Contributions au développement des méthodes formelles de preuves et applications à la géométrie

Habilitation à diriger des recherches

Nicolas Magaud

Pôle API - Université de Strasbourg - A302 - 6 novembre 2020

Structure de la présentation

Contexte scientifique

- Domaines de recherche :
  - Preuves formelles en Coq
  - Géométrie (algorithmique, combinatoire, calcul numérique)
  - interactions / intersections entre ces deux domaines
- Etude et formalisation en Coq de trois problèmes
  - Modélisation géométrique à base topologique
  - Automatisation des preuves en géométrie projective
  - Calcul réel exact pour la géométrie

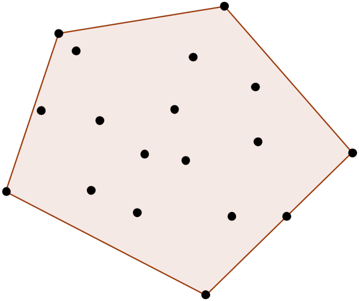
Encadrement et collaborations

- Encadrement de doctorants
  - Christophe Brun (thèse soutenue en 2010)
  - David Braun (thèse soutenue en 2019)
- Collaborations
  - ANR Galapagos (2007-2011) - porteur local Strasbourg
  - collaboration avec l'équipe MIV/Images de ICube
  - collaboration avec l'Université de Poitiers

Structure de la présentation

Enveloppe convexe

- distinction topologie / géométrie
- structure topologique :
  utilisation des cartes combinatoires
- aspects géométriques :
  axiomes de Knuth

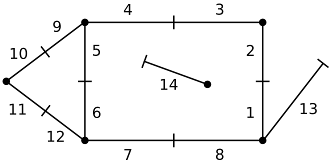


- preuves formelles de deux variantes de la fonction d'insertion d'un point dans une enveloppe déjà construite

Topologie : hypercartes

Définition (Hypercarte)

(1) Une *hypercarte* (en dimension 2) est une structure algébrique $M = (D, \alpha_0, \alpha_1)$, où D est un ensemble fini, dont les éléments s'appellent des *brins*, et où α_0 et α_1 sont des permutations sur D.
(2) Quand α_0 est une involution sur D, M est appelé une *carte combinatoire orientée*.



D	1	2	3	4	5	6	7	8	9	10	11	12	13	14
a0	2	1	4	3	6	5	8	7	10	9	12	11	13	14
a1	8	3	2	9	4	12	6	13	5	11	10	7	1	14

Topologie : modélisation des hypercartes en Coq

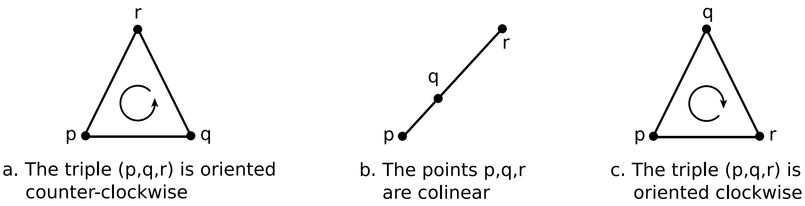
- Les cartes libres inductivement en Coq

```
Inductive fmap : Set :=
  V : fmap
| I : fmap -> dart -> point -> fmap
| L : fmap -> dim -> dart -> dart -> fmap.
```

- Préconditions nécessaires
  - pour l'insertion de *x* : `x <> nil /\ ~ exd m x`
  - pour la couture de *x* et *y* à la dimension *k* :
    `exd m x /\ exd m y /\ ~ succ m k x /\ ~ pred m k y /\ cA m k x <> y`
- Propriété d'invariance pour les cartes `inv_hmap`

Géométrie : les axiomes de Knuth

- le prédicat d'orientation $ccw(p, q, r)$



- les axiomes de Knuth
  - Hypothèse que les points sont en position générale
- **But : abstraire les questions de précisions des calculs**

Géométrie : les axiomes de Knuth

- 6 axiomes, qui capturent les propriétés de ce prédicat

  P.1 (cyclicité) : $ccw(p, q, r) \Rightarrow ccw(q, r, p)$.

  P.2 (symétrie) : $ccw(p, q, r) \Rightarrow \neg ccw(p, r, q)$.

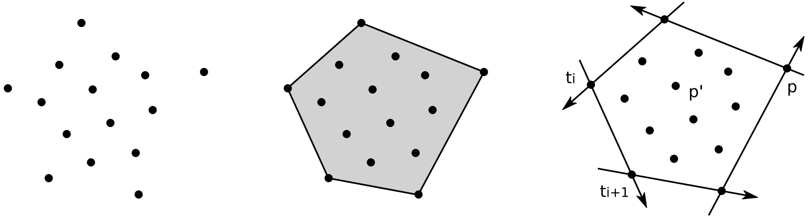
  P.3 (non-dégénérescence) :
  $\neg collinear(p, q, r) \Rightarrow ccw(p, q, r) \vee ccw(p, r, q)$.

  P.4 (intériorité) : ...

  P.5 (transitivité) : ...

  P.5 bis (transitivité bis) : ...

Enveloppe convexe : définition



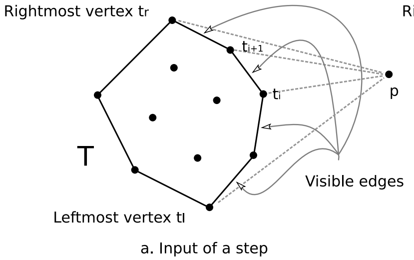
a. A finite set P of points

b. A convex polygon T

c. A convex polygon with its oriented edges

Calcul de l'enveloppe convexe

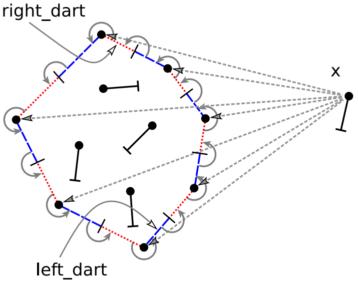
- Algorithme incrémental :
  calcul d'une nouvelle enveloppe convexe T' à partir du polygone convexe T et d'un point p

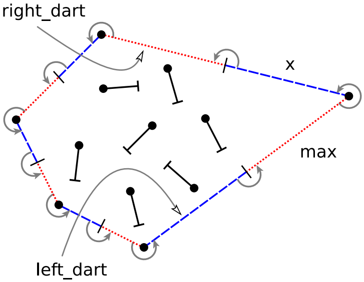


- Deux implantations de l'algorithme incrémental sous forme de fonctions en Coq

Calcul structurel en Coq

- récursion structurelle sur la structure des cartes libres

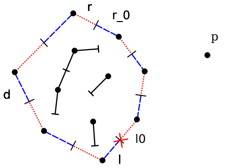




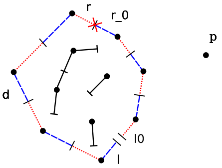
- reconstruction de la nouvelle carte à partir d'une carte vide

Calcul géométrique en Coq

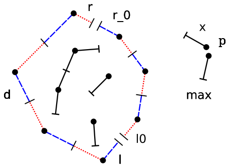
- récursion en suivant la forme de l'enveloppe convexe
  - recherche des extrémités gauche et droite si elles existent
  - introduction d'une mesure pour garantir la terminaison



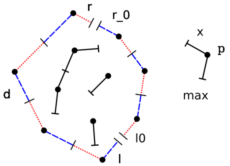
1 : Split m zero l l0



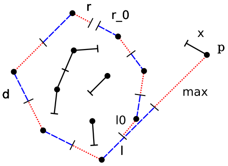
2 : Split m1 zero r_0 r



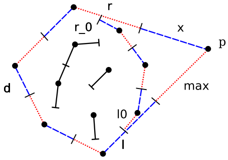
3 : I (I m2 x p) max p



4 : Merge m3 one max x



5 : Merge m4 zero l max



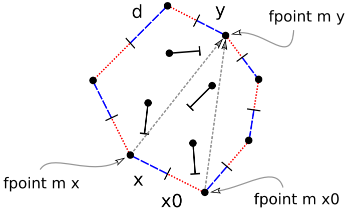
6 : Merge m5 zero x r

- Implantation prenant en compte la géométrie

Propriétés topologiques et géométriques

- Propriétés topologiques
  - préservation de la structure d'hypercarte
  - la carte représentant l'enveloppe convexe est un polygone (et des points isolés)
  - la carte représentant l'enveloppe convexe est planaire

- Propriétés géométriques
  - correction du plongement
  - propriété de convexité



Bilan et perspectives

- Deux programmes de calcul de l'enveloppe convexe
  - implantés fonctionnellement en Coq
  - prouvés formellement en Coq
  - des dizaines de milliers de lignes de Coq dans chaque cas
- Extensions possibles
  - en 3D et plus
  - cas dégénérés : points confondus, alignés.
- Ingénierie de la preuve
  - automatisation partielle des aspects géométriques

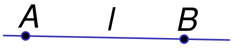
Structure de la présentation

Motivations, contexte et objectifs

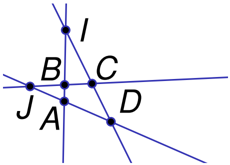
- Motivation :
  - faire des outils d'aide à la preuve en géométrie
- Contexte :
  - une théorie géométrique simple :
    la géométrie d'incidence projective
  - Une approche combinatoire automatisable
- Objectifs
  - un prouveur automatique
  - et un résultat certifié en Coq

Axiomes usuels pour la 3D

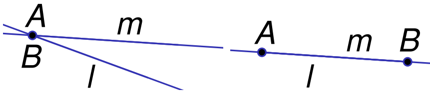
A1P3



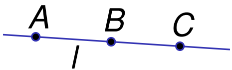
A2P3



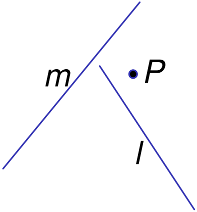
A3P3



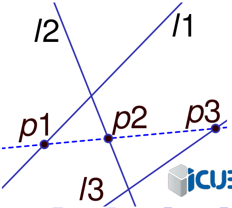
A4P3



A5P3



A6P3



Une approche combinatoire

- Notion de rang (rk) :
  fonction à valeurs entières permettant de capturer la dimension (point, droite, plan, espace tout entier) d'un ensemble fini E de points
- Approche extensible en dimension supérieure à 3,
- plus homogène, mais plus combinatoire
- Propriétés de matroïde de la fonction de calcul du rang
  - (A1R3) Non-Negative and Subcardinal :
    $\forall X \subseteq E, 0 \leq rk(X) \leq |X|$
  - (A2R3) Non-Decreasing :
    $\forall X \subseteq Y, rk(X) \leq rk(Y)$
  - (A3R3) Submodular :
    $\forall X,Y \subseteq E, rk(X \cup Y) + rk(X \cap Y) \leq rk(X) + rk(Y)$

Quelques exemples d'ensembles de points et leurs rangs

rk{A,B} = 1	A = B
rk{A,B} = 2	A $\neq$ B
rk{A,B,C} = 2	A,B,C sont alignés avec au moins 2 points distincts
rk{A,B,C} $\leq$ 2	A,B,C sont alignés
rk{A,B,C} = 3	A,B,C ne sont pas alignés
rk{A,B,C,D} = 3	A,B,C,D sont coplanaires, et ne sont pas alignés
rk{A,B,C,D} = 4	A,B,C,D ne sont pas coplanaires

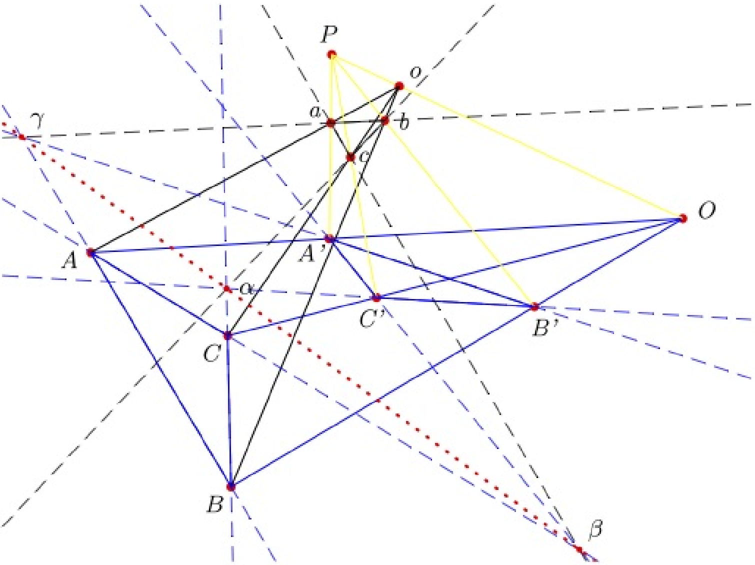
Adaptation des rangs à la géométrie

- (A4R3) Rk-Singleton : $\forall$ P : Point, rk{P} = 1
- (A5R3) Rk-Couple : $\forall$ P Q : Point, P $\neq$ Q $\Rightarrow$ rk{P, Q} = 2
- (A6R3) Rk-Pasch : $\forall$ A B C D : Point, rk{A, B, C, D} $\leq$ 3 $\Rightarrow$ $\exists$ J : Point, rk{A, B, J} = rk{C, D, J} = 2
- (A7R3) Rk-Three-Points : ...
- (A8R3) Rk-Lower-Dimension : $\exists$ A B C D : Point, rk{A, B, C, D} $\geq$ 4
- (A9R3) Rk-Upper-Dimension : ...

Equivalence des représentations

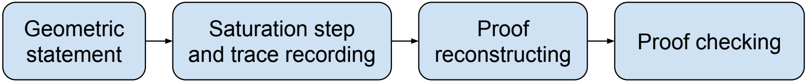
- Le système d'axiomes de l'approche synthétique est équivalent au système basé sur les matroïdes et les rangs.
- Cette équivalence reste vraie en 2D et en dimension supérieure à 3.
- Comparaison des deux approches sur des modèles finis
- Un premier exemple fait à la main : la preuve du théorème de Desargues plongé en 3D.

Théorème de Desargues



Un prouveur automatique

- Un outil basé sur la saturation du contexte
- qui génére un script de preuve Coq vérifiable



- intervalle de rangs pour chaque sous-ensemble
- initialisation des intervalles avec les hypothèses
- réduction d'intervalles

Encodage et règles de réécriture

- Codage de l'ensemble et ses rangs min. et max. sur 32 bits

rkMax = 3

Set of points = {A, B, E, H}

10	01	0000 0000 0000 0000 0000 1001 0011

rkMin = 2

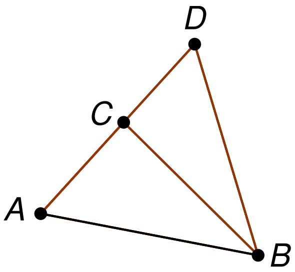
- Transformation des propriétés de *rkMin* et *rkMax* en règles de réécriture :
  la propriété $X \subseteq Y \subseteq E$, $rkMin(Y) \geq rkMin(X)$
  devient la règle
  if $X \subseteq Y$ and $rkMin(X) > rkMin(Y)$
  then $rkMin(Y) \leftarrow rkMin(X)$

Un exemple de preuve automatique

- Enoncé en Coq

```
Lemma example : forall A B C D : Point,
rk(A, B, D) = 3 -> rk(A, C, D) = 2 ->
rk(A, C) = 2 -> rk(C, D) = 2 ->
rk(A, B, C) = 3.
```

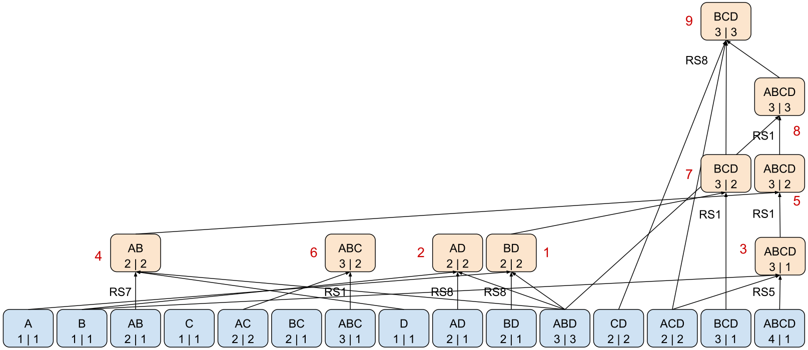
- géométriquement



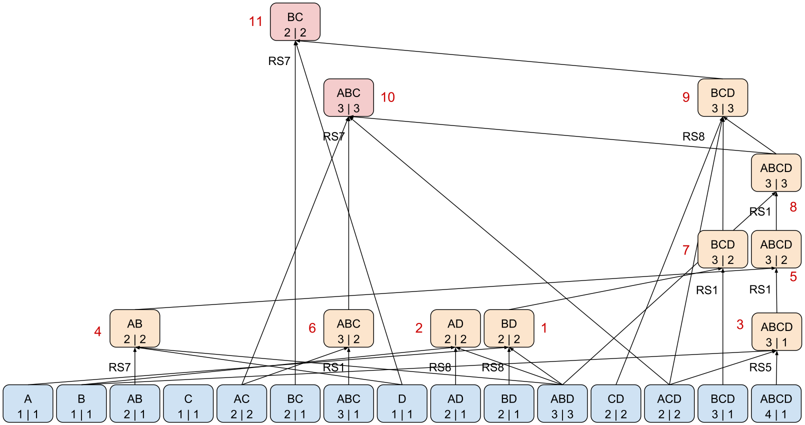
Initialisation

A	B	AB	C	AC	BC	ABC	D	AD	BD	ABD	CD	ACD	BCD	ABCD
1 | 1	1 | 1	2 | 1	1 | 1	2 | 2	2 | 1	3 | 1	1 | 1	2 | 1	2 | 1	3 | 3	2 | 2	2 | 2	3 | 1	4 | 1

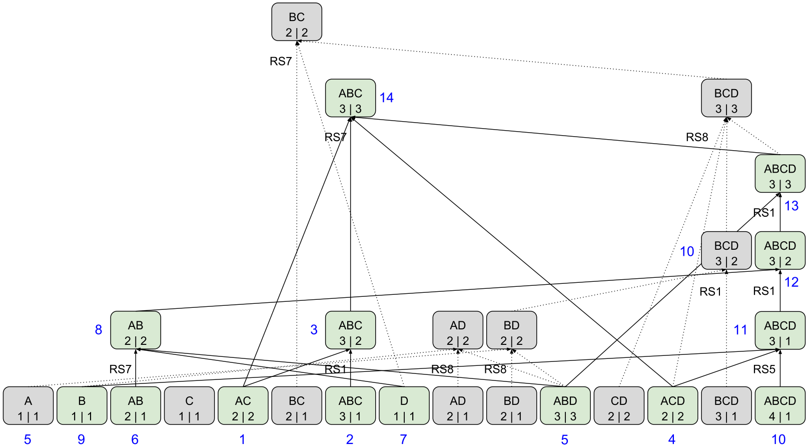
Saturation partielle



Saturation complète



Reconstruction



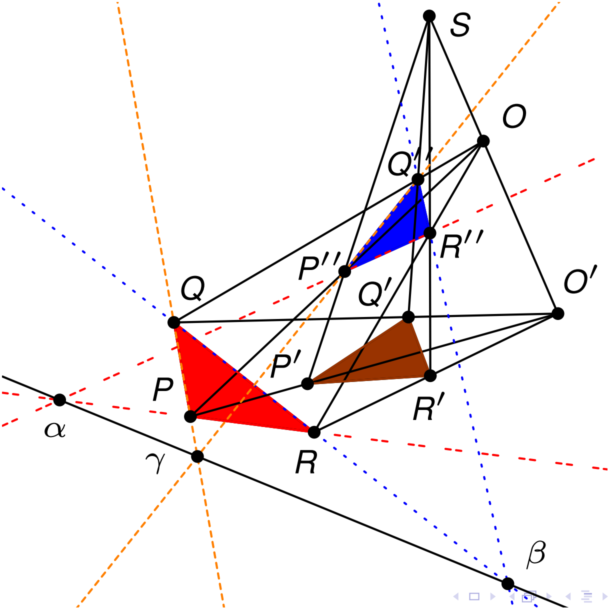
Production d'un script de preuve Coq

- parcours récursif postfixe à partir du noeud dont on cherchait le rang
- lorsque le nombre de points et donc le graphe grossissent : structuration en couches et réutilisation de lemmes
- pour le petit exemple précédent (pas de couche)
  65 lignes de script Coq
  5 pour l'énoncé et 60 pour la preuve

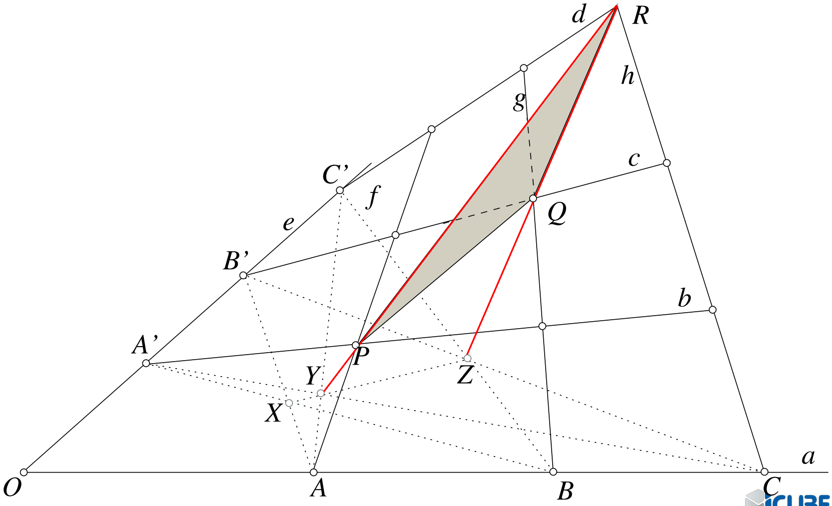
Quelques preuves automatiques

- le théorème de Desargues
- le conjugué harmonique
- le théorème de Dandelin-Gallucci (3D)
  - équivalence entre la propriété de Pappus et la propriété de Dandelin-Gallucci
  - intégration de la règle de Pappus au système de preuves
  - gestion de la création de points réservée à l'utilisateur

Théorème de Desargues



Théorème de Dandelin-Gallucci



Quelques preuves automatiques

- le théorème de Desargues
  15 points, 6 000 lignes,
  génération du script et vérification < 3 minutes
- le conjugué harmonique
  14 points, 10 000 lignes de Coq
- le théorème de Dandelin-Gallucci (Pappus -> DG)
  19 points, 50 000 lignes, 16h pour produire la saturation,
  ingénierie logicielle pour obtenir un script validable par Coq
- le théorème de Dandelin-Gallucci (DG -> Pappus)
  17 points (2h, 34 000 lignes de script en Coq)

Discussion et perspectives

- prouveur automatique (de style hammer)
- pas de création automatique de points, mais plutôt un guidage par l'utilisateur
- Extension à venir : transformer le prouveur automatique en un outil d'aide à la preuve intégré à Coq
- Optimisations
  - amélioration de la réduction d'intervalle
  - génération de scripts Coq plus concis

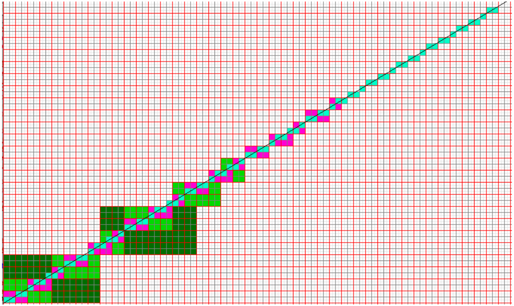
Structure de la présentation

Motivations

- Description des algorithmes géométriques avec une arithmétique réelle exacte.
- Implantations avec des nombres flottants.
- Norme IEEE-754 : prédiction et analyse du comportement de nombreux algorithmes numériques
- Nombres flottants : précision limitée et propriétés différentes de l'arithmétique réelle
- Un point de vue calculatoire : la droite d'Harthong-Reeb
    - Faire des calculs réels uniquement avec des entiers.
    - En s'appuyant sur une arithmétique non-standard.
    - Travailler avec des objets continus dans un cadre discret.

Le continu sur un ordinateur ?

- Idée : travailler à une échelle donnée.
  - A cette échelle, les points ont une taille spécifique.
  - Pouvoir choisir autant d'échelles différentes qu'on le veut.
  - *Zoomer* pour trouver autant de points qu'on veut entre deux points.
- Exemple : une droite de pente $3/5$ à différentes échelles



Un modèle discret du continu

Avoir autant de nombres qu'on veut entre 2 nombres donnés ?

- Utilisation d'une arithmétique non-standard
  - On choisit un entier infiniment grand ω comme nouvelle unité $1_\omega =_{def} \omega$.
  - Deux classes d'éléments : les nombres limités/standards et les nombres infiniment grands
  - Ainsi, entre deux entiers, on peut toujours trouver autant d'entiers qu'on le souhaite.

La droite d'Harthong-Reeb

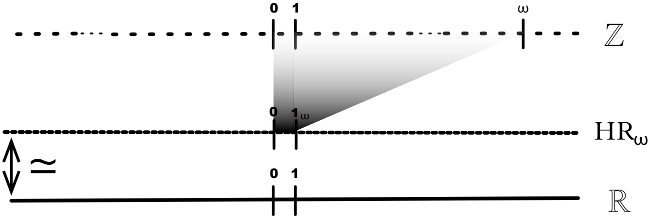
$$\mathcal{HR}_\omega = \{X \in \mathbb{Z}_\Omega,\ \exists n \in \mathbb{N},\ |X| \leq n\omega\}$$

$\mathcal{HR}_\omega$ est une remise à l'échelle de l'arithmétique non-standard choisie.

Un modèle discret du continu

$\mathbb{R}$, c'est $\mathbb{Z}$ vu de loin. (J. Harthong)

Illustation de la droite d'Harthong-Reeb



La droite $\mathcal{HR}_\omega$ est-elle constructive ?

La droite réelle constructive (Douglas Bridges, 1999)

- Un système $(R, +, \times, =, >, 0, 1, \mathrm{Opp}, \mathrm{Inv})$ qui satisfait les 3 groupes d'axiomes suivants :
  - Opérations algébriques (9 axiomes)
  - Structure ordonnée (5 axiomes)
  - Axiome d'Archimede et principe de la borne supérieure constructive (2 axiomes)
- La droite d'Harthong-Reeb vérifie-t-elle les axiomes de Bridges ? En s'appuyant sur quels entiers ?
- Deux propositions de solutions :
  - Une interface : les entiers non-standards axiomatiques
  - Une implantation : les entiers de Laugwitz-Schmieden

Une théorie des entiers non-standards

- Un paramètre abstrait `A:Type`
  représentant les entiers non-standards en Coq
- Des opérations usuelles +,-,*, <,$\leq$ et leurs propriétés
- $\mathcal{HR}_\omega$ = {x :A | $\exists n : A, \texttt{lim}\ n \wedge 0 < n \wedge (|x| \leq n * w)$}.
- Cette propriété est stable par les opérations algébriques.
- Propriétés supplémentaires liées au non-standard

  (LIM1) *L'entier 1 est limité.*
  (LIM2) *La somme et le produit de deux entiers limités sont limités.*
  (LIM3) *Il existe des entiers qui ne sont pas limités (i.e. ω).*
  (LIM4) *Si X est limité et $|Y| \leq |X|$, alors Y est aussi limité.*
  (LIM5) (extension du calcul et du raisonnement usuel au cas non-standard)

De l'interface vers une implantation

- La droite d'Harthong-Reeb avec le type abstrait des entiers non-standards vérifie les axiomes de Bridges.
- Une implantation concrète des entiers non-standards ?
  - On considère des suites $a = (a_n)_{n \in \mathbb{N}}$avec $a_n \in \mathbb{Z}$.
  - munies de l'égalité suivante :

$$a = b \text{ if there exists } N \in \mathbb{N} \text{ s.t. } \forall n > N,\ a_n = b_n.$$

- Exemples
  - $(2, 2, 2, 2, 2, 2, \ldots)$ dénote l'Ω-entier 2.
  - $(1, 5, 4, 2, 2, 2, \ldots) \equiv (2, 2, 2, 2, 2, 2, \ldots)$
  - $\omega = (2^0, 2^1, 2^2, 2^3, 2^4, 2^5, \ldots)$

Opérations pour les entiers de Laugwitz-Schmieden

- Opérations *terme à terme* sur $\mathbb{Z}_\Omega$:
  - $a+b =_{def} (a_n + b_n)$ et $-a =_{def} (-a_n)$ et $a \times b =_{def} (a_n \times b_n)$;
  - $a > b =_{def} [(\exists N\, \forall n > N)\ a_n > b_n]$
  - $|a| =_{def} (|a_n|)$.
- Propriétés
  - Tous les axiomes (R1) sont vérifiés !
  - Seules 3 propriétés du groupe (R2) doivent être adaptées (voir [Chollet et al. TCS 2012])
  - Le principe de la borne supérieure doit aussi être adapté.
- Une forme alternative du continu
  - Propriété non vérifiée : $(\forall a, b \in \mathbb{Z}_\Omega) \quad (a \geqslant b) \vee (b \geqslant a)$
  - Exemple avec $a = ((-1)^n)_{n\in\mathbb{N}}$ et $b = ((-1)^{n+1})_{n\in\mathbb{N}}$.

Arithmétisation et courbes discrètes

Schéma d'arithmétisation d'Euler

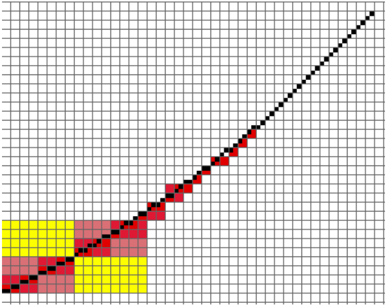
- Objectif : représenter $X : T \mapsto X(T)$
- Solution du problème de Cauchy $X' = F(X, T), X(A) = B$
- Schéma d'Euler

$$\begin{cases} T_0 = A; X_0 = B \\ T_{k+1} = T_k + \frac{1}{h} \\ X_{k+1} = X_k + \frac{1}{h} \times F(T_k, X_k) \end{cases}$$

- Utilisation avec $\mathcal{HR}_\omega$:
  il faut remplacer h par un Ω-entier infiniment grand.

Arithmétisation et courbes discrètes

- L'arithmétisation de la fonction $t \mapsto \frac{t^2}{6}$.



- Calculée avec le code extrait de Coq vers Ocaml
- Elle approxime $X : T \mapsto X(T)$ qui est la solution de $X' = F(X, T), X(A) = B$. Ici, $F(X, T) = T/3$.
- L'Ω-arithmétisation est une représentation fidèle de la fonction continue $T \mapsto X(T)$.

Conclusions

- Une représentation du continu adaptée pour la géométrie
  - Un modèle abstrait
    - avec une description axiomatique des entiers non-standards
  - Un modèle calculatoire
    - avec les entiers de Laugwitz-Schmieden
    - Presque tous les axiomes sont vérifiés
    - Alternatives :
      un sous-ensemble de $\mathcal{HR}_\omega$ ou bien une adaptation des axiomes
- Formalisation en Coq
  - un contexte original
  - nombreux éclaircissements de la description mathématique
- Perspectives : lien avec les nombres B-approximables

Structure de la présentation

Bilan des travaux présentés

- Etudes de cas dans 3 domaines complémentaires
  - Preuves en géométrie algorithmique
  - Automatisation de preuves en géométrie projective
  - Représentation informatique exacte des réels
- Passage d'une modélisation mathématique à la mise en œuvre de solutions informatiques
- Enrichissement mutuel de deux domaines de recherche
  - Utilisation des méthodes formelles de preuve en géométrie
  - Utilisation de la géométrie pour affirmer les capacités des systèmes d'aide à la preuve comme Coq

Perspectives

- Perspectives sur les preuves à venir
  - Preuves automatiques de nouveaux théorèmes de géométrie projective : Pascal, Brianchon, . . .
  - Etude des géométries projectives finies
  - Algorithmes géométriques en 3D
  - Harthong-Reeb et les nombres B-approximables
- Perspectives sur les outils d'aide à la preuve
  - Intégration d'outils externes à Coq comme des *plug-in*
  - Outils d'aide à la gestion des inégalités et encadrements
  - Maintenabilité et pérénnité des développements formels
    - Post-processing des preuves

Merci de votre attention