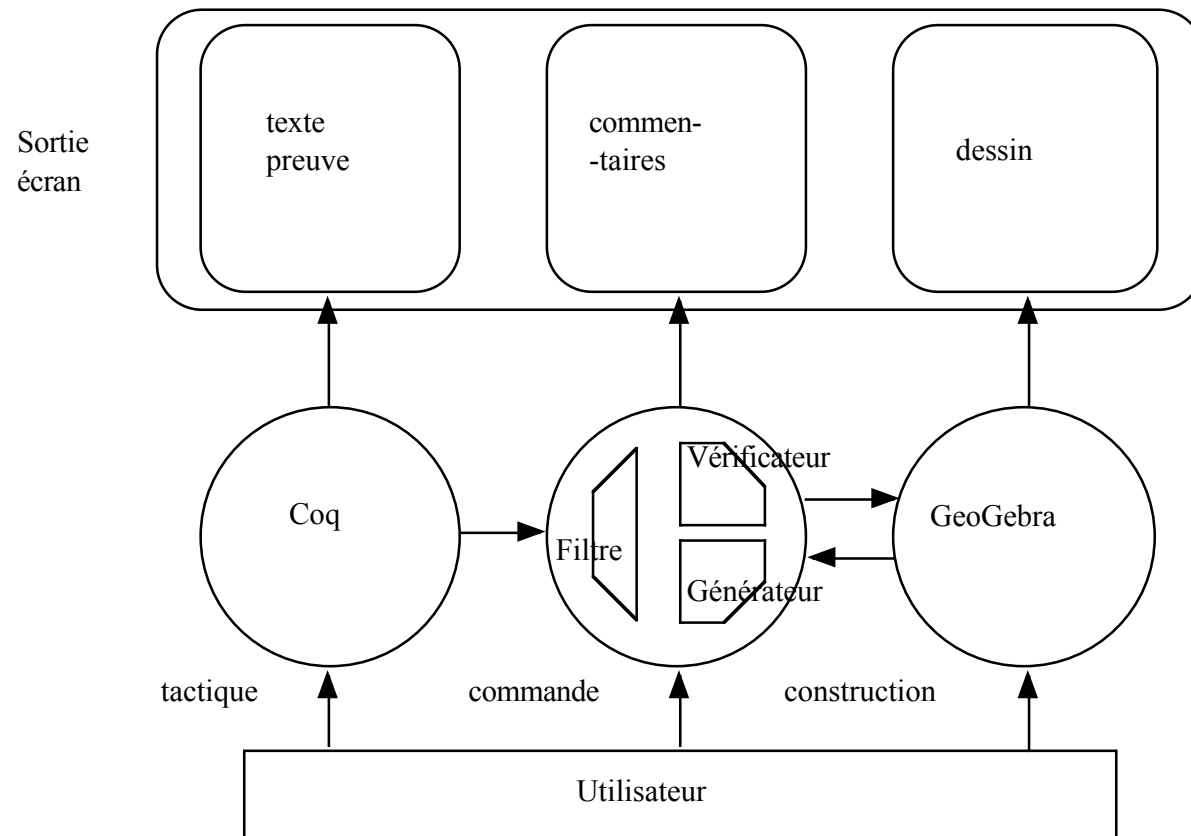
# Interface Coq-Geogebra

Jean Duprat  
Vincent Laporte

# L'objectif.

- Illustrer les démonstrations de géométrie en Coq avec une figure.
- Choix du logiciel de dessin :
  - Geogebra : à cause de son utilisation dans les lycées et de son caractère ouvert.
- 2 étapes :
  - vérification de l'adéquation entre une figure et un environnement de la preuve dans Coq,
  - génération d'une figure.

# Schéma général.



# Le filtre.

- Source :
  - sortie texte de Coq,
- informations :
  - identificateur : type
    - types de base
    - propriétés
  - identificateur := (constructeur paramètres) : type
    - objets construits
- opérations sur les hypothèses :
  - retenue :
    - elle sera alors satisfaite ou non par la figure,
  - non retenue
    - elle sera ignorée et transmise telle quelle sur l'écran

# Hypothèses retenues par le filtre.

- nom : Point (rmq Geogebra interdit x et y).
- nom : Line.
- nom : Circle.
- Hyp : Collinear \_ \_ \_ \_
- Hyp : Clockwise \_ \_ \_ \_
- Hyp : OpenRay \_ \_ \_ \_
- Hyp : ClosedRay \_ \_ \_ \_
- Hyp : Between \_ \_ \_ \_
- Hyp : Segment \_ \_ \_ \_
- Hyp : EquiOriented \_ \_ \_ \_ \_
- Hyp : EquiDirected \_ \_ \_ \_ \_
- Hyp : EquiDistant \_ \_ \_ \_ \_
- Hyp :  $\_ = \_$  (entre points)
  - Remarque : l'hypothèse n'est retenue que si les paramètres sont sous une forme simple (identificateur).

# La logique des propriétés.

- Le Not :
  - utile de filtrer la négation en tête d'une propriété retenue
- Le And :
  - pas très difficile à vérifier, mais aussi facile à contourner en demandant à l'utilisateur de décomposer (destruct) l'hypothèse.
- Le Or :
  - plus compliqué à vérifier, satisfaction sans grand intérêt.
- L'implication :
  - à priori complexe à vérifier.
- Les quantificateurs :
  - souvent très compliqué voire impossible à vérifier.
- Les expressions ayant plusieurs connecteurs :
  - demande un petit calculateur logique,
  - choix qui n'est pas retenu
- Conclusion :
  - SEUL LE NOT EN TETE EST RETENU.

# Aspect incrémental du filtre.

- La liste des objets reconnus par le filtre n'est pas exhaustive.
- Possibilité d'ajouter de nouvelles notions.
  - Difficulté : le traitement à appliquer pour la vérification (puis éventuellement pour la construction dans le générateur).
  - Nécessité d'un format ou d'un petit langage.

# Le vérificateur.

- En entrée :
  - la liste des hypothèses à vérifier,
  - un dessin GeoGebra réalisé par l'utilisateur.
- En sortie :
  - l'environnement colorié :
    - vert : hypothèse retenue satisfaite par la figure,
    - rouge : hypothèse retenue non satisfaite par la figure
    - noir : hypothèse ignorée.

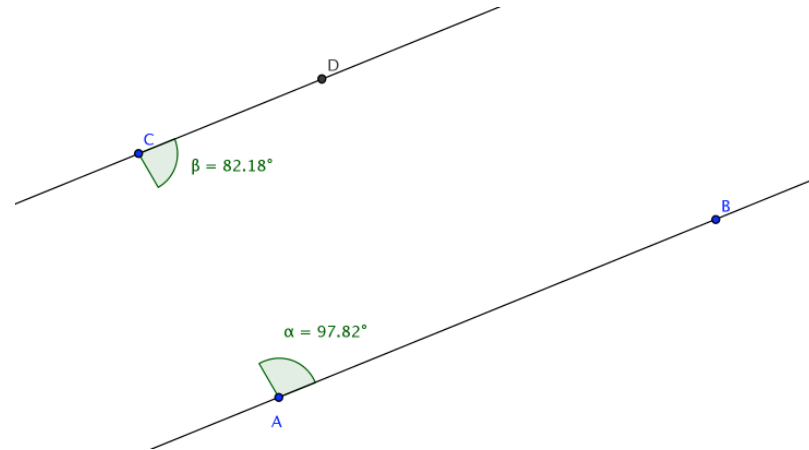


# La méthode.

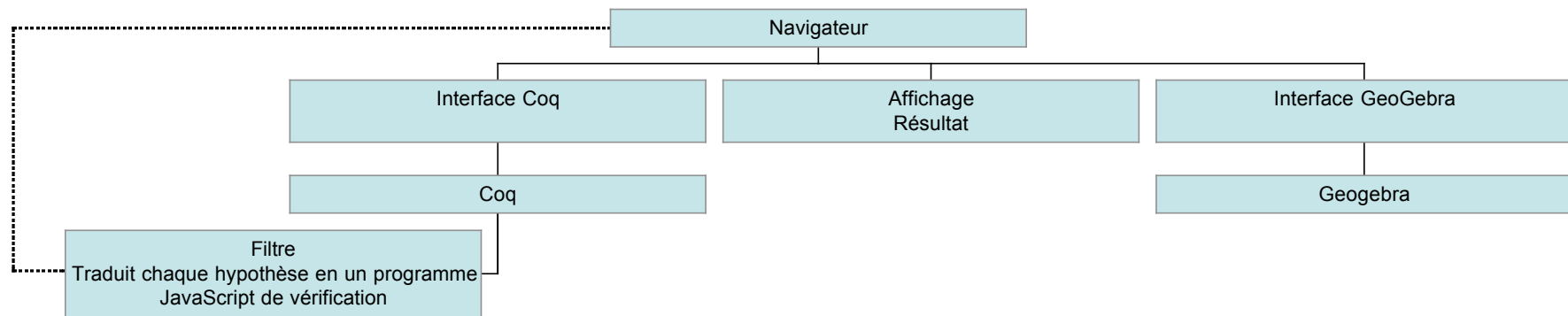
- Première idée :
  - récupérer les coordonnées des points et faire du calcul.
    - Difficultés : calcul en flottant, gestion des arrondis, formules complexes, ...
- Seconde idée :
  - utiliser les capacités de GeoGebra.
    - Nécessité d'adresser des requêtes :
      - Exemple : donner la valeur d'un angle (pour Clockwise)
    - Chaque type retenu doit être traduit en un programme de dialogue avec GeoGebra

# Exemple : EquiOriented A B C D.

```
function areEquiOriented(A,B,C,D) {  
  if (areEqualPoints(A,B))  
    return true ;  
  
  if (areEqualPoints(C,D))  
    return false ;  
  
  if (areEqualPoints(A,C))  
    return isNaN(getAngle(B,A,D)) ;  
  
  return areEqualNum(PI,  
    getAngle(B,A,C) +  
    getAngle(A,C,D)) ;  
}
```



# L'implantation.



Souci : les droits limités de l'Interface Coq (applet Java) empêchent l'exécution de Coq, il faut donc « authentifier » le programme.

# Résultat

Echier Édition Affichage Historique Marque-pages Outils Aide

file:///home/laporte/html/main.htm

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

itWorksWell.png (Image PNG, ...) Geogebra embarqué, v.2

---

Register

TriangleSpec AB A B A B

subgoal 2 is:

Point

1 subgoal

A : Point  
 B : Point  
 H : A <> B  
 c1 := Compass A A B : Circle  
 c2 := Compass B A B : Circle  
 C : Point  
 Hoc : EquiDistant A C A B  
 Hoc0 : EquiDistant B C A B  
 Hck : ~ Clockwise B C A  
 Hun : Unicity C  
 (fun M : Point =>  
   EquiDistant A M A B /\ EquiDistant B M A B /\  
   ~ Clockwise B M A)

=====

Point

1 subgoal

A : Point  
 B : Point  
 H : A <> B  
 c1 := Compass A A B : Circle  
 c2 := Compass B A B : Circle  
 C : Point  
 Hoc : EquiDistant A C A B  
 Hoc0 : EquiDistant B C A B  
 Hck : ~ Clockwise B C A  
 Hun : Unicity C  
 (fun M : Point =>  
   EquiDistant A M A B /\ EquiDistant B M A B /\  
   ~ Clockwise B M A)

=====

Point

Fichier Editer Affichage Options Outils Aide

Objets libres

- A = (-2.66, 4.5)
- B = (-1.6, 3.3)

Objets dépendants

- C = (-3.17, 2.98)
- D = (-1.09, 4.82)
- c1 :  $(x + 2.66)^2 + (y - 4.5)^2 = 2.56$
- c2 :  $(x + 1.6)^2 + (y - 3.3)^2 = 2.56$

Objets auxiliaires

---

Applet geogebra.GeoGebraApplet started

# Le générateur.

- Futur.
- Ligne directrice :
  - utiliser au mieux les fonctionnalités de GeoGebra :
    - création de points par construction :
      - utilisation de l'hypothèse sous forme :  
Identificateur := construction : type
      - traduction de la construction

# Problèmes identifiés.

- Gérer les degrés de libertés :
  - 2 niveaux : libre / contraint
  - 3 niveaux : libre / appartenant à une figure / appartenant à 2 figures.
- Gérer les environnements incohérents :
  - démonstration par l'absurde
  - choix des hypothèses non vérifiées.
- Gérer la complexité :
  - contraintes trop nombreuses pour toutes être satisfaites automatiquement
  - possibilité d'intervention de l'utilisateur sur la figure.