



HAL
open science

Construction automatique de figures géométriques et programmation logique avec contraintes

Denis Bouhineau

► **To cite this version:**

Denis Bouhineau. Construction automatique de figures géométriques et programmation logique avec contraintes. Autre [cs.OH]. Université Joseph-Fourier - Grenoble I, 1997. Français. NNT: . tel-00004922

HAL Id: tel-00004922

<https://theses.hal.science/tel-00004922>

Submitted on 20 Feb 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Pour obtenir le titre de Docteur
de l'Université Joseph Fourier-Grenoble I
(spécialité informatique)
(Arrêté ministériel du 5 Juillet 1984 et du 30 Mars 1992)

Construction automatique
de figures géométriques
&
Programmation Logique avec Contraintes.

Thèse soutenue par

Denis Bouhineau

le 17 Juin 1997

Composition du jury :

Jacques Cohen
Patrice Enjalbert
Paul Jacquet
Jean-Marie Laborde
Dominique Michelucci
Jean-François Nicaud
Laurent Trilling

Thèse préparée au sein du laboratoire Logiciels Systèmes Réseaux
de l'institut d'Informatique et de Mathématiques Appliquées de Grenoble.

Table des matières

I. Observations sur l'univers de la géométrie à la règle et au compas, en vue de sa mise en œuvre automatique.....	17
I-1. Preuves, transformations, constructions, et algèbre en géométrie : Petite histoire de quelques géométries.....	19
a) La géométrie qui débuta avec Euclide : Les géométries synthétiques.....	19
Euclide.....	19
Hilbert.....	21
Lobachevsky.....	22
b) La géométrie avec et après Descartes : Les géométries analytiques.....	23
Descartes.....	23
Klein.....	23
Tarsky.....	24
Wu 24	
c) La géométrie du jardinier et de l'astronome : géométries des constructions.....	26
d) La géométrie pour l'informaticien ? géométrie des nombres constructibles.	29
I-2. Les nombres constructibles: histoire et mathématiques.	33
a) Section préalable : définitions et propriétés des points et nombres constructibles à la règle et au compas.	33
Constructibilité géométrique.....	33
Nombres constructibles.	34
b) Histoire et résultats sur les nombres constructibles.	37
c) Introduction de la notion de complexité algébrique.....	39
Constructions avec des droites.	40
Constructions avec des cercles.	41
Construction d'un triangle équilatéral.....	42
Les polygones réguliers.	43
I-3. Constructions géométriques élémentaires, nombres constructibles et complexité.....	47
a) Constructions rationnelles.....	48
Définition.....	48
Constructions rationnelles.....	49
Milieu.	49
Autres constructions rationnelles	51
b) Constructions géométriques non-rationnelles.....	52
c) Récapitulatif des constructions rationnelles et non-rationnelles.....	55
d) Spécifications géométriques et complexité.....	56
I-4. Étude pratique de la complexité de quelques situations géométriques courantes, d'un ensemble de 512 théorèmes de géométrie et de situations arbitrairement complexes.	59
a) Étude de quelques situations particulières.	59
Géométrie projective, affine ou à la règle et avec outils rationnels.	59
Autour du cercle des neufs points.....	60
Autour du problème de Napoléon.....	61
b) Étude de 512 théorèmes.....	63
La source : les 512 théorèmes regroupés par Shang-Ching Chou.....	63
Complexité algébrique de 512 théorèmes.	64
Analyse des résultats.....	66
Éléments d'explication.....	68
c) Quelques cas de figures complexes, et arbitrairement complexes d'un point de vue algébrique, Spirales.....	69

Spirales Rationnelles	70
Spirales irrationnelles.....	71
II. Le paradigme CLP(Géométrie).....	77
II-1. CLP - définition d'un paradigme de programmation.....	79
a) Contraintes et Programmation Logique : petite histoire des contributions à la Programmation Logique avec Contraintes.	79
Programmation Logique (Prolog).....	80
Programmation par contraintes.....	82
Prémises de la programmation logique avec contraintes.	82
b) Le paradigme de programmation : CLP.	83
Histoire contemporaine.....	83
Programmation CLP.....	84
Contraintes numériques.	85
Environnement.	85
Exemple.....	87
c) Quelques exemples de langages CLP.....	87
II-2. CLP (Géométrie) : définition du domaine.....	89
a) Définition d'un domaine et tentative de représentation.....	89
b) Représentation à l'aide d'une structure naïve.....	90
Notation.....	92
Addition et soustraction.	92
Multiplication et division.....	93
Extraction d'une racine carrée (provisoire).	94
Ordre et égalité.	94
Commentaires.....	94
c) Représentation normale du domaine et arithmétique associée.....	96
Représentation normale.	96
Arithmétique.....	100
d) Extensibilité du domaine.....	101
Propositions	101
Mise en place.....	101
Exemple.....	102
e) Complexité de la représentation et de l'arithmétique associée.....	103
Conclusion de ce chapitre :.....	104
II-3. CLP (Géométrie) : algorithme de satisfaction de systèmes de contraintes..	107
a) Algorithme naïf de résolution de systèmes d'équations non-linéaires. ...	108
Exemples.....	109
Commentaires.....	110
b) Algorithme de résolution de systèmes de contraintes quadratiques.	110
Algorithme.....	112
Complexité, correction et complétude.....	112
c) Recherche de relations linéaires.	113
d) Algorithme d'élimination de Gauss et nombres constructibles.....	115
Résolution d'un système linéaire sur les nombres constructibles.	115
Correction et complétude.....	117
e) Complétude géométrique.....	118
Complétude pour les spécifications constructives.....	118
Complétude pour les constructions sans ajouts.	120
II-4. Résultats.....	123
a) Représentation des figures géométriques.....	123
Langage géométrique.....	124
Représentation algébrique.....	124
Choix de la grille.....	125
Spécification d'une figure.....	125

b) Figures géométriques où l'algorithme est complet.	126
Problèmes linéaires.	127
Spécifications constructives.....	128
c) Résultats positifs.	130
Segment porté par deux droites connaissant son milieu.	130
Cercle passant par 3 points donnés.	130
Triangle équilatéral à partir d'un sommet et de son centre de gravité g.	131
Cas 1.	131
Cas 2.	132
Triangle inscrit.....	132
d) Résultats négatifs.....	133
Problèmes ardues.....	133
Problèmes "faciles"	133
Triangle rectangle isocèle	134
e) Récapitulatif pour le triangle et ses éléments caractéristiques.	134
Triangle et orthocentre.....	135
Triangle et centre de gravité.....	135
Triangle et cercle circonscrit.	136
Une étude plus large.	137
III. Applications.....	141
III-1. Simplification d'expressions arithmétiques avec racines carrées pour le calcul formel.	143
a) Représentation d'expressions avec racines carrées en calcul formel	143
Modes de représentation d'une suite.....	143
Évaluation d'une formule booléenne.....	145
b) Algorithme de simplification, exemples.....	146
c) Comparaison avec l'algorithme de simplification de Borodin, Fagin, Hopcroft et Tompa.	147
III-2. Vérification de contraintes redondantes en géométrie.	149
a) Aspect algébrique.....	149
Arithmétique exacte.	149
Arithmétique usuelle.	150
Arithmétiques évoluées.....	150
b) Aspect géométrique : Recherche d'une construction rationnelle.....	151
Point sur cercle.	152
Inverse par rapport au cercle.....	153
Bissectrice.....	154
Intersection avec un cercle.	155
Résultats.....	156
III-3. Démonstration numérique exacte de théorèmes de géométrie.	159
a) Méthodes de démonstrations automatiques.....	160
Démonstrations automatiques algébriques : Wentsun Wu.....	160
Démonstrations automatiques numériques : Hong.	162
b) Méthodologie de démonstration automatique par évaluation numérique exacte.	165
Majoration de la forme du reste $R(\dots)$	165
Choix d'un cas particulier unique pour effectuer la transposition de la preuve algébrique à une évaluation numérique.....	167
c) Vérification de contraintes numériques redondantes.	168
d) Démonstration de théorèmes significatifs.	168
Théorème de l'orthocentre	168
Théorème du centroïde.	169
Théorème de Pappus.....	170
Théorème de Simson.	172

III-4. Environnements informatisés d'apprentissage humain de la géométrie....	175
a) Environnements informatisés impératifs ou déclaratifs d'apprentissage	
humain de la géométrie.	176
Exemples.....	177
Résolution d'un problème de construction en utilisant la méthode	
des lieux.....	177
Autour de la démonstration d'un théorème, illustration d'une	
notion de hiérarchie de classes de figures	179
Transformations du plan, invariants et boîtes noires.....	181
Vers une approche déclarative.....	182
Géométrie des animations et géométrie déclarative.....	182
Statut d'un objet dans une figure dynamique	183
Description du système.	184
b) Génération d'exemples et de contre-exemples géométriques dans un	
"préceptuel".....	185
Production de contre-exemples.....	186
Exemples.....	189
c) Réalisation d'un environnement de dessin géométrique à main levée....	192
Acquisition d'un dessin réalisé interactivement par l'utilisateur.....	193
Inférence automatique d'une spécification logique à partir du dessin	
proposé.....	193
Conclusion générale du manuscrit.....	197
Bibliographies.....	201
Bibliographie personnelle:.....	201
Bibliographie internationale:	202
Bibliographie locale :.....	211
Annexes.....	212
- Annexe 1-.....	212
- Annexe 2-.....	213
- Annexe 3-.....	213
- Annexe 4-.....	217
- Annexe 5-.....	218
- Annexe 6-.....	219

Remerciements

Les quelques années passées au sein de l'équipe PLIAGE ont été placées sous le signe de l'enthousiasme, celui de Laurent Trilling, bien sûr, et sous le signe de la bonne humeur qui règne autour de lui. C'est donc en premier à lui et à tous ceux que j'ai fréquenté ces dernières années que j'adresse mes remerciements les plus sincères. C'est avec confiance que je leur transmet mon travail et les encourage à continuer dans la réalisation d'environnements d'enseignement de la géométrie et la promotion de la programmation logique : Stéphane Channac, Pierre Ostier, Edmundo Palacios, Pascal Coste, Paul Amblard, Michel Levy, Jeanne Paumier-Idt, Gérard Vivier.

Je n'oublie pas les stagiaires avec qui j'ai travaillé, ceux qui ont quitté l'équipe PLIAGE et plus généralement tous ceux, en commençant par Paul Jacquet, directeur du LSR, l'équipe administrative et technique, auxquels mon travail est redevable : Cyrille Desmoulins, Pierre Bourgeois, Béatrice Poyet, Christophe Garcia, Florence Nébon, Patrick Lixi, Lilianne Di Giacomo, Martine Pernice, François Challier, Bernard Martinet.

Je remercie également les membres de mon jury d'avoir accepté d'évaluer mon travail et de leur indulgence quant à ses faiblesses. En particulier, je remercie Jacques Cohen, Dominique Michelucci et Patrice Enjalbert pour leur lecture critique du manuscrit et les nombreux commentaires qu'ils en ont tiré, me permettant ainsi d'en améliorer autant la forme que le fond.

Enfin, je remercie la communauté Cabri et au delà les membres du groupe EIAO du PRC-IA, dont Jean-François Nicaud, Martial Vivet et Nicolas Balacheff, pour leur dévouement dans un domaine important à mes yeux. Et, finalement je tiens à remercier énergiquement Jean-Marie Laborde sans qui Cabri et, accessoirement, la quasi-totalité des dessins de ce manuscrit ne seraient pas.

À ma famille et à Evelyne je témoigne tout mon attachement.

Introduction générale du manuscrit.

Cadre

Le travail que nous présentons se situe à l'intersection de trois domaines au moins : la géométrie, l'enseignement et l'informatique.

Le premier remonte à la Grèce antique, c'est la géométrie. Elle suscite à nouveau un intérêt dû à des considérations informatiques. Le renouveau de la géométrie nous intéresse sur deux points. Le premier point est l'avènement des premiers démonstrateurs automatiques de théorèmes utilisables en géométrie. Depuis 1988, avec la démonstration par Chang-Shing Chou de 512 théorèmes de géométrie, le domaine de la démonstration automatique propose des outils réalistes et efficaces. Le second point est lié à l'enseignement de la géométrie, ce qui permet un enchaînement vers le second domaine qui nous intéresse.

Depuis 1986, des projets, tels que le projet IMAG¹ CABRI-Géomètre initié par Jean-Marie Laborde, ont défini ce que pourrait être un CAhier de BRouillon Informatique pour l'enseignement de la géométrie. Ce "Brouillon" se révèle être un véritable succès. L'apport de CABRI-Géomètre est double. D'une part, il introduit la notion de manipulation directe pour les figures géométriques. Une figure construite avec CABRI-Géomètre peut être animée en "tirant" sur les points caractéristiques de la construction. L'ensemble de la figure est reconstruit en temps réel en respectant la spécification initiale de la construction. D'autre part, pour l'enseignement, CABRI-Géomètre définit un nouveau paradigme, la "géométrie dynamique". Cette nouvelle géométrie qui s'intéresse aux animations des figures géométriques, nous semble d'une très grande richesse pédagogique. Elle permet la définition, l'observation et l'utilisation des notions géométriques délicates d'invariant, de classe de figures et de lieu de points.

Le sujet concerné par l'informatique c'est la Programmation Logique avec Contraintes telle que l'ont définie Jean-Louis Lassez, Jaxon Joffar, Alain Colmerauer, Jacques Cohen, William Older et d'autres encore...(PLC ou CLP, acronyme pour Constraints Logic Programming). Issue de travaux récents des années 1980, la programmation logique avec contraintes apporte à la programmation logique (Prolog) la puissance qui lui manquait pour les traitements numériques. L'ambition de ce nouveau paradigme de programmation est la définition d'un cadre de programmation le plus déclaratif possible. Notons également que c'est l'un des environnements de programmation qui comporte le plus d'intelligence, ce qui laisse la liberté au programmeur d'exercer son intelligence ailleurs. Ainsi, la gestion de la mémoire lui est épargnée et il peut se consacrer, dans certains cas, à l'étude d'heuristiques.

Le cadre de notre travail est constitué par l'intersection de ces trois domaines. Cette intersection n'est pas vide. CABRI-Géomètre en fait déjà partie.

¹ IMAG = Informatique et Mathématiques Appliquées de Grenoble

Problématique

L'ensemble des problématiques poursuivies dans ce manuscrit se décline selon les trois domaines décrits précédemment.

Pour l'enseignement, notre ambition est de prolonger ou compléter le travail de CABRI-Géomètre. Plus précisément, nous voulons proposer un Environnement Informatisé d'Apprentissage Humain de la géométrie (en abrégé un EIAH de géométrie) permettant la manipulation directe de *figures géométriques exactes* obtenues à partir d'une *spécification logique déclarative*.

L'informatique apparaît nécessairement dans ce projet avec les deux problématiques suivantes. D'une part, étant *donnée une spécification géométrique, comment **construire automatiquement** une figure géométrique qui vérifie cette spécification*. D'autre part, *comment représenter de manière **exacte** le domaine des nombres constructibles* (i.e. les nombres réels nécessaires à la géométrie de la règle et du compas) *de telle sorte que les spécifications **redondantes** soient autorisées et vérifiées ?* L'apport à la programmation logique avec contraintes est évident puisque ses objectifs rejoignent les nôtres (résolution de systèmes de contraintes tels que ceux qui décrivent une spécification géométrique, et vérification des contraintes redondantes). Les contraintes géométriques correspondent en général à des systèmes de contraintes "quadratiques". Aussi ces deux problèmes sont difficiles et intéressants. Ils constituent la limite entre les problèmes "linéaires" qui sont associés aux nombres rationnels, et les problèmes "non-linéaires" qui sont associés aux réels. Si les problèmes linéaires sont bien traités par la programmation logique avec contraintes, il n'y a pas de solution satisfaisante pour les problèmes non-linéaires.

Les études géométriques nécessaires sont directement liées à ce qui précède. Elles doivent nous permettre d'aborder les problématiques suivantes. *Quelle est la "**complexité intrinsèque**" des figures géométriques ? Est-il possible de "**linéariser**" la construction d'une figure géométrique ?* En quelque sorte, est-il possible d'échapper au problème de la non-linéarité ou au moins de la diminuer grâce à l'étude de la situation géométrique.

L'enchaînement de ces problématiques montre l'imbrication des problèmes que nous voulons résoudre. Il est primordial d'arriver à résoudre chaque problème pour réaliser notre objectif premier : la réalisation d'un EIAH de Géométrie exact et déclaratif.

Démarche

Pour la démarche globale, revenons rapidement au cadre et à la problématique.

La démarche globale s'organise autour de l'utilisation du langage de programmation Prolog III, représentatif de l'approche CLP. Une de nos problématiques est la recherche de constructions automatiques de figures géométriques. Le noyau proposé par la programmation logique avec contraintes pour atteindre cet objectif est constitué par une "boite noire" qui résout les systèmes de contraintes linéaires sur les rationnels. C'est le solveur linéaire de Prolog III dans notre cas. Les caractéristiques algorithmiques du solveur linéaire (Gauss ou simplexe) sont excellentes : correction, complétude et vérification des contraintes redondantes. Notre intention est d'étendre le domaine d'application du solveur linéaire des rationnels aux nombres constructibles, et d'étendre la résolution aux équations quadratiques.

Des solutions générales à ces deux problèmes sont proposées depuis longtemps par les mathématiques. Ceci nous assure de la décidabilité de ces problèmes. Mais les contraintes de l'informatique ne sont pas les mêmes que celles des mathématiques. Le langage des mathématiques, où "existence" ne signifie pas nécessairement "obtention", et la complexité des "algorithmes" mathématiques ne conviennent pas toujours à une application directe en informatique. C'est le cas ici. Aussi, nous devons développer nos propres structures de données pour représenter les nombres constructibles, et nos propres algorithmes pour résoudre le problème de la construction automatique de figures géométriques.

Pour la résolution du problème de construction, nous employons au maximum le résolveur linéaire. De plus nous introduisons un mécanisme pour résoudre les équations du second degré en une seule variable. Cet algorithme est correct et simple, mais il introduit des nombres irrationnels (les nombres constructibles). Par ailleurs, la classe des problèmes qu'il résout est faible. Aussi, pour augmenter cette classe nous enrichissons les systèmes de contraintes à résoudre par des contraintes géométriques redondantes (par exemple en ajoutant les cordes communes à une figure comportant des points à l'intersection de deux cercles).

En ce qui concerne la structure des données représentant les nombres constructibles, notre démarche part d'une représentation naïve. Cette représentation définit les nombres avec racine carrée (les nombres constructibles) selon une structure formelle $a+b*\sqrt{c}$ où a , b et c sont des constantes. Les insuffisances de cette représentation vis à vis du résolveur linéaire et de la vérification des contraintes redondantes impose la recherche d'une représentation normale.

Cette représentation normale constitue l'un des résultats théoriques importants de notre travail. Cependant, cette représentation est de complexité exponentielle. Son utilisation nécessite donc que l'on se penche sur la complexité des situations géométriques courantes. C'est ce que nous entreprenons avec l'étude de 512 théorèmes de géométrie. Les résultats de cette étude constituent un résultat expérimental important de ce manuscrit. En dépend la définition d'environnements d'apprentissage humain de la géométrie. La réalisation de ces EIAH doit constituer le résultat pratique de ce manuscrit.

Pour une démarche a priori qui justifie notre plan, un point d'épistémologie est nécessaire.

Bien sûr, l'informatique est une science à part entière, autonome, dont les préoccupations propres sont des sujets de recherche en soi. C'est une science normale, au sens de Kuhn², qui a développé son propre univers et ses paradigmes. Mais aussi, l'informatique est un outil nécessaire que s'est approprié la réflexion scientifique moderne, au même titre que la logique ou les mathématiques. En particulier, la géométrie et les sciences de l'éducation se servent de l'informatique.

Considérant l'informatique comme une science normale, nous devons nous attacher à développer les notions de correction, de complétude et de complexité attachées aux algorithmes mis en œuvre.

Considérant l'informatique comme moyen, notre objectif étant l'enseignement de la géométrie, nous devons entreprendre une étude de la géométrie avec cette question : «Que peut

² dans Structure des révolutions scientifiques (1972)

nous apprendre la géométrie, en tant que science normale sur la "géométrie scolaire" telle que l'usage des enseignants la limite ? Et plus précisément que peut-elle nous apprendre qui intéresse l'informatique, en vue d'une modélisation du domaine ?»

Par suite, notre manuscrit doit s'organiser autour des trois axes Géométrie, Informatique et Enseignement. Nous choisissons de consacrer une partie pour chaque axe, et nous ordonnons selon l'ordre qui suit. La première partie et la dernière étant respectivement en deçà et au delà de l'informatique "fondamentale" que développe la seconde :

- 1 Géométrie.
- 2 Modélisation informatique du domaine.
- 3 Application à l'enseignement.

La démarche pratique suit le plan suivant.

La première partie de ce manuscrit porte sur la géométrie de la règle et du compas. En apparence, les préoccupations informatiques sont absentes de cette partie. En fait, elles sous-tendent l'ensemble des travaux qui y sont présentés. Le premier chapitre entreprend un historique informel de la géométrie. Il aboutit à la géométrie des constructions et à la géométrie des nombres qui nous intéressent. Le second chapitre poursuit l'étude historique précédente et développe la notion de nombre constructible. Le troisième chapitre entreprend une recherche de constructions "rationnelles". Cette recherche est justifiée par nos capacités en tant qu'informaticien à représenter "facilement" les nombres rationnels et à nos difficultés à représenter les nombres réels. Cette recherche est entreprise sur le plan théorique ainsi que sur le plan expérimental. Elle prend une importance décisive au chapitre quatre avec l'analyse de 512 situations de théorèmes classiques de la géométrie euclidienne. Le défi de ce quatrième chapitre est de trouver des constructions rationnelles pour l'ensemble de ces 512 situations. Les résultats expérimentaux et formels de cette étude fournissent la matière sur laquelle notre travail d'informaticien peut s'établir pour la seconde partie du manuscrit.

La seconde partie porte sur l'informatique et le paradigme de programmation CLP. Elle poursuit un double objectif : représenter le domaine des nombres constructibles et construire automatiquement des figures géométriques à partir de spécifications géométriques. Chacun de ces deux problèmes est d'autant plus difficile qu'il faut produire des figures géométriques exactes. Le premier chapitre commence avec un court historique de la programmation logique avec contraintes. Un panorama des langages de programmation de la communauté CLP est donné. L'intersection de ce panorama avec le domaine de la géométrie est étudiée. Le second chapitre porte sur le premier objectif que nous nous sommes fixé (représentation du domaine). Le troisième chapitre porte sur le second objectif (résolution des systèmes d'équations quadratiques). Nous appelons CLP(Géométrie) le résultat des efforts produits dans ces deux chapitres. Le quatrième chapitre de cette partie porte sur l'utilisation naïve de CLP(Géométrie) pour la géométrie euclidienne. En particulier, la géométrie du triangle est abordée.

La dernière partie porte sur la présentation d'applications de l'outil informatique CLP(Géométrie) défini au cours de la seconde partie. Bien évidemment, l'objectif général est la réalisation d'un EIAH de géométrie. Ceci dit, l'outil CLP(Géométrie) permet aussi la définition d'autres applications qui tendent vers l'enseignement. Nous proposons au premier chapitre un modèle de représentation des expressions numériques avec racines carrées. Notre ambition est

de proposer une représentation normale qui soit la plus proche de l'expression initiale. Le domaine d'application visé est l'enseignement scientifique du secondaire où mathématiques, sciences physiques et chimie emploient ces expressions. Deux applications de CLP(Géométrie) sont aussi présentées qui mènent à la démonstration automatique au second et troisième chapitre : vérification de systèmes de contraintes non-linéaires redondantes et démonstration automatique numérique en géométrie. Pour la vérification de systèmes de contraintes non-linéaires redondantes notre démarche est fondée sur d'une part l'utilisation de l'arithmétique normale présentée en seconde partie et d'autre part sur une recherche de procédures de construction rationnelles qui reprend les résultats de la première partie. Pour le chapitre sur démonstration automatique numérique en géométrie, nous commençons par un exposé des méthodes usuelles de démonstration automatique qui ont suivi les travaux de Wenstun Wu : méthodes algébriques (Type Chou) et méthodes numériques (type Hong). Puis nous proposons une méthode de démonstration automatique qui prolonge les travaux de Hong. Enfin, nous abordons notre objectif premier au dernier chapitre : la réalisation d'un environnement informatisé d'apprentissage humain de la géométrie. Cet EIAH de la géométrie développe une approche déclarative dans l'apprentissage des concepts géométriques qui prolonge naturellement la notion de manipulation directe. Elle a pour objectif supplémentaire l'acquisition et la manipulation de figures géométriques exactes.

Préambule

Notation :

les prédicats Prolog sont présentés dans la syntaxe Marseillaise de Prolog III, ils sont précédés d'un commentaire informel qui donne le profil du prédicat à la manière de Lan Nguyen Phuong : Un système déclaratif de types pour Prolog, thèse de l'Université Joseph Fourier, Grenoble-I, Septembre 1992.

En résumé, la déclaration "tri(Liste!,->Liste)" signifie que "tri" est un prédicat à deux argument, le premier étant une liste déjà instanciée, le second est une variable ou une liste, si c'est une variable elle doit être instanciée par une liste. Un argument peut aussi être de la forme ?Type, si c'est un variable instanciée ou non, qui peut être instanciée ou non, mais en cas d'instanciation elle doit être instanciée avec un objet de type "Type"

De plus nous nous référons aux arguments du prédicat en utilisant la notation suivante : %1 représente le premier argument d'un prédicat, %2 le second etc.

Ainsi le prédicat "tri" a pour profil : tri(Liste!,->Liste) : %2 est la liste triée des éléments de la liste %1.

Pré-requis pour la suite :

Géométrie : niveau Baccalauréat,

Algèbre : niveau Deug (ce travail comporte quelques notions d'algèbre relatives aux éléments algébriques sur Q, un rappel de ces notions est fourni),

Informatique : niveau Licence.

I. Observations sur l'univers de la géométrie à la règle et au compas, en vue de sa mise en œuvre automatique.

Introduction de la première partie :

Cette partie porte essentiellement sur les nombres constructibles, notion commune à la géométrie et à l'algèbre. La géométrie considérée est celle de la règle et du compas; l'algèbre considérée est celle induite par l'utilisation des quatre opérations arithmétiques élémentaires et de la racine carrée.

La notion de nombre constructible fut tout d'abord décrite dans le contexte de la géométrie au dix-septième siècle par Descartes. Cette notion fut enrichie et étudiée au dix-huitième siècle par Wantzel. En particulier, elle permit la résolution de quelques problèmes fameux de construction géométrique vieux de deux mille ans. Aujourd'hui, cette même notion nous est utile pour caractériser la faisabilité et la complexité d'une représentation exacte des coordonnées des figures géométriques.

Cette partie s'ouvre sur un exposé informel de quelques points d'histoire de la géométrie mettant en relief les différentes définitions de ce que l'usage nomme communément la géométrie (chapitre 1).

Le concept de constructibilité introduit dans ce premier chapitre, ainsi que la notion de nombres constructibles, sont développés dans le chapitre 2. Une mesure de la complexité des figures géométriques est également introduite dans ce second chapitre. Toutes ces notions servent à l'étude pratique de la géométrie qui est entreprise ensuite.

Aux chapitres 3 et 4, les situations géométriques usuelles sont examinées d'un point de vue pragmatique, lié à une modélisation informatique du domaine de la géométrie de la règle et du compas. Chapitre 3, les constructions géométriques élémentaires seules (perpendiculaires, médiatrices, bissectrices, etc...) sont étudiées. Chapitre 4, des situations géométriques complètes sont prises en compte.

Cette partie se clôt avec une étude de la complexité de cinq cent douze théorèmes de géométrie. Les résultats de cette étude originale sont particulièrement encourageants.

Notre objectif dans cette première partie porte, dans un premier temps, sur la recherche de critères de complexité pour qualifier les constructions géométriques à la règle et au compas. Ces critères nous sont donnés par la notion de nombre constructible, et la complexité algébrique qui les caractérise. Dans un second temps, nous étudions au travers des situations géométriques concrètes comment optimiser les constructions géométriques vis à vis de ces critères de complexité.

I-1. Preuves, transformations, constructions, et algèbre en géométrie : Petite histoire de quelques géométries.

Introduction :

L'histoire globale de la géométrie reste à écrire. En comparaison, les mathématiques, ou les nombres ont inspiré plus d'historiens des sciences, tels que [Dahan-Damedico, Peiffer-86], [Dedron, Itard-59], [Bourbaki-69], etc... L'écriture d'une histoire de la géométrie est une tâche ardue, qui ne sera pas entreprise ici, car il n'y a pas une géométrie, mais une multitude de géométries. La plupart apparaissent aux siècles derniers, âge d'or pour la géométrie, où elles représentent la science par excellence. Elles offrent la caution d'une rigueur intellectuelle à la philosophie³ et représentent le langage de la nature même : «La nature est écrite en caractères géométriques» -Galilée-

Mais qu'est-ce qu'une géométrie ? Avant d'apporter quelques éléments de réponse, notons que certaines géométries sont oubliées aujourd'hui ou ont changé de nom depuis. Ceci restreint à peine le champ de ce terme "géométrie". Ainsi, au dix-septième siècle, se développe la géométrie "des indivisibles" avec d'illustres géomètres, mathématiciens et physiciens : Pascal, Fermat, Kepler, Roberval, Torricelli,... L'intitulé original est aujourd'hui oublié, remplacé par celui de calcul intégral, beaucoup plus représentatif d'une discipline importante⁴.

Même si nous nous restreignons aux géométries tel que le sens usuel les comprend, il y a eu et il reste plusieurs définitions. Chaque famille de définitions donne lieu à des visions différentes de la géométrie. Nous reconnaissons, pour notre part, quatre familles distinctes par ordre d'importance dans la culture mathématique courante : géométrie synthétique, géométrie analytique, géométrie des constructions et géométrie des nombres constructibles.

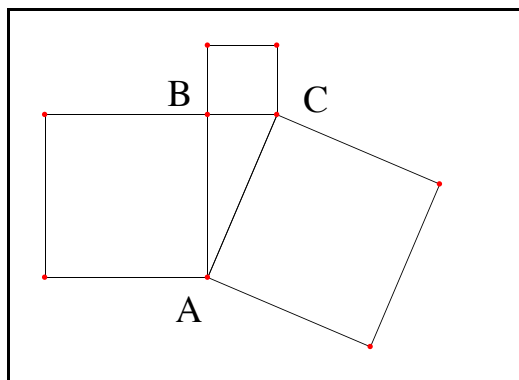
a) La géométrie qui débuta avec Euclide : Les géométries synthétiques.

Euclide

Les historiens de sciences s'accordent pour dire qu'Euclide est le premier à définir correctement la notion formelle de géométrie. Dans Les éléments, la géométrie qu'il décrit apparaît sous une forme qui correspond à la notion moderne des systèmes axiomatiques de

³ A l'époque toute entreprise intellectuelle se devait de suivre une démarche géométrique. Le raisonnement philosophique se nourrissait de la démarche axiomatique de la géométrie d'Euclide. (cf. D'Alembert et Pascal)

⁴La géométrie des indivisibles ne sera pas étudiée ici.



A prouver : les deux carrés construits sur les petits cotés d'un triangle rectangle ont une aire totale égale à celle du carré construit sur l'hypoténuse, c'est à dire :

$$AC^2 = AB^2 + BC^2$$

Idée de la preuve d'Euclide :
voir [Dedron, Itard-59] pp. 352-353

Figure I.1-a: Théorème de Pythagore.

déduction. La géométrie d'Euclide est définie comme un triplet comportant un ensemble d'objets dits "géométriques" (points, droites, segments, cercles, diamètres, angles obtus, droits, aigus, triangles équilatéraux, rectangles, losanges, carrés,...), un ensemble de relations dites "géométriques" (incidence, perpendicularité, parallélisme, congruence, ordre,...), et des règles implicites ou explicites de transitions permettant de déduire de nouvelles propriétés d'une figure en possédant déjà. Parmi ces règles de transitions explicites, certaines ne nécessitent pas d'hypothèses sur la figure et peuvent être appliquées directement, elles sont appelées postulats ou axiomes.

La distinction entre postulat et axiome est peu significative. L'emploi moderne du terme "axiome" prévaut, mais le terme de "postulat" constitue une référence plus forte au texte original d'Euclide. En effet, "postulat" signifie "demande", et les principes de base des Éléments d'Euclide s'exprimaient non pas comme des propriétés dont la véracité était admise par principe comme chez Hilbert (des axiomes), mais comme des demandes dont la satisfaction étaient accordée par principe (des postulats). Euclide demandait : 1) de pouvoir construire une droite reliant 2 points, 2) de pouvoir prolonger une droite, 3) de pouvoir dessiner un cercle de centre et de rayon connus, 4) d'avoir des angles droits toujours égaux, et 5) de toujours pouvoir construire le point d'intersection de deux droites non parallèles.

Le système mis en place par Euclide avait pour vocation principale l'élaboration de démonstrations et de preuves en géométrie qui soient incontestables. Il réalisait ainsi un objectif hérité de Platon : détacher la géométrie du monde sensible pour en faire une science indépendante des fluctuations de sensibilité des individus, de telle sorte que le savoir géométrique soit universel. Un exemple de problème posé est donné par le célèbre théorème de Pythagore illustré figure I.1-a.

La méthode de démonstration issue de la tradition euclidienne reste aujourd'hui très vivante. En intelligence artificielle par exemple, les premiers démonstrateurs en géométries des années 1970 (cf. [Gelernter-63]) étaient construits sur ces théories euclidiennes de la géométrie. Malheureusement, la forte complexité algorithmique des recherches de démonstration dans les systèmes axiomatiques de la géométrie ne permet pas d'obtenir des résultats concluants.

Cependant aujourd'hui encore, après l'avènement des systèmes de calcul formel qui

produisent des démonstrateurs de géométrie performants, un retour s'effectue vers la recherche de démonstrations euclidiennes plus lisibles et pédagogiques, cf. [Wang-96].

En classe enfin, malgré la difficulté inhérente à la recherche de démonstrations euclidiennes elle reste un pilier de l'apprentissage de la démonstration formelle (hypothèses, règles de déduction -> conclusions).

Hilbert

Le progrès essentiel du système d'Euclide sur ceux de ces prédécesseurs tient dans ses axiomes ou postulats qui sont des énoncés vrais admis sans démonstration. Ce sont des preuves justes, sans hypothèses. Ce sont des données du système axiomatique qui ne sont pas rattachées à la réalité, alors que les objets de la géométrie (points, droites, cercles, ...) le restent.

Le plus célèbre des postulats d'Euclide est le cinquième, dit postulat des parallèles. Ce postulat, si important, s'exprime aujourd'hui sous la forme de l'axiome de Playfair.

Axiome de Playfair : Pour un point et une droite, il y a une et une seule parallèle à la droite passant par le point.

Cependant le choix d'un système d'axiomes ou de postulats n'est pas immédiat. Divers problèmes apparaissent : complétude, non-contradiction, indépendance et recherche d'un ensemble minimum d'axiomes. Dès la formulation de ce cinquième postulat et jusqu'au dix-huitième siècle, les géomètres cherchèrent à prouver que cet énoncé pouvait être démontré à partir des autres postulats d'Euclide. «La bataille pour démontrer le cinquième postulat d'Euclide, qui débuta peu après la parution des Éléments (i.e. 300 av. J.C.) continua jusqu'au beau milieu du dix-huitième siècle, mais finalement les mathématiciens réalisèrent que ce cinquième postulat était indépendant des quatre autres.» in [Cederberg-89].

Hilbert poursuit dans la formalisation de la géométrie en opérant une coupure encore plus radicale entre les concepts géométriques et la réalité qu'ils pourraient décrire. Avec Hilbert, les concepts géométriques ne sont pas rattachés explicitement à la réalité, et aucune définition de ces concepts n'est donnée à partir d'autres objets rattachés à la réalité. Ainsi, la géométrie entrait dans le domaine des Mathématiques les plus formelles.

Plus tard, les sciences expérimentales (la philosophie naturelle du dix-septième) eurent naturellement à se demander si la nature était euclidienne ou non. Sans pouvoir apporter de réponses définitives, ou même préliminaires à cette question, nous noterons que «Des recherches depuis la seconde guerre mondiale indiquent que l'espace observé à travers des jumelles binoculaires, est un espace hyperbolique» [Cederberg-89] Mais qu'est-ce qu'un espace hyperbolique ? C'est justement ce que l'histoire va nous apprendre ci-après, en poursuivant l'étude des descendants de la géométrie euclidienne.

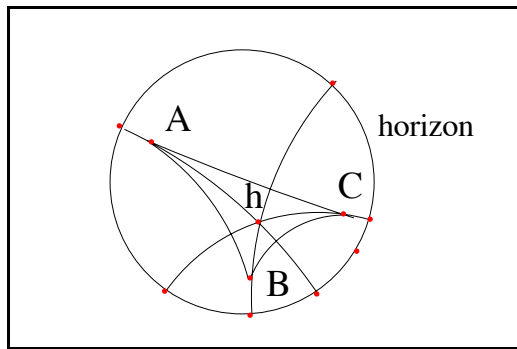


Figure I.1-b: Triangle et hauteurs en géométrie hyperbolique.

Modèle de Poincaré pour la géométrie hyperbolique :

Les points sont des points "euclidiens" intérieurs à un cercle nommé horizon.

Les lignes sont des arcs de cercles "euclidiens" intérieurs à l'horizon et orthogonaux à lui.

Les perpendiculaires sont des arcs de cercles "euclidiens" orthogonaux.

Lobachevsky

Et pourtant, l'axiomatisation de Hilbert et celles qui apparaissent à la même période décrivent essentiellement la même géométrie. Si la géométrie définie par Hilbert ne correspond pas exactement à celle d'Euclide, l'esprit est le même. Hilbert, comme Euclide, veut axiomatiser la géométrie observée dans la nature, et l'idée qu'il se fait de cette nature est sensiblement la même que celle d'Euclide. Le projet de cette nouvelle axiomatisation est d'obtenir une axiomatisation minimale de la géométrie, et la solution au problème de dépendance ou d'indépendance du postulat des parallèles. Mais ce postulat résiste, les géomètres ne savent toujours pas s'il est indépendant ou non.

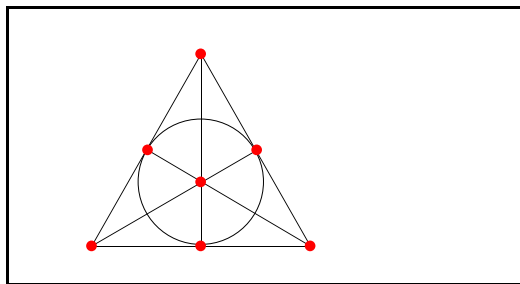
Le résultat de cette recherche fut inattendu. Non seulement ce postulat est indépendant, mais encore il peut être remplacé par d'autres axiomes très différents. Ainsi, d'autres géométries axiomatiques voient leur naissance avec Lobachevsky et Riemann à la même période. En particulier, les géométries hyperboliques et elliptiques. La géométrie usuelle fut alors appelée géométrie euclidienne.

Ainsi, la géométrie hyperbolique remplace l'axiome des parallèles par l'axiome suivant : tout couple de droites se coupe en un point et un seul. Un exemple de modèle pour la géométrie hyperbolique, le modèle de Poincaré, et la construction de l'orthocentre d'un triangle, est donné figure I.1-b.

À la même époque, les géométries finies voient le jour, suite à des axiomatisations plus simples que celle de la géométrie euclidienne. Un exemple d'axiomatisation et de modèle d'une géométrie finie est donné figure I.1-c. C'est un exemple paradoxal de situation qui ne peut se produire en géométrie euclidienne.

Essentiellement, ces géométries ne comportent pas d'axiome de continuité et dégagent les notions clés des systèmes axiomatiques comme le note Judith Cederberg : «Les géométries finies se sont développées à la fin du dix-neuvième siècle en partie pour mettre en évidence et tester les notions de complétude, de consistance et d'indépendance intervenant dans les systèmes axiomatiques.» Elles sont en effet fort intéressantes à étudier dans un cours portant sur ces notions et [Cederberg-89] en donne une illustration parfaite.

Elles permettent par ailleurs aussi de dégager certaines notions de géométrie. Par exemple, quelques géométries finies particulières «illustrent quelques propriétés fondamentales



Modèle comprenant sept points et sept droites, pour la géométrie finie définie par les axiomes suivants :

- Axiome 1 : Par tout couple de points passe une droite.
- Axiome 2 : Tout couple de droites a un point en commun.
- Axiome 3 : Pour chaque couple de point il existe un point aligné.
- Axiome 4 : Il existe trois points non alignés.

Figure I.1-c: Exemple de modèle pour une géométrie finie.

des géométries non euclidiennes et projectives.» in [Cederberg-89]. Mais qu'est-ce que la géométrie projective dont parle Judith Cederberg ? La géométrie projective tient une place importante dans l'histoire de la géométrie. Elle apparaît au dix-septième siècle avec Desargues, Pascal et Lahire, et n'a pour seuls concepts de base que ceux de points, de droite et d'incidence.

L'ensemble des géométries présentées dans cette section I.1-b prennent le nom aujourd'hui de géométries synthétiques. Une hiérarchie des géométries synthétiques est définie naturellement suivant une hiérarchie des axiomatisations.

b) La géométrie avec et après Descartes : Les géométries analytiques.

Descartes

«Au 17^{ième} siècle, Pierre de Fermat (1601-1665) et René Descartes (1596-1650) ont commencé à utiliser des représentations algébriques des figures. Ils réalisèrent qu'en assignant à chaque point du plan un couple de nombres réels, des techniques d'algèbre pouvaient être employées dans l'étude de la géométrie euclidienne.» in [Cederberg-89].

La géométrie analytique est née, à laquelle Wallis, oublié dans cette citation, a aussi contribué. Pour longtemps, la géométrie sera associée avec les coordonnées "cartésiennes", mélange d'arithmétique et de géométrie euclidienne.

Ce mélange est désapprouvé par nombres de géomètres, comme Steiner, mais l'efficacité de cette association en assure le succès. Les énoncés du baccalauréat français, ces dernières années encore, comportent plus de géométrie analytique, source de calculs faciles à corriger, que de géométrie euclidienne. Et pourtant, l'apparition de calculatrices de plus en plus perfectionnées, possédant des fonctions de calcul formel, rend la plupart de ces calculs caduques. Ceci réduit l'intérêt de la géométrie analytique dans le domaine de l'apprentissage humain et doit renouveler celui de la géométrie euclidienne. Cependant la géométrie analytique continue à être un pilier de l'enseignement de la géométrie.

Klein

Peu après la découverte des géométries axiomatiques et la naissance de la géométrie analytique, la notion de géométrie est redéfinie par Klein dans le programme d'Erlangen (1872). Avec Klein, la géométrie apparaît non plus construite à partir de concepts plus ou moins identifiables à des objets et des propriétés observables dans la nature, mais en prenant comme point de départ les transformations du plan : «Une géométrie définit l'étude des propriétés d'un ensemble S restant invariante (inchangées) quand les éléments de S subissent une transformation d'un certain groupe de transformations.» in [Cederberg-89].

Cette nouvelle définition renforce l'introduction de l'algèbre et de l'arithmétique au sens large dans la géométrie. Avant même l'avènement du calcul formel, la géométrie associée à la recherche de preuves (en abrégé géométrie de la preuve, qui regroupe géométries synthétiques et analytiques) imagine utiliser le passage de la géométrie vers l'algèbre pour produire des preuves ou démonstrations.

Tarsky

Un développement majeur est donné à la géométrie de la preuve avec les travaux avant-guerre de Tarsky. La géométrie de Tarsky est analytique. Même plus, Tarsky présente la géométrie "élémentaire" comme équivalente à l'arithmétique "élémentaire".

Le principal résultat de Tarsky porte sur la notion de complétude des systèmes axiomatiques de la géométrie. La géométrie (dont Tarsky donne une définition supplémentaire : d'où le terme géométrie élémentaire utilisé ci dessus), ainsi que l'arithmétique élémentaire (qui, entre autres, ne contient pas tous les réels et qui est équivalente à la géométrie élémentaire) sont des systèmes formels décidables. Clairement, toute propriété bien écrite est décidable, elle peut être démontrée si elle est vraie, et réfutée si elle est fausse.

L'importance de ce résultat apparaît par contraste avec les travaux de Gödel. À la même époque, Gödel démontrait son résultat d'incomplétude.

Théorème d'incomplétude de Gödel : Tout système axiomatique de déduction formelle suffisamment complexe pour décrire l'arithmétique est incomplet : il comporte des propriétés vraies qui ne trouveront pas de démonstrations, des propriétés dont la vérité ou la fausseté ne saura être déterminée, enfin des propriétés fausses dont on ne pourra prouver la fausseté.

La différence entre la géométrie élémentaire et l'arithmétique usuelle tient dans la notion essentielle d'arithmétique (+ et *) sur les entiers. Cette notion permet la définition d'un procédé de diagonalisation "à la Gödel" qui n'apparaît pas en géométrie élémentaire. Elle garantit à l'arithmétique une puissance bien plus grande, mais au prix d'une perte de décidabilité dont la géométrie élémentaire ne souffre pas. À l'appui de sa thèse, Tarsky exhibe une procédure pour déterminer la validité d'une proposition quelconque (finie) de géométrie élémentaire. Malheureusement, la complexité algorithmique de cette procédure (associée à la décomposition cylindrique de Collins) est doublement exponentielle et limite son emploi.

Wu

Beaucoup plus récemment, l'avènement du calcul formel sur ordinateur et l'aisance avec laquelle les manipulations algébriques sont possibles entraînent la naissance de nouvelles

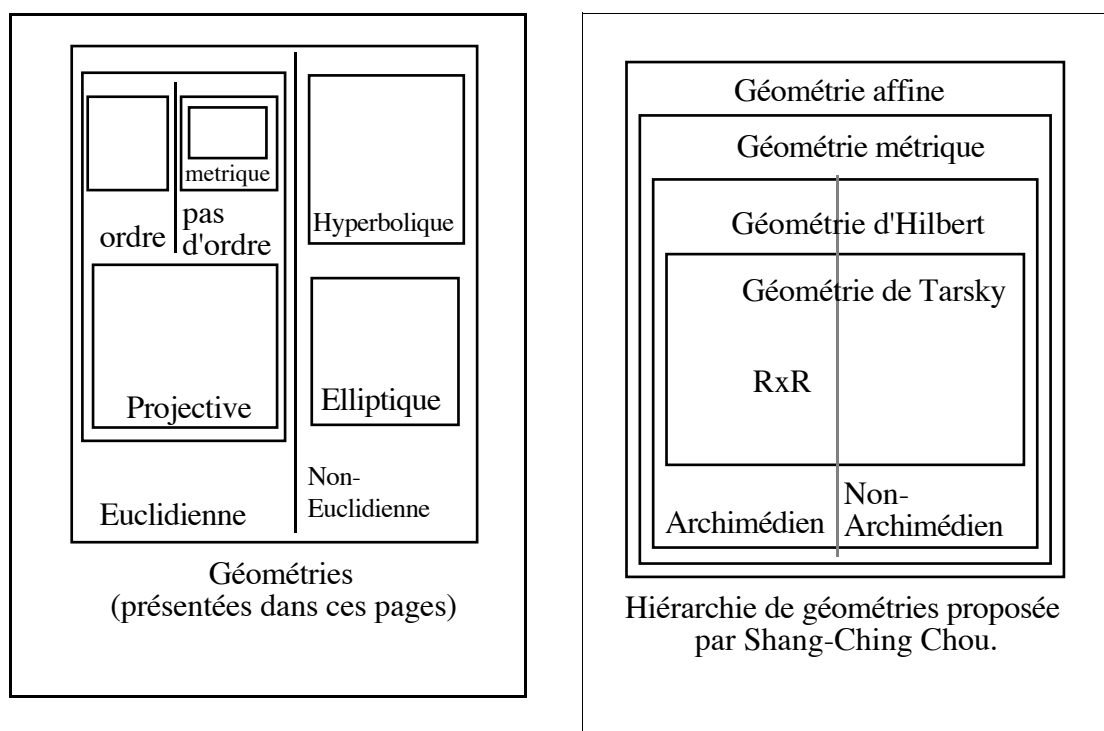


Figure I.1-d: Hiérarchies des géométries.

géométries. Elles sont décrites à la suite de Wu par Chou et ses disciples, cf. figure I.1-d. Ces nouvelles géométries, introduites pour la nécessité des nouveaux algorithmes de décisions, réalisent en pratique l'objectif de Tarsky et Descartes de réduire la géométrie à l'algèbre. Elles permettent de décider mécaniquement (par le calcul) la validité d'une propriété géométrique. Le plus souvent, elles réduisent la géométrie euclidienne au domaine de validité et de complétude des algorithmes de décisions employés.

Les nouvelles géométries définies révèlent dans la classification des théorèmes qu'ils peuvent décider, les difficultés qui subsistent en algèbre pour transposer toutes les notions géométriques. Ainsi les notions d'ordre ou de clôture de corps différent entre géométrie et algèbre. La notion d'ordre, si aisée à comprendre en géométrie, est le plus souvent difficile à mettre en oeuvre en algèbre au travers des idéaux de polynômes (méthodologie usuelle pour représenter les figures géométriques, via les bases de Gröbner, ou d'autres décompositions). Et inversement, la notion de corps clos, naturelle dans les bases de Gröbner qui "travaillent" dans \mathbb{C} et non dans \mathbb{R} , est absente de la géométrie. Cette absence est souvent éludée. Et pourtant, deux cercles du plan peuvent ne pas se couper !

Les premiers démonstrateurs performants de géométrie apparaissent dans les années quatre-vingts, et fournissent des preuves en des temps convenables, dans des environnements aisément disponibles, Macintosh, ou calculatrice dans certains cas. L'objectif actuel des recherches dans ce domaine est l'amélioration de la nature géométrique des démonstrations.

À propos des démonstrations produites par ces algorithmes de décision, il faut noter la nature algébrique de ces démonstrations. Pour un oeil humain, ces démonstrations sont

illisibles et difficiles à vérifier. Par ailleurs, elles produisent des conditions de non-dégénérescence du théorème parfois délicates à interpréter en termes géométriques.

En ce qui nous concerne, nous notons qu'une préférence est donnée, par ces démonstrateurs, aux propositions portant sur une figure donnée sous la forme d'une construction, cf. [Chou et Gao-92]. La norme pour les constructions étant de ne considérer que celles obtenues à la règle et au compas.

c) La géométrie du jardinier et de l'astronome : géométries des constructions.

L'ensemble des géométries décrites jusqu'ici ont pour origine une réflexion sur la notion de théorème (c'est à dire d'invariant pour Klein) et/ou sur la notion de preuve (en prenant une terminologie à la Lakatos). Ce sont essentiellement des géométries de la preuve. Mais réduire la géométrie à la recherche de théorèmes ou de preuves, c'est oublier la moitié ou presque des richesses de la géométrie. C'est oublier quelques uns de ses plus beaux développements au sein des mathématiques, avec l'algèbre pour complice : la géométrie des constructions, celle qu'utilise par exemple le jardinier pour dessiner au cordeau, tour à tour règle et compas, les jardins "à la française".

C'est oublier aussi l'origine, le cœur de la géométrie, les éléments sur lesquelles elle s'est fondée : la règle et le compas. Pourquoi la géométrie s'est-elle développée, avec Euclide, en prenant comme objets de base le point, la droite et le cercle ? Bien sûr, ce choix s'explique par la simplicité de ces figures. Par ailleurs l'exemple de la droite et du cercle nous est donné par la nature avec l'horizon et le Soleil. Enfin le caractère idéal de ces deux figures, sans début ni fin, qui réalisent parfaitement les symétries de translation et de rotation, en font des éléments de base de la géométrie. Mais ces explications ne suffisent pas à expliquer ces choix. C'est Platon et son école, l'Académie⁵, qui avant Euclide porte la responsabilité du choix de la droite et du cercle.

En fait, bien plus que le cercle, c'est le compas qui est choisi. Comme figures idéales pouvant servir à la définition de la géométrie euclidienne, la chaînette, la parabole, la spirale usuelle⁶ peuvent être choisies. Mais seul le cercle possède un moyen de construction facile, fiable et accessible à tous, le compas (ou le cordeau). De même, la droite relève d'un moyen de construction trivial, la règle. Depuis Platon, le choix de la règle et du compas reste stable, comme l'est le modèle de la démonstration donnée par Euclide.

Beaucoup plus tard, à la Renaissance, la règle et le compas sont choisis et employés par divers corps de métiers assez éloignés (jardiniers, architectes, astronomes), pour leur qualité intrinsèque principale : la précision des tracés. Et même, avec l'apparition de procédés de construction plus perfectionnés et plus immédiats utilisant d'autres figures géométriques et

⁵Académie, qui signifie "portique" (celui qui se trouvait à l'entrée de l'école de Platon), et dont la légende raconte qu'il portait l'inscription "Que nul n'entre ici s'il n'est géomètre" (en grec, cela s'entend).

⁶ «Spirale usuelle dite d'Archimède, où un point m se déplace sur une droite D avec une vitesse constante en partant de O, alors que la droite tourne autour de O avec une vitesse angulaire constante» in [Carrega-89]

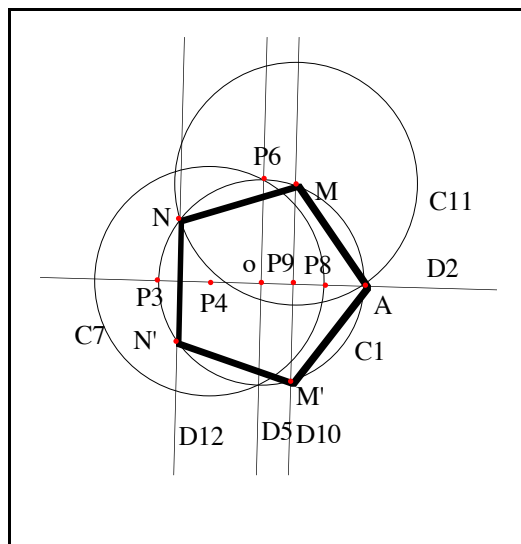


Figure I.1.e : Construction à la règle et au compas.

Construction du pentagone régulier (AMNM'N') à la règle et au compas à partir du centre O et du sommet A :

- construction de C1, cercle de centre O passant par A
- construction de la droite D2, des points P3, et P4 milieu de [O,P3], de la perpendiculaire D5 passant par O à D2, du point P6 sur P5 et C1,
- construction du cercle C7, de centre P3 passant par P6, des points P8 sur 2 et 7, et P9 milieu de [O,P8],
- construction de la perpendiculaire D10 à D2 passant par P9, des points M et M' sur C1 et D10, du cercle C11, du points N de la droite D12 et du point N'.

d'autres outils, le critère de choix, celui de la précision des tracés, fut souvent en faveur de la règle et du compas. A ce sujet, une mesure de la complexité des figures fut mise en place, pour comparer deux constructions, la longueur de chacune représentant sa complexité : la complexité d'Anaxia.

Au dix-septième siècle, la règle et le compas sont mis en concurrence pour savoir lequel de ces deux instruments est le plus précis. Ainsi naît la géométrie du compas seul. De même, naît celle de la règle seule. Par la suite, diverses géométries liés à d'autres procédés et ensembles de procédés de construction voient le jour. Notons les ensembles suivants : la règle accompagnée selon les cas d'un traceur de parallèles, d'une équerre, d'un cercle unique (centré à l'origine et unitaire par exemple), d'un bissecteur, ou encore d'un transporteur de distances (compas non traçant, ou compas à pointe sèche).

La géométrie du compas seul pour mentionner les travaux et résultats de Mohr et surtout de Macheroni avaient des préoccupations utilitaires : les applications visées étaient la construction de cadrans solaires et la graduation d'instruments d'astronomie de haute précision.

La géométrie de la règle seule progressera en suivant les travaux au dix-septième siècle de Desargues, Pascal et Lahire en géométrie projective. La géométrie projective fut redécouverte à l'occasion, et servit largement d'inspiration.

Nous reviendrons sur les géométries de la règle seule, et du compas seul à la section suivante pour les résultats de constructibilité de Mohr, Steiner et Poncelet.

Encore plus éloignée de la géométrie euclidienne, se trouve la géométrie des pliages (équivalente à celle de la règle accompagnée d'un bissecteur). Notons enfin la géométrie qui se déroule sur une feuille finie, elle pose des problèmes concrets de constructions intéressants et pédagogiques, comme le montre la figure I.1.f.

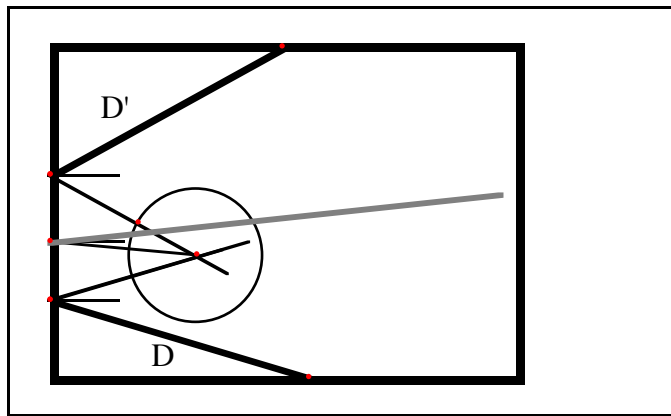


Figure I.1.f: Construction d'une bissectrice sur une feuille.

Problème :

Comment construire la bissectrice de deux droites D et D' données sur une feuille (finie), quand les deux droites données se coupent en dehors de la feuille ?

(Solution ci-contre)

L'équivalent des théorèmes à démontrer et des nouvelles propriétés à découvrir en géométrie synthétique (ou analytique) consiste dans cette géométrie en la recherche de constructions de figures particulières ou en la recherche de nouvelles constructions plus rapides, plus précises.

L'utilisation de la règle et du compas ne constituait pas un cadre limitant. D'autres outils de constructions ou d'autres courbes accessoires ont été utilisés (la règle de Nicomède qui permet le dessin de la conchoïde du même Nicomède, la quadratrice de Dinostrate, l'intégraphe,...)

Des résolutions pragmatiques de plusieurs types, utilisant ces instruments ou procédés de constructions, ont ainsi été données :

- Résolutions graphiques quand la courbe peut être obtenue point par point à la règle et au compas (c'est à dire lorsque le point générique de la courbe auxiliaire est constructible à la règle et au compas)

- Résolutions mécaniques quand la courbe ne peut être obtenue point par point continûment à la règle et au compas mais seulement par un système mécanique, et résolutions approchées.

L'exemple le plus frappant est donné par la recherche d'une construction de π .

Dans tous les cas, ces résolutions pragmatiques aux problèmes difficiles de constructions ne constituaient pas une solution finale. Les géomètres poursuivaient leur recherche pour trouver une construction à la règle et au compas répondant aux canons de la géométrie.

Parmi les problèmes de construction difficiles, les plus connus sont la quadrature du cercle (construire un carré ayant une surface égale à celle d'un cercle), la trisection d'un angle, et la duplication du cube. Tout comme la recherche d'une démonstration du cinquième postulat d'Euclide, ces trois problèmes ne reçurent une réponse qu'à l'aube de l'ère moderne, près de 2000 ans après avoir été formulés. Et c'est sur un résultat négatif, là aussi⁷, que se termina cette quête.

⁷ tout comme fut soldée par un échec, la recherche d'une démonstration du cinquième postulat d'Euclide.

d) La géométrie pour l'informaticien ? géométrie des nombres constructibles.

Pour clore cette courte histoire de la géométrie, il faut étudier les rapports qu'ont entretenus la géométrie et les nombres. Ce morceau d'histoire est traité succinctement dans cette section, car le sujet est repris plus finement dans le chapitre qui suit. D'une certaine manière, c'est l'histoire qui intéresse le plus l'informaticien, car elle concerne les nombres, données qui vont lui servir à représenter la géométrie.

La connaissance populaire associe facilement Descartes avec le passage de la géométrie aux nombres, mais en fait, les relations entre la géométrie et les nombres existaient bien avant Descartes. La géométrie servait à l'algèbre depuis la Grèce antique. Descartes n'a fait qu'inverser le lien qui unissait nombres et figures géométriques, auparavant la géométrie servait aux calculs arithmétiques et non l'inverse.

C'est l'une des histoires les plus fascinantes et les plus intéressantes, et pourtant l'une des moins connues et enseignées. Souvent, dans les histoires des nombres, des chiffres des systèmes de numérations -qui sont nombreuses- cette histoire est oubliée. C'est un exemple où l'ingéniosité humaine s'est particulièrement exercée pour accorder une théorie humaine, a priori évidente, représentant sans défaillir une notion immédiate, celle des nombres, avec une réalité plus éloignée, appréhendée par notre seule intuition, celle de la mesure des longueurs et surfaces. Le problème consiste à modéliser l'intuition des réels que nous avons, à partir des notions de longueurs, avec les outils que nous offre la théorie des nombres entiers. Il s'agit en même temps d'éclaircir le rapport entre la géométrie et les nombres, sujet étudié dès Euclide.

Le terme essentiel de ces travaux, la commensurabilité, la recherche d'un rapport commun entre deux longueurs, montre l'effort pour trouver l'adéquation d'une théorie avec la réalité. Elle a été dès le départ au centre des préoccupations des mathématiciens géomètres.

Le premier théorème qui regroupe la géométrie et la théorie des nombres est aussi l'un des plus anciens théorèmes de géométrie, c'est le théorème de Thalès. Connus depuis longtemps, c'est un rapport d'homothétie qui exprime une simple règle de trois. Avec ce théorème et la règle seulement, les géomètres grecs se rendent compte qu'ils peuvent exprimer tous les nombres entiers et rationnels en termes de longueurs et de rapports de longueur. La réciproque leur semble immédiate. Et par suite, l'école pythagoricienne prend le nombre entier comme l'élément de base de toute longueur, et accorde à l'entier un caractère divin.

L'étude des nombres entiers prend son essor avec l'école pythagoricienne. Les nombres "entiers" sont naturellement associés à la géométrie sous forme figurative. Les nombres sont "linéaires", "plans" ou "solides". Ils prennent toutes les formes possibles, "triangulaires", "carrés", "oblongs", "pentagonaux", "pyramidaux", cf. figure I.1-g.

Finalement les géomètres statuent que deux longueurs sont commensurables si l'une et l'autre démultipliées par des coefficients entiers respectifs sont alors égales. La question "toutes les longueurs sont-elles commensurables ?" vient naturellement puisque tous les nombres connus alors (les nombres rationnels) le sont.

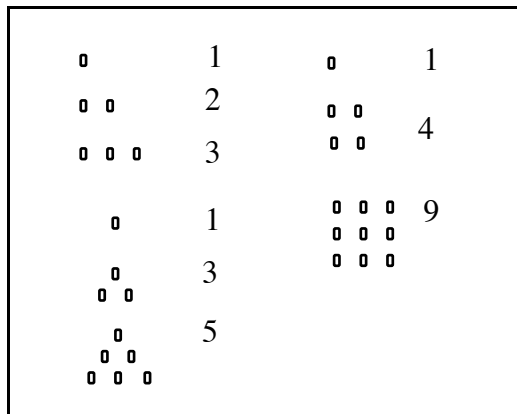


Figure I.1-g : Les nombres pythagoriciens.

Trois séries de trois nombres :

- Nombres linéaires : 1, 2, 3; c'est à dire tous les nombres entiers.

- Nombres triangulaires : 1, 3, 5; c'est à dire tous les nombres de la forme $n(n+1)/2$.

- Nombres carrés : 1, 4, 9; c'est à dire tous les entiers carrés.

Mais avant de répondre à cette question, notons que le dogme pythagoricien est critiqué. Les paradoxes de Zénon d'Élée appartiennent à cet univers. Zénon d'Élée, contemporain des pythagoriciens, s'établit contre ceux-ci et dénonce au travers de ses paradoxes l'emploi d'un paradigme unique pour décrire la réalité. Ainsi, contre l'idée de la continuité et la divisibilité infinie de l'éther, Zénon observe le parcours d'une flèche qui, à force de ne franchir que la moitié du chemin qui lui reste, n'arrive jamais à son but. Contre l'idée de quantification de l'espace, Zénon observe la même flèche immobile dans l'espace, car immobile à chaque instant.

Il est intéressant de constater que le choix d'un modèle pour décrire le réel reste toujours un épineux problème, et que le choix entre le continu et le discontinu se pose quand le physicien veut parler de sable ou de fluide. Bien sûr, d'autres modélisations sont aujourd'hui offertes, ce qui rend le problème du choix encore plus délicat.

Pour répondre à ces critiques, la "physique" se démarqua du dogme pythagoricien, en introduisant la théorie de l'atome, l'insécable, dû à Démocrite et avant lui à Leucippe.

Pour la question de la commensurabilité entre longueurs, la réponse vint d'un second théorème de géométrie très ancien aussi, bien que moins que le théorème de Thalès. Il s'agit du théorème de Pythagore, réunissant géométrie et théorie des nombres. La réponse donnée par le théorème de Pythagore fut une catastrophe. La réponse était négative, toutes les longueurs n'étaient pas commensurables.

Le théorème de Pythagore avait entraîné la découverte des irrationnels, et en particulier de la racine carrée de 2, rapport de la longueur de la diagonale d'un carré à la longueur du côté, c'est à dire rapport de l'hypoténuse au côté d'un triangle rectangle isocèle (triangle scalène pour Platon). Platon montre, dans le Ménon, que ce nombre n'est pas rationnel. Il ne va pas pour autant lui donner de statut.

Pour cette même démonstration de l'incommensurabilité, c'est à dire l'irrationalité de $\sqrt{2}$, quelque temps plus tôt Hypostase de Métaponte fut noyé, cf. [Dahan-Damedico, Peiffer-86]. En effet, l'école pythagoricienne, à laquelle appartenait Métaponte, formait une secte au sens moderne. Cette école était très stricte sur la conservation du silence sur cette irrationalité. Le secret devait être conservé car les pythagoriciens vivaient dans la croyance que les dieux avaient

créé les éléments, la nature et les hommes avec des nombres entiers. Ainsi la nature faite d'entiers exprimait la perfection que les dieux lui avaient donnée. La découverte des irrationnels dans la nature était le signe pour les pythagoriciens d'une imperfection de la nature et donc des dieux. La révélation de cette imperfection risquait d'entraîner la colère des dieux, c'était donc un sujet qu'il fallait garder secret. «Selon Proclus, Euclide aurait commenté la légende d'après laquelle celui qui le premier avait divulgué l'irrationalité de $\sqrt{2}$ aurait péri noyé dans un naufrage : les auteurs de la légende ont voulu parler par allégorie. Ils ont voulu dire que tout ce qui est irrationnel et privé de forme doit demeurer caché. Que si l'âme veut pénétrer dans cette région secrète et la laisser ouverte, alors elle est entraînée dans la mer du devenir et noyée dans l'incessant mouvement de ses courants.» in [Dahan-Damedico, Peiffer-86].

Pour Aristote et Platon, c'était un secret connu fort inquiétant. Après la découverte de l'irrationalité de $\sqrt{2}$, d'autres survinrent pour les racines carrées de 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15 et 17 comme le commente Platon. Par suite, quand Euclide écrivit les Éléments, il savait que la règle ne sera pas suffisante pour construire tous les nombres. Ce fut une raison d'employer comme second outil de construction le compas. Le compas permet de résoudre le problème de l'incommensurabilité des racines carrées connues, puisqu'il permet de construire le nombre recherché. Ainsi, ces nombres deviennent commensurables à l'aide du compas⁸.

A l'aube de l'ère chrétienne, les géomètres croyaient avoir ainsi atteint la fin des problèmes concernant les relations entre nombres et longueurs. Malgré les problèmes de construction cités au paragraphe précédent, les géomètres pensaient pouvoir construire tout nombre à la règle et au compas. Plus même, puisque Platon dans le Ménon affirme que tous les nombres sont constructibles à l'aide de fractions rationnelles et des deux racines carrées $\sqrt{2}$ et $\sqrt{3}$ seulement. Avec $\sqrt{2}$ et $\sqrt{3}$, Platon pensait avoir trouvé parmi tous les réels une formule simple pour π . Cette erreur vient probablement d'une méprise pardonnable car $\sqrt{2}+\sqrt{3}$ est très proche de π qui était recherché avec beaucoup d'intérêt par les géomètres.

L'histoire attendra le renouveau mathématique du dix-septième siècle pour rectifier les erreurs et trouver les solutions aux problèmes. La première étape fut l'apparition avec Descartes d'une formalisation des nombres constructibles. Un siècle plus tard avec les travaux de Galois

⁸Jean-Claude Carrega trouve ici la raison principale du choix du compas : «Les pythagoriciens connaissaient bien les entiers naturels permettant le dénombrement, ainsi que les fractions d'entiers naturels, celles-ci leur permettaient entre autres de mesurer les segments [...] On a cru que l'on pouvait ainsi mesurer tous les segments, mais les conséquences du théorème de Pythagore (vers 550 av. J.C.) allaient provoquer une crise. On obtient en effet à partir d'un triangle rectangle isocèle ABC, $(BC/AB)^2 = 2$, et l'on démontre qu'il n'existe pas de fraction p/q telles que $(p/q)^2 = 2$. Ainsi les segments AB et BC sont incommensurables à l'aide des fractions. Il fallait donc accepter de nouveaux nombres, mais ces nouveaux nombres quel statut leur donner ? On remarque que le segment BC est obtenu à partir du segment AB par une construction à la règle et au compas. [...] On peut donc penser, mais ceci, n'est qu'une hypothèse, que les constructions à la règle et au compas ont été mises en avant pour servir de cautions géométriques aux nouveaux nombres mis en évidence par le théorème de Pythagore. De plus, en s'interdisant d'autres types de construction on se préservait contre de nouvelles crises.» in [Carrega-89]. À la fin, nous pouvons voir une référence discrète à Lakatos, volontaire ou non.

comme initiateur, de Wantzel pour l'essentiel et de Hilbert en ce qui concerne quelques raffinements, les mathématiciens purent exploiter complètement le lien unissant nombres et géométrie. Ces travaux algébriques introduisant la notion de nombres constructibles ont permis l'obtention de résultats purement géométriques sur la constructibilité à la règle et au compas.

Quant au problème métaphysique du lien entre théorie et réalité, entre nombres et longueurs, il reste malheureusement entier. Tout comme reste entier le problème de l'efficacité ou de la réfutabilité des modèles euclidiens et non euclidiens pour représenter l'espace. Cette autre question est un résultat que l'on doit plutôt à la découverte des théories non euclidiennes.

Conclusion de ce chapitre :

Des quatre types de géométries observés ici, géométries synthétiques, géométries analytiques, géométries des constructions et géométries des nombres, les deux dernières retiennent particulièrement notre attention. Le domaine de la géométrie de construction pour nous donner quelques résultats théoriques de référence sur la constructibilité de figures géométriques; la géométrie des nombres pour nous donner aussi quelques résultats théoriques de référence et pour servir de base à une réflexion sur la possibilité de représenter numériquement les coordonnées intervenant dans une situation géométrique. Ces éléments sont abordés dans le chapitre suivant, en reprenant l'histoire des nombres intervenant en géométrie afin d'observer plus systématiquement les développements modernes.

I-2. Les nombres constructibles: histoire et mathématiques.

Introduction :

La notion de nombre constructible introduite dans le chapitre précédent est abordée ici de manière plus complète. Dans une section préalable, une définition formelle adaptée à une utilisation en informatique est donnée. À la section suivante, l'histoire de la géométrie et les nombres constructibles est reprise à partir des résultats de Descartes, jusqu'aux découvertes conclusives de Wantzel sur la constructibilité, et au delà avec Hilbert pour quelques théorèmes annexes. Enfin, quelques conclusions d'ordre pratique sont données et une étude de la constructibilité des polygones réguliers est effectuée comme point de repère.

Seuls les aspects géométriques, mathématiques (algébriques) et historiques sont exposés dans ce chapitre. Les aspects liés à l'informatique sont, pour la plupart, laissés pour la seconde partie de ce manuscrit.

a) Section préalable : définitions et propriétés des points et nombres constructibles à la règle et au compas.

Constructibilité géométrique.

Commençons par la définition classique des points constructibles à la règle et au compas telle que Descartes aurait pu la proposer.

Définition I.2-a Un point M du plan P est dit constructible à la règle et au compas à partir d'un ensemble de points B initial s'il existe une suite $(M_i)_{1..n}$ finie de points de P se terminant par M , tels que pour tout i allant de 1 à n , M_i est point d'intersection soit de deux droites, soit d'une droite et d'un cercle, soit de deux cercles. Ces droites et cercles doivent être eux-mêmes constructibles à partir de B et des $(M_i)_{1...(i-1)}$.

Définition I.2-b Une droite passant par deux points constructibles distincts est dite constructible.

Définition I.2-c Un cercle ayant pour centre un point constructible et passant par un second point constructible différent du premier est dit constructible.

Le choix de l'ensemble B initial peut donner lieu à plusieurs définitions distinctes ou équivalentes de la notion de constructibilité.

La plus naturelle consiste à prendre pour B un ensemble minimal. Les cas où B est réduit à l'ensemble vide ou à un singleton n'ont pas d'intérêt. Ils ne permettent pas d'ajouter de nouveaux points par construction. Le cas minimal intéressant revient à prendre pour B un ensemble B_0 de deux points distincts. L'usage consiste à appeler "O" le premier point et "I" le second. À partir de ces deux points, à la règle et au compas, deux axes (Ox) et (Oy) perpendiculaires peuvent être construits ainsi qu'un repère orthonormé (O, i, j) . Bien d'autres points peuvent être construits de même. Par exemple, tous points aux coordonnées rationnelles dans ce repère peuvent être construits.

La constructibilité à partir de B_0 est par conséquent équivalente à la constructibilité à partir de l'ensemble B_Q , ensemble des points à coordonnées rationnelles dans un repère orthonormé R donné. Il suffit de faire coïncider "O" et "I" de B_0 avec les points $(0,0)$ et $(1,0)$ de R .

En orientant la définition précédente pour une utilisation en informatique, nous pouvons prendre pour B , l'ensemble B_G des points d'une grille définie par l'ensemble des points qui peuvent être désignés à l'écran d'un ordinateur à l'aide d'une souris. Il va de soit que la constructibilité à partir de B_G est équivalente à la constructibilité à partir de B_0 , quels que soient l'écran et l'ordinateur classique particuliers choisis.

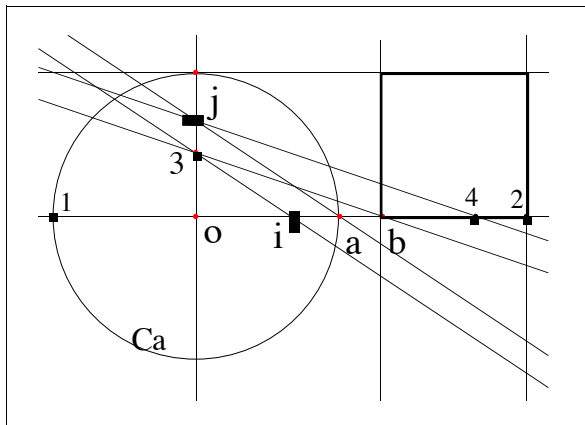
En prenant pour B un ensemble B_n de n points quelconques du plan ($n > 2$), la définition de la constructibilité à partir de B_n est, en général, différente de la constructibilité à partir de B_0 . Sur le plan mathématique, cela revient à remplacer Q (dans B_Q) par une extension de Q avec un certain nombre d'indéterminées a, b, c, \dots (pour donner $B_{Q[a,b,c,\dots]}$). Mais cette autre définition plus puissante n'est pas utilisée ici. Elle introduit une puissance bien délicate à utiliser qui ne correspond pas à la notion de constructibilité la plus courante.

Notons également l'existence d'autres définitions de la constructibilité. Celle présentée ci-dessus repose essentiellement sur la construction de points. À partir d'une figure géométrique initiale, une notion de constructibilité étend la définition de la constructibilité à partir de B à l'ensemble des points obtenus à partir de nouveaux points et des éléments "caractéristiques" de la figure initiale. Ces définitions plus "riches" de la constructibilité ne seront pas utilisées dans ce qui suit.

Nombres constructibles.

Le choix du repère et des coordonnées introduites précédemment permet le passage des notions purement géométriques aux notions numériques. Ils permettent la définition de la notion de nombre constructible.

Mais si le passage des points constructibles aux nombres constructibles se fait par le choix d'un repère R et d'une grille G , quelle est l'incidence de ces choix ? Comme nous l'avons noté ci-dessus, ces choix donnent des résultats équivalents en ce qui concerne la notion classique de points constructibles, car la constructibilité à partir de $B_0/B_G/B_Q$ est équivalente à la constructibilité à partir de B_0 . Pour faciliter notre exposé, nous supposons donc que nous avons choisi un repère R et une grille G , et que la constructibilité ainsi définie est équivalente à la constructibilité à partir de B_0 . Voici la définition des nombres constructibles :



Soient (o, i, j) un repère orthonormé, et a, b deux longueurs,

(1) obtenu avec le cercle C_a vaut l'opposé de a , i.e. $-a$,

(2) obtenu avec C_a et un carré vaut la somme $a+b$,

(3) obtenu avec deux parallèles (Thalès) vaut l'inverse de a , i.e. $1/a$,

(4) obtenu avec deux fois deux parallèles (2*Thalès) vaut le produit $a.b$

Figure I.2-a: Opérations arithmétiques sur les longueurs.

Définition I.2-d Un nombre réel est dit constructible si c'est l'une des coordonnées dans le repère R d'un point constructible à la règle et au compas à partir de B_G .

D'après la définition précédente, l'écriture canonique d'un point sous la forme d'un couple de nombres (ou coordonnées cartésiennes) réalise l'adéquation entre les notions de nombre constructible et d'objet géométrique constructible. La proposition qui suit complète et poursuit cette adéquation pour les droites, et les cercles.

Proposition I.2-a : Étant données un repère R et une grille G , il y a équivalence entre les énoncés suivants :

(1) un objet M (point, droite, cercle respectivement) est constructible à la règle et au compas à partir de la grille G

(2) l'équation d'un objet M (point, droite, cercle respectivement) peut être écrite à l'aide de nombres constructibles uniquement.

[[preuve]]

La preuve de la proposition utilise le lemme auxiliaire suivant illustré par les figures I.2-a et I.2-b .

Lemme auxiliaire I.2-b : Dans un repère donné, la somme, la différence, le produit, le rapport de deux longueurs et l'extraction de la racine carrée d'une longueur sont représentables par des longueurs constructibles à la règle et au compas.

[[La démonstration de ce résultat dû à Descartes est laissée au lecteur.]]

Passons à la preuve de la proposition :

Nous décrivons un point P par deux coordonnées (P_x, P_y) dans le repère R .

(1) \Rightarrow (2) Soit P un point constructible. Montrons que les coordonnées du point sont des nombres constructibles. Soient P_x et P_y les coordonnées de P , alors par définition, P_x et P_y sont des coordonnées d'un point constructible, ce sont donc des nombres constructibles.

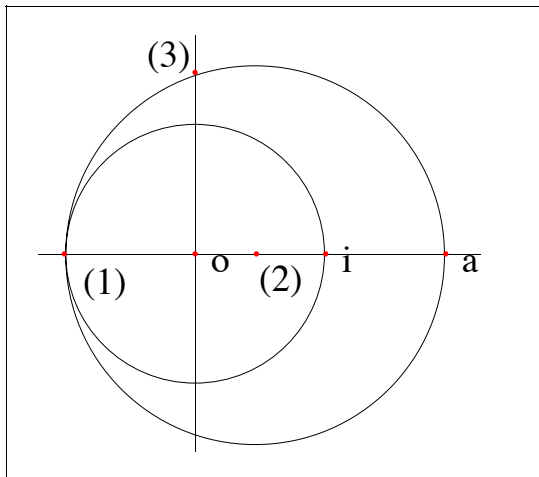


Figure I.2-b: Construction de la racine carré d'une longueur.

Soient (o, i) le repère, et a une longueur,
 (1) vaut l'opposé de i ,
 (2) est le milieu de $[a, (1)]$, il sert à construire le cercle qui donne (3).

Que vaut (3) ?

Soit $K=[o, (3)]$, $X=[(1), (3)]$ et $Y=[(3), a]$.

Dans le triangle rectangle $(1, 3, a)$ il y a $(1+a)^2 = X^2 + Y^2$.

Dans $(1, o, 3)$, il y a $X^2 = 1 + K^2$.

Dans $(a, o, 3)$, il y a $Y^2 = a^2 + K^2$.

Par suite, $(1+a)^2 = 1 + a^2 + 2K^2$, d'où $K^2 = a$, $K = \sqrt{a}$. Le point (3) est à \sqrt{a} .

(2) \Rightarrow (1) Soient x et y deux nombres constructibles, alors il existe deux points constructibles à la règle et au compas P_x et P_y tels que x et y soient coordonnées de ces points. Quitte à échanger P_x et P_y avec leur symétrique par rapport à la première bissectrice (par une construction à la règle et au compas), nous supposons que P_x a pour abscisse x et P_y pour ordonnée y . Alors le point P , intersection de l'horizontale passant par P_y et de la verticale passant par P_x , est constructible à la règle et au compas, et il a pour coordonnées P_x et P_y .

Nous décrivons une droite par une équation $y = m \cdot x + p$ dans le repère R . Les droites horizontales ($x = c$) sont laissées au lecteur.

(1) \Rightarrow (2) Soit D une droite constructible. Montrons que son équation peut être décrite à partir de nombres constructibles. Par définition, il existe M_0 et M_1 deux points constructibles distincts tels que D passe par M_0 et M_1 . En notant $M_0 = (x_0, y_0)$ et $M_1 = (x_1, y_1)$, l'équation de D est donnée par $y = m \cdot x + p$ avec $m = (y_0 - y_1) / (x_0 - x_1)$ et $p = (x_0 \cdot y_1 - x_1 \cdot y_0) / (x_0 - x_1)$. En adoptant le lemme auxiliaire I.2-b, nous supposons vérifié que la somme, la différence, le produit, le rapport de deux nombres constructibles sont des nombres constructibles. Par suite, l'équation de D s'exprime à l'aide de nombres constructibles.

(2) \Rightarrow (1) Soient m et p deux nombres constructibles, montrons que l'équation $y = m \cdot x + p$ décrit une droite constructible. Les points $(0, p)$ et $(1, m+p)$ sont constructibles d'après ce qui précède. La droite passant par ces deux points est la droite recherchée, elle est constructible.

Nous décrivons un cercle par une équation $(x - x_0)^2 + (y - y_0)^2 = K^2$ dans R .

(1) \Rightarrow (2) Soit C un cercle constructible. Si C a pour centre le point $M_c : (x_c, y_c)$ et passe par le point $M_i : (x_i, y_i)$, alors C a pour équation $(x - x_c)^2 + (y - y_c)^2 = K^2$ avec $x_0 = x_c$, $y_0 = y_c$ et $K = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$. En adoptant le lemme auxiliaire I.2-b, nous supposons vérifié que la somme, la différence, le produit, la fraction de deux nombres constructibles est un nombre constructible, ainsi que la racine carré d'un nombre constructible. Par suite, l'équation de C

s'exprime à l'aide de nombres constructibles.

(2) \Rightarrow (1) Soient x_0, y_0, K trois nombres constructibles. Alors les points M_0 de coordonnées (x_0, y_0) et $M_1(x_0, y_0 + K)$ sont constructibles d'après ce qui précède. Par suite, le cercle de centre M_0 passant par M_1 répond au problème.

[[fin de la preuve]]

b) Histoire et résultats sur les nombres constructibles.

Après les premiers résultats obtenus par les pythagoriciens sur les rapports entre nombres et géométrie (découverte de l'irrationalité de $\sqrt{2}$), il faut attendre le dix-septième siècle pour que de nouveaux résultats apparaissent sur les nombres constructibles.

En 1637, R. Descartes, après avoir montré que la géométrie est réductible à l'arithmétique, démontre que la somme, la différence, le produit, le rapport de deux distances et la racine carrée d'une distance sont constructibles à la règle et au compas (ces résultats sont repris aux figures I.2-a et I.2-b). Il montre donc que l'ensemble des longueurs obtenues rationnellement et avec l'emploi de la racine carrée à partir de longueurs données est inclus dans l'ensemble des longueurs constructibles à la règle et au compas, mais la réciproque lui échappe.

Gauss à la fin du dix-huitième siècle énonça le résultat suivant, qu'il ne démontra pas, concernant la constructibilité des polygones réguliers.

Théorème de Gauss : Les polygones réguliers constructibles à la règle et au compas sont ceux pour lesquels le nombre de cotés est de la forme $2^k p_1 \dots p_q$ où k est entier, et les p_i sont des nombres de Fermat premiers distincts.

Ce résultat détermine les angles $2\pi/n$ constructibles, c'est à dire les nombres réels de la forme $\cos(2\pi/n)$ et $\sin(2\pi/n)$ constructibles. Ce résultat fait intervenir les nombres de Fermat premiers. Que sont-ils ?

Fermat pensait avoir trouver une formule pour décrire de nouveaux nombres premiers. Il proposa les nombres, auxquels il donna son nom : $F_n = 2^{2^n} + 1$. Les premiers nombres $F_0 = 3$, $F_1 = 5$, $F_2 = 17$, $F_3 = 257$, $F_4 = 65537$ sont premiers. Par conséquent, un certain nombre de polygones réguliers sont constructibles à la règle et au compas. Le triangle équilatéral, le pentagone régulier, le polygone régulier à dix-sept cotés...

Jusqu'à Euler, personne ne réussit à factoriser un nombre de Fermat. Mais F_5 (qui vaut 4294967297) est divisible par 641 comme le découvrit Euler ($4294967297 = 641 * 6700417$). Par suite, d'autres nombres de Fermat subirent le même déclassement. Pour l'instant, il est prouvé que jusqu'à F_{20} tous les nombres de Fermat après F_4 sont des produits par conséquent il est prouvé qu'ils ne sont pas premiers. Et même plus, F_6, F_7, \dots, F_{10} ont été complètement factorisés. Très probablement, avec le temps, les autres nombres de Fermat vont connaître la même histoire.

Il faut donc attendre P.L. Wantzel en 1837, répétiteur à l'École Polytechnique et contemporain d'Évariste Galois, pour obtenir les résultats suivants. Tout d'abord la réciproque du résultat de Descartes :

Théorème : L'ensemble des nombres constructibles est le plus petit sous-corps de \mathbb{R} stable par racine carrée.

Enfin le résultat principal de Wantzel :

Théorème de Wantzel : Soit x un réel, x est constructible si et seulement si il existe une suite finie d'extensions quadratiques⁹ successives de \mathbb{Q} telle que x appartient à un élément de cette suite.

Principalement, c'est la conséquence, ci-dessous, immédiate de ce résultat qui fut utilisée, elle permet en effet de déterminer facilement les nombres non constructibles et par suite les figures géométriques constructibles à la règle et au compas :

Critère de Wantzel : Tout nombre constructible est algébrique sur \mathbb{Q} et son degré est une puissance de 2.

L'utilisation de ce critère permet de résoudre les trois problèmes de constructions cités au premier chapitre dans la section consacrée à la géométrie des constructions.

Par exemple, ce critère permet d'affirmer que la duplication du cube n'est pas possible à la règle et au compas. En effet, dupliquer un cube revient à construire la racine cubique de 2, $K = \sqrt[3]{2}$. Or K n'est pas constructible, car le degré algébrique de K est 3. Notons cependant, que en dimension 4, la duplication du cube est possible (pour peu que l'on rencontre un espace de dimension 4). En effet, en dimension 4, dupliquer le cube correspond à la construction du rapport $\sqrt[4]{2}$, qui est constructible. D'une manière générale, en dimension 2^n , la duplication du cube est possible car la construction de la racine 2^n de 2 est obtenue par n extractions successives de la racine carrée de 2.

Le second résultat important que ce critère met en évidence concerne la trisection de l'angle : en général un angle n'est pas trisectable à la règle et au compas. La démonstration de ce résultat vient de la résolution générale de l'équation du troisième degré qui introduit des racines cubiques. Un exemple d'angle non trisectable est donné par l'angle de 60° . Par conséquent, l'angle de 20° n'est pas constructible, et par voie de conséquence l'ensemble des diviseurs entiers de 20 correspond à des angles non constructibles : $1^\circ, 2^\circ, 4^\circ, 5^\circ, 10^\circ$. Cependant, l'angle de 3° est constructible. D'après ce qui précède cet angle n'est pas trisectable, car sinon l'angle de 1° serait constructible, ce qui est faux d'après ce qui précède. La constructibilité de l'angle de 3° vient de la construction du polygone régulier de 15 cotés, qui par dédoublement donne les polygones réguliers de 30, 60 et 120 cotés et par conséquent l'angle de 3° .

Enfin, le dernier problème de constructibilité résolu par ce critère concerne la quadrature du cercle ou la recherche d'une construction pour π . Pour nous aujourd'hui, le résultat est évident, puisque nous savons que π et $\sqrt{\pi}$ (qui plus précisément est équivalent à la quadrature

⁹ c'est à dire par des racines carrées.

du cercle) sont des nombres transcendants. Par conséquent l'ordre de π et $\sqrt{\pi}$ sur \mathbb{Q} n'est pas défini, et par suite π et $\sqrt{\pi}$ sont non constructibles à la règle et au compas. Il fallut attendre cependant longtemps après Wantzel pour arriver à ce résultat auquel ont contribué Lambert, Legendre et enfin Lindermann.

Malheureusement pour le critère de Wantzel, proposition plus simple que son théorème, la réciproque est fautive. Il existe des réels algébriques sur \mathbb{Q} dont le degré est une puissance de 2 non constructibles. Par exemple les racines de X^4+X+1 ne sont pas constructibles alors qu'ils ont bien un degré qui est une puissance de 2.

La réciproque du critère de Wantzel s'exprime en termes quelque peu plus compliqués, soit en reprenant le théorème de Wantzel, soit en utilisant la théorie de Galois : «La condition de constructibilité de z à partir de \mathbb{Q} équivaut à l'existence d'une extension de \mathbb{Q} contenant z et déduite de \mathbb{Q} par extensions quadratiques successives. [...] La traduction en terme de groupes de Galois part du polynôme minimum de z sur \mathbb{Q} , dont le groupe de Galois doit posséder une chaîne de sous-groupes chacun d'indice 2 dans le précédent : l'ordre de ce groupe doit être puissance de 2.» in [Carrega-89].

Après Wantzel et les solutions aux problèmes de constructions qu'a entraîné le critère de constructibilité qu'il a mis à jour, il y eut peu de résultats novateurs.

Il y a eu des résultats d'équivalence entre la constructibilité règle-compas, la constructibilité compas seul, et la constructibilité à partir de la règle seule et d'un cercle unitaire.

Citons également la démonstration de la non équivalence entre la géométrie règle-compas et de la géométrie règle-bissecteur. C'est Hilbert qui a été le premier à montrer l'inclusion stricte de la géométrie règle-bissecteur dans la géométrie règle-compas. Les nombres de la géométrie règle-bissecteur ne sont pas tous les nombres constructibles, ce sont les nombres euclidiens (de la forme $\sqrt{1+x^2}$ avec x euclidien). L'inclusion simple entre géométrie règle-bissecteur et règle-compas ou entre nombres euclidiens et nombres constructibles est immédiate puisque la règle et le compas permettent de construire la bissectrice d'un angle. Une démonstration algébrique assez rapide de l'inclusion stricte est donnée dans [Carrega-89], utilisant les résultats de Wantzel.

Notons enfin l'explicitation par Lebesgue d'un algorithme pour la construction de figures géométriques à la règle et au compas, cf. [Lebesgue-89]. Malheureusement, l'algorithme de Lebesgue en utilisant une résolvante de Galois est de complexité algorithmique doublement exponentielle.

c) Introduction de la notion de complexité algébrique.

Jusqu'ici, la notion de constructibilité a été le centre d'intérêt principal. À partir de maintenant, et jusqu'à la fin, ce sera la notion de complexité qui sera la plus importante. Elle est en effet plus proche des préoccupations algorithmiques qui nous concernent.

La complexité algébrique est apparue dans le résultat de Wantzel dans le degré algébrique. Sans augurer de la suite, sur le plan de la complexité des calculs et des représentations des données, il paraît clair qu'un nombre constructible sera d'autant plus coûteux à manipuler qu'il

possède un degré algébrique élevé. Aussi allons nous dans cette section nous familiariser avec la complexité algébrique des figures géométriques, à l'aide d'exemples.

Nous supposons que nous avons choisi un système de coordonnées qui correspond à B_G , que les points s'expriment avec deux coordonnées dans le repère associé à B_G , que les droites sont représentées par des équations de la forme $y=m.x+p$, et les cercles par des équations de la forme $(x-x_c)^2+(y-y_c)^2=K$. (attention ici et dans la suite, c'est K et non K^2 utilisé classiquement. Ce choix réduit la complexité algébrique, et permet d'évoluer à complexité constante comme nous allons le voir).

Définition I.2-e : La complexité algébrique d'une figure, est le degré de la plus petite extension algébrique de Q qui permet de définir toutes les coordonnées et équations des objets de la figure. (La définition est consistante car B_G et le système de coordonnées sont fixés.)

Est-il nécessaire de donner quelques rudiments sur les nombres algébriques, les racines carrées, les degrés, l'évolution des degrés dans un domaine de rationalité ?

Comme **B.A. BA** voici le résultat principal :

Toute fonction rationnelle de nombres appartenant à une extension algébrique donnée appartient à cette même extension.

Autrement dit, tant que des calculs avec les quatre opérations élémentaires sont effectués sur des coordonnées de la figure, la complexité de la figure est constante. Et le pendant : dès qu'une racine carrée est extraite, la complexité de la figure peut augmenter (doubler).

En pratique, si a, b sont des éléments de K extension algébrique de Q , alors $a+b, a-b, a * b$ et a / b sont dans K , mais \sqrt{a} peut ne pas appartenir à K . En général, \sqrt{a} n'appartient pas à K .

Constructions avec des droites.

La construction d'une droite à partir de deux points, ou de l'intersection de deux droites, n'augmente pas la complexité algébrique de la figure :

Proposition I.2-c : Les procédures de construction à la règle seule sont de complexité constante.

[[preuve]]

En effet, comme nous l'avons vu dans la démonstration de l'équivalence entre constructibilité géométrique et nombres constructibles, en notant $M_0=(x_0,y_0)$ et $M_1=(x_1,y_1)$ les coordonnées de deux points, alors l'équation de D passant par ces deux points est donnée par $y=m.x+p$ avec $m = (y_0-y_1)/(x_0-x_1)$ et $p = (x_0.y_1-x_1.y_0)/(x_0-x_1)$. Nous observons que m et p sont fonctions rationnelles des coefficients de la figure. Par suite la construction de la droite s'effectue à complexité algébrique constante.

De même, l'intersection $M(x_0, y_0)$ des deux droites $D:y=m.x+p$ et $D':y=m'.x+p'$ a pour coordonnées : $x_0=(p-p')/(m'-m)$, $y_0=(p.m'-p'.m)/(m'-m)$. Nous observons que x_0 et y_0

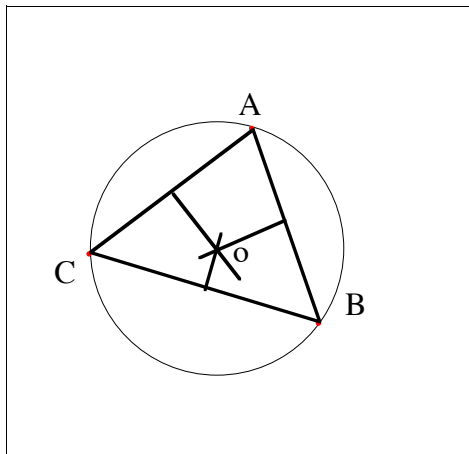


Figure I.2.c: Construction du cercle circonscrit.

Construction du cercle circonscrit aux trois points A, B, C :

- prendre la médiatrice de [AB]
- prendre l'intersection avec la médiatrice de [BC]
- c'est le centre du cercle recherché.

Il appartient au domaine de rationalité de la figure initiale car chaque construction (milieu et perpendiculaire pour les médiatrices, intersection de droite et construction d'un cercle de centre et de rayon donné) s'effectue à complexité constante.

sont fonctions rationnelles des coefficients de la figure. Par suite, la construction de l'intersection de deux droites s'effectue à complexité algébrique constante.

[[fin de la preuve]]

Constructions avec des cercles.

C'est moins évident que le point précédent, puisqu'un cercle intervient ici, mais le résultat suivant est valide :

Proposition I.2-c : La procédure de construction du cercle circonscrit à trois points n'augmente pas la complexité algébrique de la figure.

[[preuve]]

Le cercle passant par trois points d'une figure de complexité fixée est de cette complexité. En bref, le triangle formé par ces trois points permet de définir trois médiatrices (de même complexité), leur point d'intersection (de même complexité). Le cercle de centre ce point et passant par un sommet du triangle est de même complexité, c'est le cercle recherché.

Montrons que l'équation de la médiatrice d'un segment est fonction rationnelle des points. Si M_0 et M_1 sont deux points, les coordonnées du milieu sont fonctions rationnelles des coordonnées de M_0 et M_1 . Par ailleurs, la construction d'une perpendiculaire à une droite est aussi trivialement rationnelle. Par suite, la médiatrice est rationnelle.

Il reste à montrer que l'équation du cercle défini par son centre et un point est rationnelle. C'est une évidence avec la convention prise pour les équations de cercle. Si le centre est $M_0(x_0, y_0)$ et le point courant $M_1(x_1, y_1)$ alors le cercle a pour équation $(x-x_c)^2+(y-y_c)^2=K$, avec $x_0=x_c$, $y_0=y_c$ et $K=(x_0-x_1)^2+(y_0-y_1)^2$. Le choix de K au lieu de K^2 dans la convention des équations de cercles trouve ici sa justification. Si K^2 avait été choisi, il faudrait ajouter $\sqrt{(x_0-x_1)^2+(y_0-y_1)^2}$ ce qui augmenterait la complexité algébrique de la figure. Et cette augmentation s'opérerait à chaque nouvelle construction de cercle, augmentation que nous évitons ici.

[[fin de la preuve]]

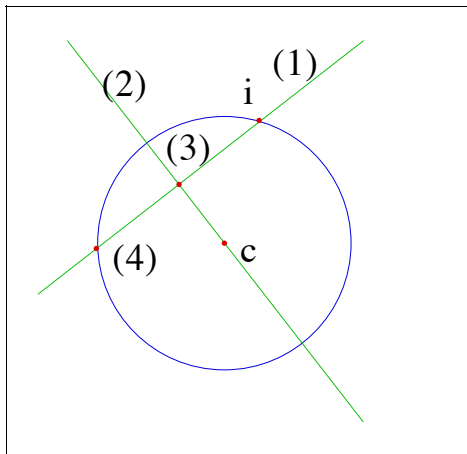


Figure I.2.d: Construction d'un point rationnel sur un cercle rationnel.

Soit un cercle rationnel défini par c et i rationnels. Comment construire un point arbitraire sur ce cercle qui soit rationnel ?

(1) Prendre une droite arbitraire rationnelle passant par i ,

(2) construire la perpendiculaire à (1) passant par c , (2) est rationnelle,

(3) est l'intersection de (1) et (2) c'est un point rationnel,

(4) le symétrique de (1) par rapport à (3) est sur le cercle, c'est un point rationnel.

Quand (1) parcourt le plan, (4) parcourt le cercle.

Un résultat aussi important que le précédent est donné par la proposition suivante. Il porte sur la construction symétrique : placer un point sur un cercle.

Proposition I.2-e : La procédure de construction de la figure I.2.d donne un point arbitraire sur le cercle sans augmenter la complexité de la figure.

[[preuve]]

voir figure I.2.c et section III.2-b

[[fin de la preuve]]

Les deux propositions précédentes permettent d'obtenir la conclusion étonnante suivante :

Résultat I.2-f : Un cercle quelconque du plan (rationnel ou non) possède 0,1,2 ou une infinité de points rationnels.

[[preuve]]

En effet si un cercle possède 3 points rationnels alors c'est un cercle rationnel d'après la première proposition. D'après la seconde proposition, il est possible de construire des points rationnels arbitraires sur le cercle.

[[fin de la preuve]]

Attention ce qui précède ne signifie pas qu'un cercle rationnel soit constitué de points rationnels uniquement. Ce n'est pas le cas. Comme pour la droite, avec des rationnels seulement le cercle est plein de trous (les réels).

Construction d'un triangle équilatéral.

Jusqu'ici, toutes les constructions se sont effectuées à complexité algébrique constante. Ce n'est plus le cas pour la construction d'un triangle équilatéral.

En effet, soient $M_1 = (x_1, y_1)$ et $M_2 = (x_2, y_2)$ deux points, alors les coordonnées du sommet M_3 du triangle équilatéral sont $M_3 = (x_3 = (x_1 + x_2)/2 \pm \sqrt{3} \cdot (y_1 - y_2)/2, y_3 = (y_1 + y_2)/2 \pm \sqrt{3} \cdot (x_2 - x_1)/2)$. Par conséquent, il y a introduction de la racine carrée de trois. Ce qui nous amène à formuler le résultat suivant, qui nous servira dans la suite.

Proposition I.2-g : Un triangle équilatéral est toujours défini dans une extension algébrique de \mathbb{Q} contenant $\sqrt{3}$.

Nous avons cependant une autre proposition évidente qui nous permet de parler de construction à complexité algébrique constante, même avec la construction d'un triangle équilatéral.

Proposition I.2-h : Il n'y a pas d'augmentation de la complexité algébrique d'une figure par la construction de nouveaux triangles équilatéraux à partir du moment qu'il en existe un dans la figure.

[[preuve]]

En effet, deux triangles équilatéraux successifs partagent cette même propriété de nécessiter de $\sqrt{3}$. Par suite, chaque nouveau triangle équilatéral trouve sa place dans l'extension algébrique courante dès que celle-ci possède $\sqrt{3}$, introduit par un premier triangle équilatéral.

[[fin de la preuve]]

Les polygones réguliers.

Nous avons déjà vu précédemment quels sont les polygones réguliers constructibles à la règle et au compas. Pour cela, il suffit d'appliquer le critère de Gauss (cf. chapitre I-1). Il y a une seconde méthode heuristique, comme nous allons le voir.

En effet, nous pouvons observer sur la figure I.2-e que le triangle de Pascal, modulo deux, donne, si chaque ligne est prise comme la décomposition d'un nombre en binaire, les nombres suivants : 1, 3, 5, 15, 17, 51, 85, 255, 257, ... Alors que les premiers polygones réguliers constructibles à la règle et au compas ayant un nombre impair de cotés sont ceux de 3, 5, 15, 17, 51, 85, 255, 257, ... cotés.

Cette coïncidence est troublante. En est-il toujours de même ? Si la $n^{\text{ième}}$ ligne du triangle de Pascal correspond à l'écriture en base deux de K_n , le polygone régulier à K_n cotés est-il constructible ?

Malheureusement, la réponse est non. Le triangle de Pascal modulo deux coïncide avec les produits des nombres de Fermat, en effet les premières lignes sont, 3 ($=F_0$), 5 ($=F_1$), 15 ($=F_0 * F_1$), 17 ($=F_2$), 51 ($=F_0 * F_2$), 85 ($=F_1 * F_2$), 255 ($=F_0 * F_1 * F_2$), 257 ($=F_3$). Et la correspondance continue. Par suite, tant que les facteurs F_i sont des nombres de Fermat premiers, le nombre produit correspond à un polygone régulier constructible, mais comme nous l'avons vu, (après Euler) F_5, F_6, \dots, F_{20} et très certainement les suivants ne sont pas premiers¹⁰. Et donc, à partir de la 32^{ième} ligne dans le triangle de Pascal qui correspond à F_5 , la septième avant la fin sur la figure I.2-e, la correspondance avec les polygones réguliers constructibles n'est plus réalisée.

Rappelons que d'autres polygones réguliers existent, par dédoublement avec des bissectrices. Mais avec un nombre impair de cotés, il semble donc qu'il n'y ait que ceux décrits par les 39 premières lignes du triangle de Pascal modulo deux.

¹⁰ La question suivante est ouverte "Les nombres de Fermat premiers sont-ils en quantité infinie"

cf. <http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Fermat.html>

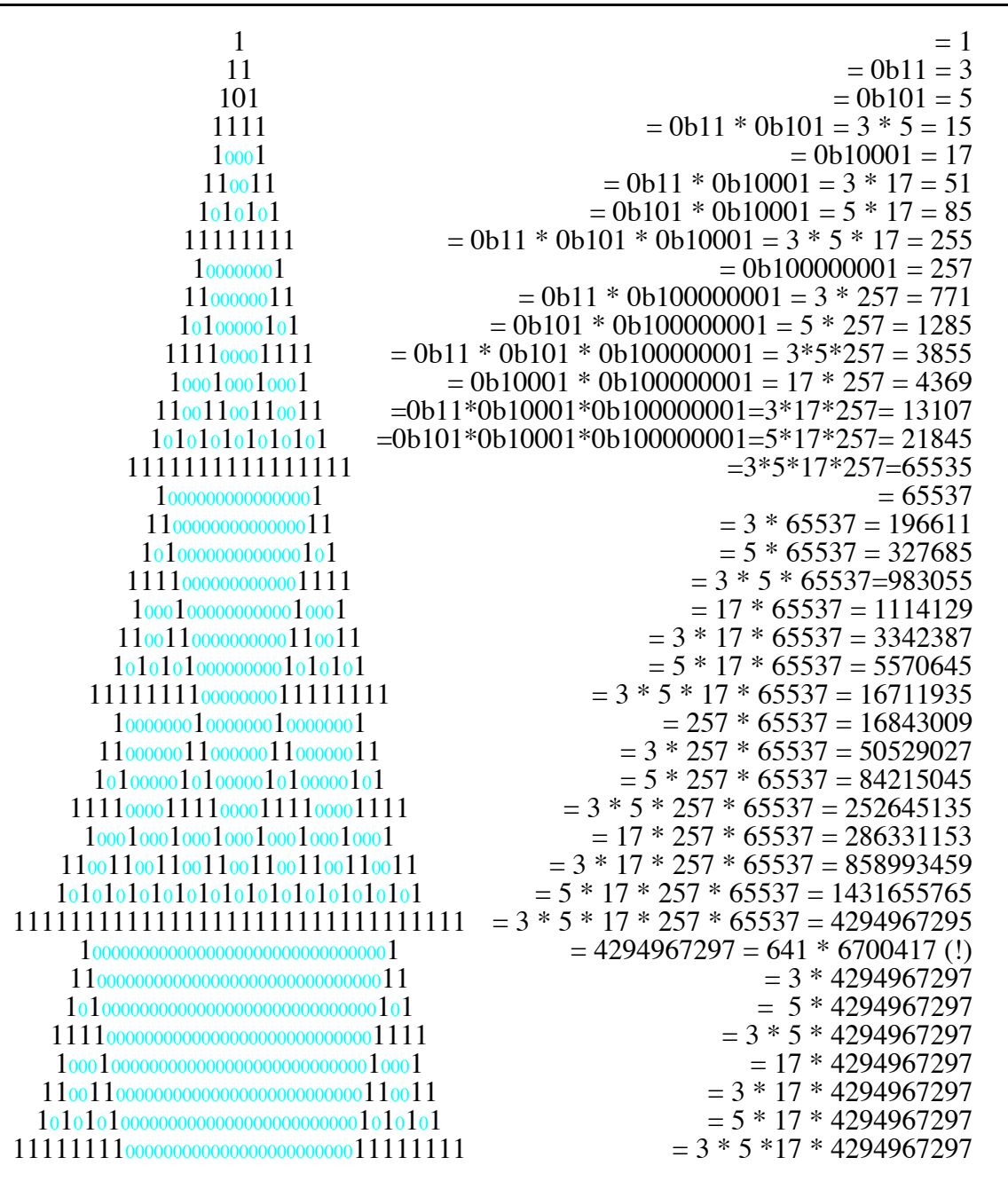


Figure I.2-e : 40 premières lignes du triangle de Pascal Modulo 2.

Remarque : notre propos se rapproche de l'algèbre, observons la puissance des logiciels de calculs formels dans ce domaine : Mathematica factorise bien les premiers nombres de Fermat, par exemple $F_5 = 4294967297 = 641 * 6700417$, et $F_6 = 18446744073709551617 = 274177 * 67280421310721$. Au delà, l'algorithme de factorisation "élémentaire" de Mathematica ne suffit plus. Mais avec d'autres moyens, et d'autres algorithmes, comme nous l'avons déjà mentionnée, jusqu'à F_{10} , les nombres de Fermat sont complètement factorisés.

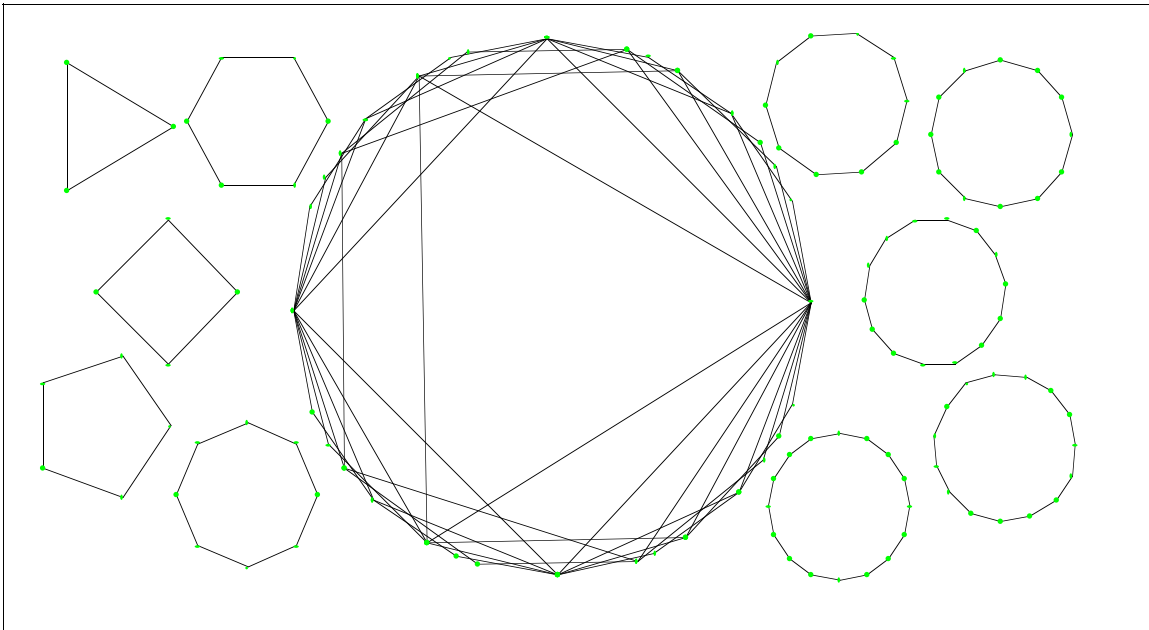


Figure I.2-f : Polygones réguliers constructibles.

Jusqu'à F_9 , Mathematica sait dire, à l'aide d'algorithmes probabilistes, que ces nombres ne sont pas premiers.

Mathematica ne sait pas factoriser F_7 et les suivants, mais il peut encore les donner. Par exemple : $F_{10}=179769313486231590772930519078902473361797697894230657273430081157732675805500963132708477322407536021120113879871393357658789768814416622492847430639474124377767893424865485276302219601246094119453082952085005768838150682342462881473913110540827237163350510684586298239947245938479716304835356329624224137217$.

Mathematica échoue dans l'évaluation des nombres de Fermat à partir du 17^{ième} terme. Ce nombre aurait 40000 chiffres environ ($2^{17} = 131072$, $\text{Log}_{10}(2^{131072}) \approx 40000$). Avec 80 caractères par ligne, l'écriture de ce nombre nécessiterait 500 lignes, soit avec 50 lignes par pages, 10 pages de chiffres.

Pour clore cette étude sur les polygones réguliers, notons, sur la notion de complexité algébrique les degrés des extensions algébriques nécessaires à l'expression des coordonnées des points des polygones réguliers.

La seconde ligne du tableau suivant contient la complexité algébrique de la figure constituée du polygone régulier à n cotés, n étant le nombre apparaissant dans la case au dessus de ce nombre sur la première ligne (les / dans le tableau indiquent les polygones réguliers non constructibles à la règle et au compas).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	1	0	2	1	/	1	/	2	/	2	/	/	3	2	4	/	/	4

Pour les polygones réguliers à 3 et 6 cotés, il faut $\sqrt{3}$; pour ceux à 5 et 10 cotés, il faut respectivement $\sqrt{5}$ (ce qui correspond au corps \mathbb{Q} [nombre d'or]) et $\sqrt{10+2\sqrt{5}}$; pour 15 cotés, il faut les trois nombres constructibles précédents; pour 8 cotés, il faut $\sqrt{2}$. Ainsi, avec les racines carrées de 2, 3, 5, et $\sqrt{10+2\sqrt{5}}$, les pentagones réguliers constructibles jusqu'à dix cotés sont exprimables.

Conclusion de ce chapitre:

Comme le montre cette partie située entre histoire, géométrie et mathématiques, la recherche sur les nombres constructibles est arrivée à de puissants formalismes et résultats qui ont contribué à la résolution de problèmes de constructions vieux de deux mille ans (quadrature du cercle, trisection de l'angle, duplication du cube). Nous retiendrons pour la suite la caractérisation des nombres constructibles comme éléments d'extensions algébriques quadratiques de \mathbb{Q} . Nous noterons le manque d'informations théoriques et pratiques sur le degré algébrique représentatif de la complexité des éléments algébriques de ces nombres. Les quelques outils obtenus ici sont d'ordre formel. Leur utilisation est limitée à l'expertise humaine et nécessite une grande force de calcul pour leur mécanisation.

I-3. Constructions géométriques élémentaires, nombres constructibles et complexité.

Introduction :

L'objectif de ce chapitre est de considérer la notion de nombre constructible d'un point de vue pratique. Cette étude est effectuée avec pour critère de référence les calculs qu'un ordinateur va devoir entreprendre pour réaliser des constructions géométriques élémentaires (intersection, perpendiculaire, milieu,...).

Sans augurer de la suite sur la faisabilité, la précision et la complexité des calculs et des représentations des nombres constructibles, il paraît clair qu'un nombre constructible est d'autant plus difficile à manipuler qu'il fait intervenir plus d'extensions algébriques quadratiques, c'est à dire que son degré algébrique est grand. Le degré algébrique représentant assez fidèlement une mesure de cette complexité, nous considérons avec attention celui-ci dans ce chapitre et le suivant.

Nous allons étudier les rapports entre les figures, et surtout les constructions géométriques élémentaires usuelles, et la complexité algébrique des nombres qu'elles introduisent.

Pour la précision des calculs, considérant que c'est un problème uniquement lié à l'informatique et l'arithmétique des ordinateurs, nous n'en ferons que de courtes mentions dans cette partie dédiée à la géométrie. Rappelons cependant que l'arithmétique usuelle, dite à virgule flottante, n'est pas exacte. Des exemples sont fournis à la section III.2-a. Aussi, un choix important doit être fait pour ou contre cette arithmétique. Le coût lié au choix d'une arithmétique exacte, comme nous le préconisons à la partie II, est directement dépendant de la complexité étudiée ici.

Parallèlement, nous observons l'importance du choix des outils de construction (règle, équerre, compas, bissecteur, etc), et leur réelle sur le plan de la complexité algébrique. De même, nous considérons les constructions géométriques usuelles comme de nouveaux outils de construction (ou "macro-construction"). Nous en profitons pour nous interroger sur le statut des objets intervenant dans une macro-construction (résultats finaux, résultats intermédiaires ?) et de l'intérêt d'une compilation de ces constructions géométriques.

Enfin, nous essayons de trouver une (impossible) correspondance entre spécifications géométriques et complexité algébrique. Une correspondance simple n'existant pas entre spécifications géométriques et figures constructibles, nous montrons qu'il en est de même entre spécification et complexité grâce à deux exemples.

a) Constructions rationnelles.

Définition.

Parmi les constructions géométriques usuelles, nous définissons les constructions rationnelles :

Définition I.3-a : Une construction géométrique est rationnelle si les coordonnées de tous les points introduits appartiennent à l'extension algébrique de la figure de départ.

Ces constructions s'opèrent donc à degré de complexité algébrique constant. Elles sont rationnelles au sens de Lebesgue : elles conservent le corps initial, le domaine de rationalité de la figure de départ pour l'expression des coordonnées des nouveaux objets.

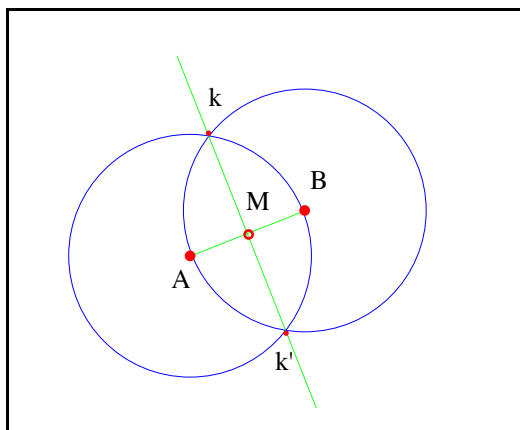
Ce n'est pas le caractère rationnel ou non des coordonnées des résultats qui détermine si un procédé de construction est ou n'est pas rationnel. Les objets construits sont rationnels si la figure de départ est elle-même rationnelle.

Nous insistons particulièrement sur ces constructions, car ce sont a priori les constructions les moins coûteuses, celles que nous voulons favoriser lors d'optimisations.

Il faut distinguer cependant deux notions sous-jacentes, et ce dans tout ce chapitre. Il y a la construction (c'est à dire son résultat uniquement) et la procédure de construction (faisant intervenir tous les objets intermédiaires éventuels). Une construction dite rationnelle peut, selon le choix d'une procédure de construction particulière, nécessiter des intermédiaires de calcul non rationnels. C'est un point que nous allons étudier conjointement.

L'intérêt de cette étude des procédés de construction vient de ce que les micro mondes de géométrie introduisent une notion qui s'en approche : les macro-constructions. Cette notion est équivalente à celle de fonction dans les langages fonctionnels, et à celle de prédicat dans les langages déclaratifs. Une macro-construction comporte des résultats intermédiaires et des résultats finaux. La question se pose de la compilation de ces macro-constructions en vue de diverses optimisations (gain en temps de construction, en place mémoire, etc...). Cette compilation peut-elle, doit-elle prendre en compte l'élimination des résultats intermédiaires ? Et une éventuelle simplification algébrique des résultats finaux par rapport aux résultats intermédiaires ? Un exemple est donné à la section suivante avec la construction du milieu d'un segment.

Notons lors de la définition d'un logiciel de géométrie, qu'il est possible d'employer un grand nombre de macro-constructions (par exemple à travers une démarche partant d'un ensemble minimal de constructions élémentaires incidence et ordre; point, droite et cercle; et d'un procédé d'extension par macro-construction). Dans cette optique, l'importance d'une vision claire sur la complexité des calculs, apparaît clairement. Le choix délicat de l'ensemble des constructions initiales en dépend, l'intérêt d'une compilation également.



Procédé de construction classique du milieu de $[A, B]$ à la règle et au compas :

- 1- tracé du cercle centré en A passant par B,
- 2- tracé du cercle centré en B passant par A,
- 3- construction de k et k' points d'intersection des deux cercles
- 4- tracer la droite reliant k et k'
- 5- tracer le segment reliant A et B
- 6- $M =$ intersection du segment et de la droite. M est le milieu.

Figure I.3-a: Construction du milieu d'un segment.

Constructions rationnelles.

Milieu.

Parmi les constructions et procédés de constructions rationnelles usuelles, il y a bien sûr, pour ne citer que les plus simples, l'intersection de deux droites, la construction d'une parallèle, ou d'une perpendiculaire. Nous avons donné la preuve de la rationalité de l'intersection précédemment. Pour la construction d'une parallèle ou d'une perpendiculaire, les preuves sont du même type, immédiates. Il y bien sûr d'autres constructions rationnelles que nous verrons plus loin. Pour l'instant, intéressons nous à la construction du milieu d'un bipoint.

Proposition I.3-a : La construction du milieu d'un bipoint est une construction rationnelle
[[preuve]]

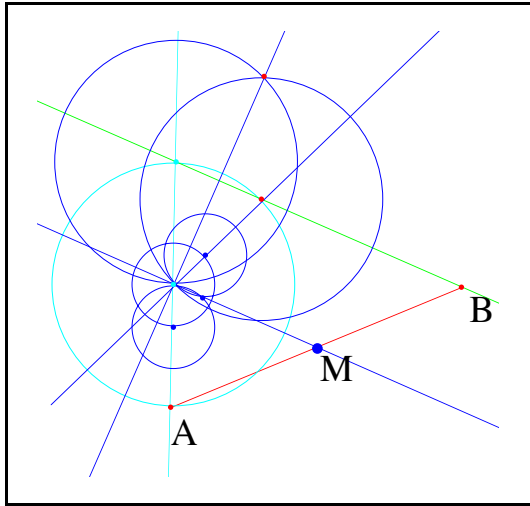
si $M: (x, y)$, $M': (x', y')$ le milieu de $[M, M']$ a pour coordonnées $((x+x')/2, (y+y')/2)$

[[fin de la preuve]]

Cependant, à la règle et au compas, les procédures de construction d'un milieu ne sont pas nécessairement rationnelles. Par exemple, la procédure de construction proposée (cf. figure I.3-a) du milieu M de $[A, B]$ introduit deux points k et k', sommets d'un triangle équilatéral, dont les coordonnées sont dans $\mathbb{Q}[\sqrt{3}]$.

Pour trouver une procédure de construction à la règle et au compas du milieu d'un segment en n'introduisant que des points aux coordonnées rationnelles, l'effort n'est pas négligeable¹¹. Il faut donc bien différencier les résultats d'une construction (comme le milieu

¹¹ Si on ajoute comme outil de construction le traceur de corde commune (outil rationnel), la procédure de construction de la figure I.3-a permet d'avoir un procédé de construction rationnel facilement.



Procédure de construction rationnelle du milieu de $[A,B]$ à la règle et au compas.

Idée de la construction : c'est une construction basée sur le théorème de Thalès.

Après construction d'un segment quelconque et de son double, la difficulté revient ensuite à construire des parallèles à la règle et au compas, en conservant le domaine de rationalité.

C'est possible, mais cela demande un certain effort.

[[cf. annexe 1.]]

Figure I.3-b: Construction rationnelle du milieu d'un segment.

d'un segment) avec la procédure de construction elle-même, qui peut introduire des points intermédiaires non rationnels.

Un exemple de procédure de construction du milieu d'un segment à la règle et au compas uniquement rationnelle est donnée Figure I.3-b. Afin d'éviter des triangles équilatéraux, l'idée de départ est une construction basée sur le théorème de Thalès. Après construction d'un segment et de son double, (ce qui est simple), la difficulté revient ensuite à construire des parallèles à la règle et au compas, en conservant le domaine de rationalité. C'est possible, mais cela demande un certain effort. Existe-t-il une solution plus simple, éventuellement partant d'une autre idée ?

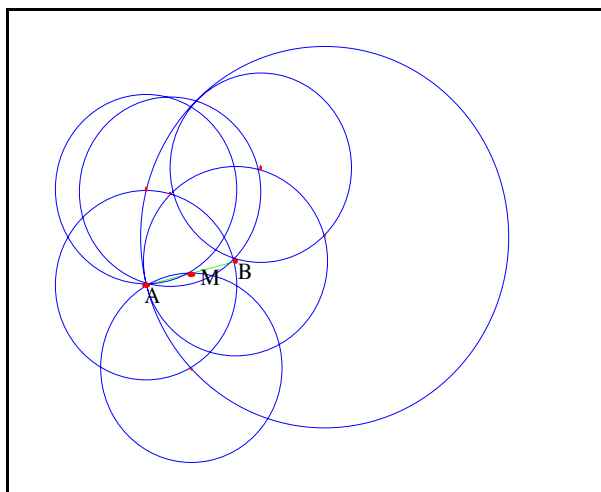
Il semble apparaître un nouvel exercice géométrique dont voici les règles : étant donné un certain nombre d'outils pris parmi la règle, le compas, le traceur de parallèle ou de perpendiculaire, le bissecteur, etc ... trouver une procédure de construction rationnelle, ou la procédure de construction géométrique la moins coûteuse algébriquement, d'une figure géométrique donnée. Cet exercice rappelle par certains côtés la recherche des outils minimaux de construction géométrique qui a mené, pour rappel, aux résultats suivants :

Théorème : Il y a équivalence de constructibilité avec les éléments suivants :

- (1) la règle + le compas
- (2) le compas seul
- (3) la règle plus un cercle

Ainsi, puisqu'il est possible de construire le milieu d'un segment en restant rationnel avec la règle et le compas, est-ce possible avec le compas seul ?

La construction du milieu d'un segment avec le compas seul est possible, un exemple est donné par la procédure de construction de la figure I.3-c, mais elle nécessite deux extensions dont $\sqrt{3}$. N'y a-t-il pas moyen de rester rationnel ? Une étude plus approfondie des



Procédure de construction
du milieu d'un segment $[A,B]$ au
compas seul.

Idée de la construction :
[[cf. annexe 2.]]

**Figure I.3-c: Construction du milieu d'un segment
au compas seul.**

constructions rationnelles avec le compas seul est effectuée plus loin, lors de l'étude de la résolution du problème de Napoléon : "Construire le centre d'un cercle avec le compas seul".

Ce type d'exercice trouve sa justification en géométrie (comme la recherche des outils minimaux de construction) mais aussi, et surtout, en informatique. Parmi ces procédés de construction minimale ceux qui sont rationnels peuvent être correctement représentés par les arithmétiques des ordinateurs à moindre coût (arithmétique rationnelle en précision infinie).

L'informatique dicte cet exercice de recherche de procédure de constructions rationnelles. En même temps, cet exercice nous informe sur la structure algébrique des problèmes de construction.

Dans le cadre de la démonstration automatique de théorèmes par ordinateur, ces procédures de construction semblent les plus "sympathiques" au sens où elles présentent les figures géométriques sous un jour semble-t-il plus "simple". Peut-être, ces exercices nous indiquent-ils pourquoi de si puissants résultats en démonstration automatique sont obtenus malgré l'utilisation d'outils très coûteux (base de Gröbner). La possible "rationalisation" des procédures de construction est corrélée à une linéarisation des calculs¹². Et ces linéarisations favorisent, semble-t-il, les calculs des bases de Gröbner, ou les calculs des ensembles caractéristiques de Ritt dans les méthodes de démonstration automatique issues des travaux de Wu.

Autres constructions rationnelles

Parmi les constructions rationnelles, il y a la construction d'un symétrique. Ici aussi, la procédure de construction usuelle d'un symétrique au compas qui procède avec trois triangles équilatéraux nécessite une extension algébrique par $\sqrt{3}$ (triangle équilatéral). Y a-t-il une construction au compas qui soit rationnelle ?

¹²Discussions informelles avec Dongming Wang.

Poursuivons plus, la construction d'une médiatrice est rationnelle, comme composée de la construction d'une perpendiculaire et de la construction d'un milieu. Et nous retrouvons le résultat déjà présenté : la construction d'un cercle passant par trois points donnés est rationnelle. Nous avons également déjà vu que prendre un point sur un cercle est une construction rationnelle.

Ensuite, viennent quelques résultats sur la construction d'un objet ayant presque la même spécification qu'un objet déjà construit. Ainsi, la construction de l'intersection d'une droite et d'un cercle (ou de deux cercles) quand la droite (ou un cercle) passe déjà par un point connu du cercle, c'est à dire quand l'on connaît déjà une intersection. Il s'agit alors de construire l'autre intersection. La construction d'une seconde bissectrice, quand la première bissectrice est connue, est rationnelle puisqu'il s'agit de construire la perpendiculaire à la première bissectrice passant par le point d'intersection des deux droites.

Étrangement peut être, la construction de la corde commune à deux cercles est une construction rationnelle. Ce peut être un résultat surprenant, car les points d'intersection de deux cercles, qui appartiennent à la corde commune (droite rationnelle), n'ont pas de raison d'être rationnels. Par conséquent, il n'est pas immédiat d'avoir une procédure de construction rationnelle pour réaliser cette construction.

Enfin, le calcul de l'inverse d'un point par rapport à un cercle est une construction rationnelle, alors que les procédés de construction que je connais ne sont pas rationnels¹³. Ici particulièrement, la possibilité de compiler la construction en réduisant la complexité algébrique ou le choix de cette construction comme construction élémentaire réduit la complexité algébrique des figures obtenues en favorisant la rationalité des constructions.

Un récapitulatif des constructions est donné en I.3-c.

b) Constructions géométriques non-rationnelles.

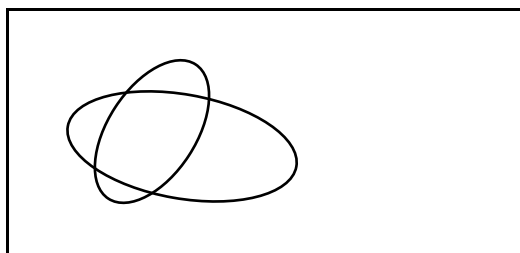
Nous définissons naturellement les constructions non-rationnelles parmi les constructions géométriques usuelles par opposition aux constructions rationnelles :

Définition I.3-b : Une construction géométrique est non-rationnelle si elle introduit dans la figure un objet au moins dont les coordonnées nécessitent une ou plusieurs extensions quadratiques supplémentaires par rapport au corps de la figure initiale.

Par suite, une construction est soit rationnelle, soit non rationnelle. Quant aux procédés de construction associés, ils ne peuvent être rationnels que quand la construction l'est elle-même.

En fait, parmi les constructions usuelles à la règle et au compas (intersection cercle-cercle, intersection droite-cercle, bissectrice), les constructions non-rationnelles introduisent au plus une seule extension algébrique. Cependant, notons que les procédures de construction peuvent nécessiter plusieurs extensions algébriques. Rappelons que c'est déjà vrai pour les

¹³Si on ajoute l'outil de construction "traceur de corde commune" (outil rationnel), il existe des procédures de construction rationnelles.



Intersection entre coniques :

En général, deux coniques s'intersectent en quatre points distincts.

Figure I.3-d: Intersection entre coniques.

constructions rationnelles. Notons également que les constructions avec des traceurs d'ellipse introduisent deux extensions algébriques. Mais les ellipses dépassent le cadre que nous nous sommes fixés (règle-compas).

Parmi les constructions non-rationnelles, il y a, nous l'avons dit, la construction de l'intersection d'une droite et d'un cercle, ou de deux cercles, ou la construction de la bissectrice d'un angle. En particulier, l'intersection d'une horizontale ou d'une verticale quelconque avec un cercle, est une construction non-rationnelle. Ce dernier point, donné ici à titre d'exemple, peut sembler d'un intérêt mineur. Effectivement, nous ne le développons pas plus dans cette partie I sur la géométrie. En III.2-b, où cette construction a une certaine importance, il est développé plus amplement et une preuve de l'affirmation donnée ci dessus est proposée.

En fait, les intersections considérées précédemment ne visent pas la détermination d'une, mais plus exactement de deux intersections entre une droite et un cercle, ou des deux intersections entre deux cercles. De même la construction usuelle d'une bissectrice permet l'obtention des deux bissectrices d'un angle donné. Sans que cela soit une règle absolue, la notion de construction non-rationnelle est corrélée avec la notion de double.

Pour une explication informelle, une construction non-rationnelle est associée à l'introduction d'une extension quadratique, qui sous-entend l'existence d'une équation du second degré à résoudre et qui introduit ces deux racines carrées. Comme cette équation du second degré donne le plus souvent deux solutions, l'une conjuguée de l'autre, la notion de double est introduite. Par suite, les constructions non-rationnelles donnent le plus souvent deux résultats qui correspondent à deux cas de figure.

Est-ce à dire qu'une procédure de construction ayant deux cas de figure présente une extension quadratique ? En général la réponse est oui. Nous rapprochons ce résultat de celui que l'on obtient pour les constructions rationnelles :

Proposition I.3-b : Toutes les constructions rationnelles donnent un seul résultat.

Il entre dans ce cadre "rationnel" les constructions citées à la fin de la précédente section : construction de l'autre intersection d'un cercle et d'une droite, ou de deux cercles, et construction de l'autre bissectrice. Rappelons que ces trois constructions, non-rationnelles a priori, deviennent rationnelles si l'une des solutions est déjà connue, et s'il s'agit de construire seulement l'autre. Ces constructions restent non-rationnelles quand les deux solutions sont inconnues.

Nous pouvons également rapprocher ce résultat du calcul des intersections entre coniques, comme par exemple entre deux ellipses. Dans le cas général, il y a 4 solutions aux équations. Par suite, il y a très certainement deux extensions quadratiques introduites, ou une extension algébrique de degré quatre. Dans certains cas particuliers, intersection d'une parabole avec cercle (ce qui permet de résoudre le problème de la duplication du cube), une extension algébrique de degré trois peut aussi apparaître. Dans le cas où il y a introduction d'une extension algébrique de degré quatre non décomposable, ou d'une extension algébrique de degré trois, les nombres obtenus ne sont pas constructibles à la règle et au compas. Ceci explique la restriction que nous avons imposée (règle, compas). D'autres raisons apparaîtront dans la partie II pour légitimer ces restrictions. (existence de représentations exactes et normales).

Le premier résultat obtenu sur les constructions non-rationnelles élémentaires est donc le suivant :

Proposition I.3-c : Le cas général d'une intersection entre une droite et un cercle, est non-rationnel. Cette construction introduit une seule extension algébrique quadratique.

[[preuve]]

Dans le cas général d'une intersection entre une droite et un cercle, les équations suffisent à donner le résultat recherché : il y a introduction au pire d'une extension quadratique.

Soit un cercle quelconque d'équation $(x-x_0)^2+(y-y_0)^2=K$, et une droite d'équation $y=mx+p$, les points d'intersection sont :

$$x = \frac{-2mp+2x_0+2my_0-\sqrt{[(2mp-2x_0-2my_0)^2-4(1+m^2)(-k+p^2+x_0^2-2py_0+y_0^2)]}}{2(1+m^2)} ;$$

$$y = p + \frac{-m^2p+mx_0+m^2y_0-m\sqrt{[k+km^2-p^2-2mpx_0-m^2x_0^2+2py_0+2mx_0y_0-y_0^2]}}{1+m^2}$$

et $x = \frac{-2mp+2x_0+2my_0+\sqrt{[(2mp-2x_0-2my_0)^2-4(1+m^2)(-k+p^2+x_0^2-2py_0+y_0^2)]}}{2(1+m^2)} ;$

$$y = p + \frac{-m^2p+mx_0+m^2y_0+m\sqrt{[k+km^2-p^2-2mpx_0-m^2x_0^2+2py_0+2mx_0y_0-y_0^2]}}{1+m^2}$$

[[fin de la preuve]]

Poursuivons avec la construction des intersections de deux cercles. Cette construction pose une énigme : elle correspond a priori à la résolution de deux équations du second degré, et pourtant n'introduit qu'une seule extension algébrique.

Sans résoudre les équations, nous pouvons résoudre cette énigme. En effet, cette intersection correspond à la construction des intersections entre l'un des deux cercles et la corde commune de ces deux cercles. Or, nous avons vu au paragraphe précédent que la corde commune de deux cercles est une droite rationnelle. Par conséquent, il est normal de trouver un résultat comparable à celui d'une intersection entre une droite et un cercle.

Proposition I.3-d : Le cas général d'une intersection entre deux cercles, est non-rationnel. Cette construction introduit une seule extension algébrique quadratique.

[[La preuve est laissée au lecteur.]]

Plus étonnant peut-être, la construction d'une bissectrice introduit une extension quadratique. L'étonnement vient peut-être de ce que les droites sont associées un peu trop rapidement à des constructions rationnelles. En fait ce n'est pas toujours le cas, comme nous le verrons aussi à la section suivante. Dans le cas de la bissectrice, l'existence de deux solutions, bissectrices intérieure et extérieure, est caractéristique de l'introduction d'une extension quadratique. Le calcul confirme l'introduction d'une extension quadratique.

Proposition I.3-e : Le cas général de la construction des bissectrices d'un couple de droites est non-rationnel. Cette construction introduit une seule extension algébrique quadratique.

[[La preuve est laissée au lecteur. Par exemple la bissectrice entre l'horizontale et la première bissectrice introduit $\tan(\pi/8) = \sqrt{2} - 1$.]]

Remarque : Quelles sont les racines carrées introduites ? Quel rapport entretiennent-elles avec la figure ? Dans le cas d'une construction de triangles équilatéraux, c'est $\sqrt{3}$ qui apparaît. Dans le cas de la construction d'une bissectrice, c'est un nombre de la forme $\sqrt{1+m^2}$ qui apparaît où m est lié à la pente entre les deux droites. Mais dans le cas général, la racine introduite ne semble pas avoir de forme aussi particulière par rapport à la figure. Par exemple comme nous l'avons vu dans le cas de l'intersection de deux cercles, il n'est pas clair a priori que les deux racines carrées introduites par la résolution successive de deux équations du second degré soient égales à un facteur rationnel près. Pourtant c'est le cas, puisque le domaine de rationalité est le même.

c) Récapitulatif des constructions rationnelles et non-rationnelles.

Construction	Nature (des résultats)	Nature des procédés de constructions.
intersection dte/dte	rationnelle	rationnelle
milieu	rationnelle	règle-compas (usuel) : 1 extension $\sqrt{3}$ règle-compas + effort : rationnelle règle-compas + corde commune : rat. compas : 2 extensions ou moins ?
symétrique	rationnelle	règle-compas (usuel) : 1 extension $\sqrt{3}$ règle-compas + corde commune : rationnel
perpendiculaire	rationnelle	rationnelle
parallèle	rationnelle	rationnelle
médiatrice	rationnelle	rationnelle
point sur droite	rationnelle	rationnelle
centre d'un cercle	rationnelle	compas seul : 2 extensions ou moins ?
inverse d'un point/cercle	rationnelle	règle-compas : 1 extension ou moins ? règle-compas + corde commune : rationnel

Construction	Nature (des résultats)	Nature des procédés de constructions.
"autre" intersection droite/cercle	rationnelle	rationnelle
"autre" intersection cercle/cercle	rationnelle	rationnelle
"autre" bissectrice	rationnelle	rationnelle
point sur cercle	non-rationnelle (usuel) rationnelle (figure I.2-c)	non-rationnelle rationnelle
intersection droite/cercle	non-rationnelle	non-rationnelle
intersection cercle/cercle	non-rationnelle	non-rationnelle
bissectrice	non-rationnelle	non-rationnelle

d) Spécifications géométriques et complexité.

Le cadre général de notre travail étant l'obtention d'une figure géométrique répondant à une spécification géométrique, nous ne pouvons nous restreindre à l'étude des constructions élémentaires. Il nous faut aussi étudier le lien existant entre une spécification géométrique et la complexité algébrique d'une figure satisfaisant cette spécification, si une telle figure existe, ou peut être construite.

Les problèmes historiques tels que : quadrature du cercle, trisection de l'angle, doublement du cube, nous fournissent un premier élément de réflexion. Ils permettent de répondre à la première question "Existe-t-il un lien simple entre une spécification et la constructibilité de la figure spécifiée ?" Ces spécifications appartiennent à la catégorie des spécifications géométriques qui ne sont pas constructibles à la règle et au compas. Ceci nous permet de répondre par la négative à la première question. Une spécification géométrique n'a aucune raison d'être constructible. Et la recherche de la constructibilité d'une spécification est un problème difficile qui nous ramène aux résultats du second chapitre.

Par ailleurs, les nombres réels que font intervenir ces problèmes célèbres sont soit non-constructibles bien qu'algébriques ($\sqrt[3]{2}$ dans le cas de la duplication du cube), soit non-constructibles car transcendants ($\sqrt{\pi}$ dans le cas de la quadrature du cercle). L'extension de la représentation proposée dans la suite de cette thèse à ces nombres algébriques et transcendants a été étudiée, mais n'a pas donné lieu à l'obtention de résultats favorables.

Notons l'apparition dans les logiciels de géométrie (dans Cabri II par exemple) de construction et d'objets comme les ellipses et les longueurs d'arc. Ils introduisent également ces nombres algébriques et transcendants, et les problèmes de représentation qui leur sont attachés.

Essayons de répondre à une seconde question : "Existe-t-il un lien entre une spécification géométrique et la complexité algébrique d'un problème ?". Elle a déjà reçu en partie une réponse précédemment. Plus généralement, la réponse est négative.

Dans les deux cas, le lien existe, mais il n'y a pas de règle simple pour déterminer le degré de complexité algébrique d'une figure à de l'examen d'une spécification de la dite figure.

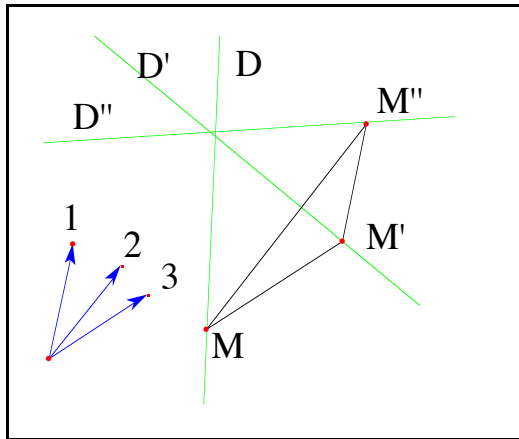


Figure I.3-e: Problème de Steiner.

Énoncé du problème de Steiner :

Construire un triangle $MM'M''$ sur les trois droites D, D', D'' de telle sorte que les cotés de ce triangle soient selon les trois directions 1, 2, 3.

Comme il n'y a pas de règle simple pour dire si telle spécification est constructible à la règle et au compas. Enfin, bien sûr (sinon les deux propositions précédentes seraient caduques), il n'y a pas de règle simple pour trouver une procédure de construction, ou même une figure (sans connaître sa construction) répondant à une spécification donnée.

Pour illustrer les deux points qui précèdent, notons que la règle "une spécification ne faisant intervenir que des équations de droites, de parallélisme, des relations de colinéarité, est rationnelle", est fautive. Tout comme une règle "toute spécification ne faisant intervenir que des équations de cercle est non-rationnelle", est fautive aussi. Un contre-exemple à la première règle est donné par le problème de Steiner figure I.3-e. D'autres spécifications ne faisant intervenir que des droites impliquent des pentagones réguliers.

Conclusion de ce chapitre :

L'étude semi-pratique qui se clôt ici avec le problème de Steiner a montré clairement le manque d'adéquation entre une spécification et la complexité d'une construction vérifiant cette spécification. Autrement dit, il n'y a pas de critères simples et utilisables pour donner une limite à la complexité d'une figure géométrique en fonction de sa spécification.

Par contre, en ce qui concerne les constructions élémentaires et les procédés de construction qui leur sont associés, une hiérarchie s'établit, favorisant les constructions et procédés de constructions rationnels. Cette hiérarchie est un bon point de départ pour la définition d'un ensemble de constructions de base rationnelles ou le plus simples possible.

En ce qui concerne la possibilité de compilation, c'est un sujet délicat qu'il faut développer. Par exemple, nous voyons que nombre de constructions introduisent des objets intermédiaires ayant une complexité algébrique plus grande que les résultats finaux, complexité dont hérite la figure s'il n'y a pas une forme de compilation. Mais ces compilations sont délicates à obtenir.

I-4. Étude pratique de la complexité de quelques situations géométriques courantes, d'un ensemble de 512 théorèmes de géométrie et de situations arbitrairement complexes.

Introduction :

La fin de cette première partie sur la géométrie consiste en une étude pratique de la complexité algébrique de situations géométriques usuelles. Elle repose sur quelques cas particuliers a priori complexes, comme la situation du cercle des neuf points et la solution du problème de Napoléon. Ces situations appartiennent à la géométrie du compas, a priori difficile, alors que la géométrie projective, ou la géométrie de la règle, sont a priori plus faciles. La complexité des figures est jugée en fonction des catégories de nombres réels concernées. Toutes les figures obtenues par une construction possible à la règle (et éventuellement avec quelques autres instruments comme le traceur de perpendiculaire, de parallèle ou de milieu et le traceur de corde commune) sont rationnelles, elles sont donc faciles.

Surtout, cette étude est appliquée à un ensemble de 512 théorèmes de géométrie regroupés par Shang-Ching Chou dans sa thèse [Chou-88]. Ils constituent un ensemble très représentatif des figures usuelles obtenues à la règle et au compas. L'étude de ces 512 situations géométriques proposées par Chou est un élément crucial d'appréciation de la complexité des situations géométriques courantes. L'objectif de l'ensemble de notre travail, la construction numérique de figures géométriques exactes, n'est valide que dans la mesure où cette étude de 512 situations géométriques indique que les figures courantes ne sont pas trop complexes. En effet s'il s'avérait qu'il existe essentiellement des figures géométriques ou des situations de théorèmes de complexité algébrique arbitrairement grande, le coût exponentiel de la gestion de tels cas limiterait considérablement son application.

Enfin, quelques cas de figures exceptionnelles, arbitrairement complexes au sens général, sont étudiés. Nous prenons comme exemple pour ces figures complexes, des spirales constituées de triangles.

a) Étude de quelques situations particulières.

Géométrie projective, affine ou à la règle et avec outils rationnels.

Nous commençons avec des cas de figures appartenant à la géométrie projective. Plus généralement, avec toutes les situations où n'interviennent que des constructions d'intersections

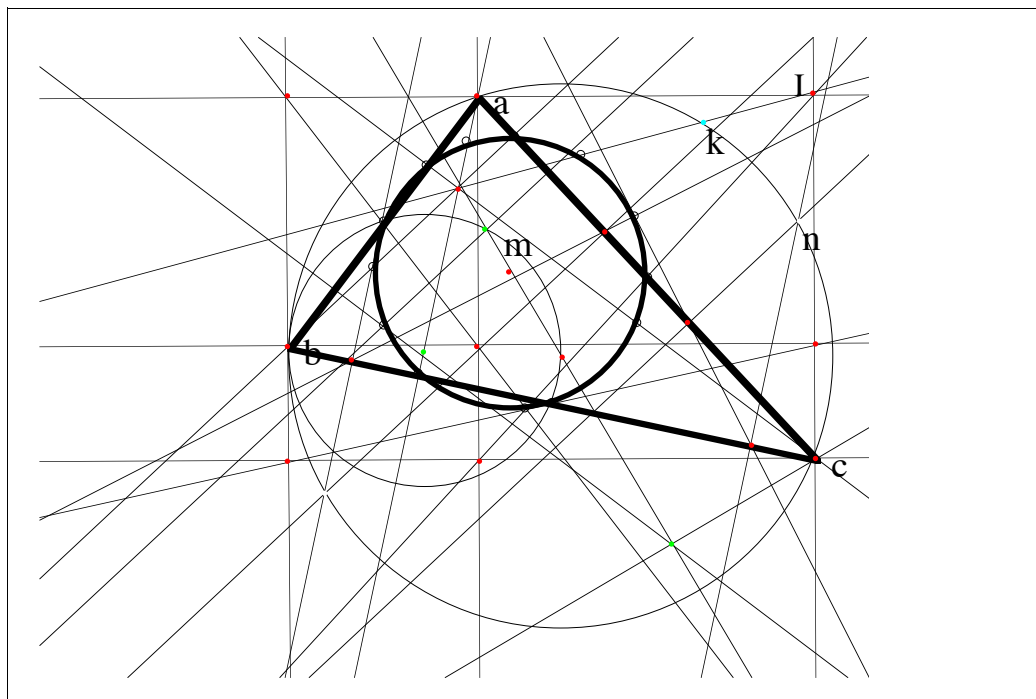


Figure I.4-a : Cercle des neuf points.

de droites, de parallèles et de perpendiculaires, et même de milieux, la complexité de la figure obtenue est nulle car toutes ces constructions sont rationnelles.

En analysant l'ensemble des 512 situations géométriques proposées par Shang-Ching Chou dans sa thèse, un premier résultat nous est fourni : parmi les 512 figures, il y a 202 figures qui n'utilisent ni le compas, ni le bissecteur, et qui sont rationnelles par conséquent.

Autour du cercle des neuf points.

Nous proposons ensuite la situation suivante figure I.4-a, florilège de résultats sur le cercle des neuf points. L'ensemble de ces résultats est tiré de [Chou-88], et les références à des théorèmes dans cette section se rapportent à cet ouvrage. Cette situation comporte les éléments suivants :

- un triangle (abc),
- son orthocentre et
- le cercle des neuf points habituellement associé qui passe par
- les milieux des cotés du triangle,
- les pieds des hauteurs du triangle et
- les points d'Euler¹⁴ du triangle (milieu entre l'orthocentre et un sommet).

De plus, ce cercle comporte la construction :

- du cercle circonscrit au triangle (abc),
- d'un point k sur ce cercle et

¹⁴ Pour la définition des éléments exotiques "points d'Euler", Anticentre, Orthopole, droite de Simson, ... nous renvoyons à [Chou-88].

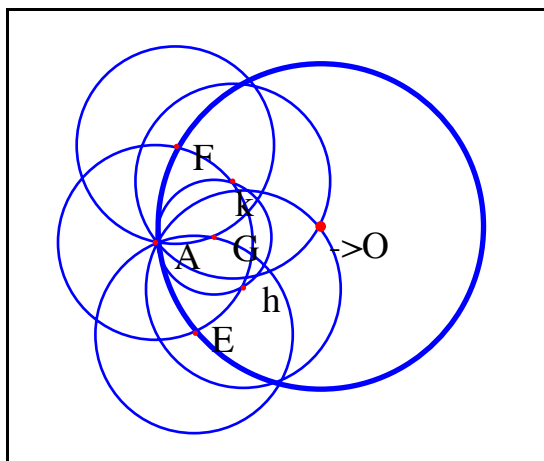


Figure I.4-b: Problème de Napoléon.

Problème de Napoléon :

Étant donné un cercle, comment construire son centre en utilisant le compas seulement ?

Construction proposée : selon [Carrega-89].

Commentaire : deux extensions algébriques quadratiques de plus que celles associées au cercle de départ sont nécessaires, alors que la solution est dans le domaine de rationalité de ce cercle.

- du milieu entre k et l'orthocentre, ce point est aussi sur le cercle des neuf points.

Enfin, cette figure comporte également la construction :

- de l'anticentre d'un quadrilatère cyclique (abck) (théorème 269 identique au théorème 210),
- du point d'intersection, défini dans le théorème 210, entre les diagonales de trois rectangles à cotés parallèles. Les cotés du triangle constituent les autres diagonales de ces rectangles,
- du point d'intersection des droites de Simson correspondant à des points opposés (n et n') sur le cercle circonscrit, cf. théorème 306,
- de l'orthopole pour un triangle d'un diamètre circonscrit (bm), cf. théorème 495.

Remarque : Cette figure ne contient pas la situation du théorème 294 qui est pourtant rationnelle car elle ressemble beaucoup aux théorèmes 210 et 269 déjà présents.

Cette figure ne nécessite que des constructions rationnelles. Elle comporte $9+k+l+3m+3n$ points (k, l, m et n étant le nombre de points de type k, l, m et n cités plus haut). Cette figure, ou une autre figure un peu plus complète mais toujours rationnelle, intervient dans de nombreux théorèmes de géométrie dont notamment, parmi les 512 théorèmes regroupés par Chou dans les théorèmes 32, 196, 197, 198, 199, 200, 203, 207, 208, 210, 211, 214, 216, 238, 239, 242, 243, 244, 246, 269, 278, 294, 306, 391, 392, 400, 449, 451 et 495.

Autour du problème de Napoléon.

Nous terminons cette section avec la situation du problème de Napoléon, et plus généralement avec le cas de la géométrie du compas.

Le problème de Napoléon consiste à trouver une procédure de construction avec le compas seul, du centre d'un cercle donné. La figure I.4-b donne une solution au problème. Elle

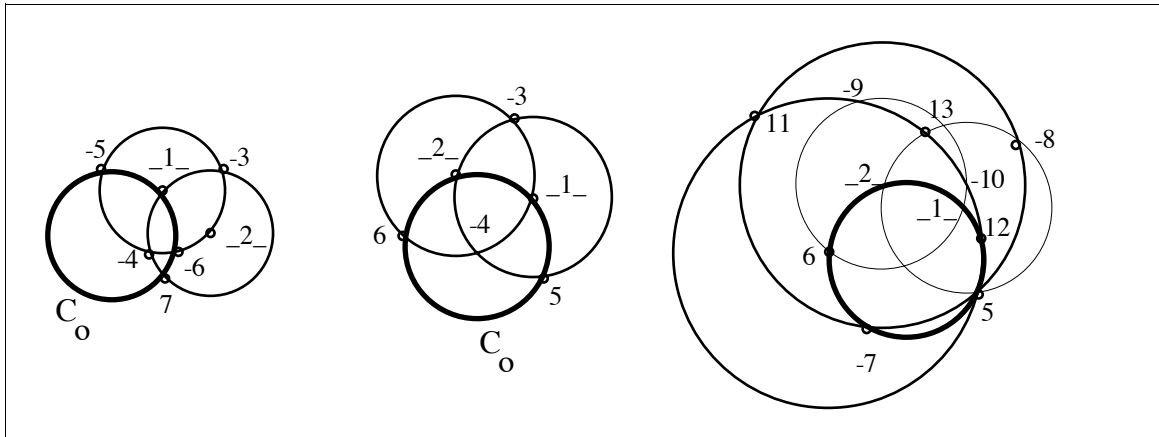


Figure I.4-c: Cercles et Points rationnels au compas.

introduit deux extensions algébriques. Existe-t-il une construction rationnelle ? La question reste entière.

Mais plus généralement, dans le cadre de la géométrie du compas, quels sont les points rationnels qui peuvent être construits au compas ? Le centre du cercle peut-il être construit ? Essayons d'atteindre le centre du cercle par une suite de constructions rationnelles.

La figure I.4-c comporte quelques points rationnels qui peuvent être construits au compas seul. Le cercle de départ C_0 est en gras, quelques points arbitraires rationnels (notés $_n\$) sont pris pour construire de nouveaux points :

- Sur la première figure, le point $_1\$ est un point arbitraire rationnel du cercle C_0 , $_2\$ est un point arbitraire rationnel du plan. La construction des deux cercles ayant ces points pour centre et le segment pour rayon donne 5 nouveaux points. Parmi eux, -3 et -4 sont les sommets de triangles équilatéraux de côté $[_1\, _2\]$, ils ne sont pas rationnels, ils appartiennent à $\mathbb{Q}[\sqrt{3}]$. Les points -5 et -6 sont irrationnels a priori, car ce sont des intersections entre deux cercles quelconques. Par contre, le point 7 est symétrique de $_1\$ par rapport aux droites qui relient le centre du cercle initial C_0 et $_2\$. Par conséquent, 7 est rationnel.

Cette figure permet seulement de construire un point rationnel supplémentaire quelconque sur le cercle, elle n'est pas intéressante.

- Sur la seconde figure, les points $_1\$ et $_2\$ sont des points arbitraires rationnels du cercle C . La construction des deux cercles ayant ces points pour centre et le segment pour rayon donne 4 nouveaux points. Parmi eux, -3 et -4 sont les sommets de triangles équilatéraux de côté $[_1\, _2\]$, ils ne sont pas rationnels, ils appartiennent à $\mathbb{Q}[\sqrt{3}]$. Les points $5, 6$ sont les symétriques respectifs de $_1\$ et $_2\$ par rapport aux droites qui relient respectivement le centre du cercle initial et $_2$ ou $_1$. Ces points sont donc rationnels.

- La troisième figure reprend les notations de la seconde figure. Les cercles de centre un point rationnel de la figure 2 passant par le point 5 sont tous construits (c'est à dire les cercle de centre 2 et 6). Ces cercles interceptent les cercles précédents en $-7, -8, -9, -10$ qui sont des points irrationnels et $11, 12, 13$ qui sont des points rationnels.

- Ainsi de suite à l'étape suivante en prenant l'ensemble des cercles passant par 6, puis par 11, 12, 13 et ... Après plus de 100 constructions, il reste des cercles et des points rationnels nouveaux à construire, mais le centre du cercle initial n'est toujours pas en vue.

b) Étude de 512 théorèmes.

Nous présentons dans cette section, l'analyse de 512 théorèmes de géométrie que Shang-Ching Chou a récoltés pour sa thèse. Nous commençons avec une présentation du travail de Chou. Ensuite nous donnons les résultats bruts de cette étude, c'est à dire la complexité algébrique de ces 512 théorèmes. Enfin une analyse de ces résultats est proposée ainsi que des éléments d'explication pour éclairer ces résultats.

La source : les 512 théorèmes regroupés par Shang-Ching Chou.

Le travail présenté par Shang-Ching Chou dans Mechanical Geometry Theorem Proving comporte en première partie l'exposé de plusieurs méthodes de démonstration automatique basées sur les idées de Wu utilisant les ensembles caractéristiques de Ritt et/ou les bases de Gröbner. En seconde partie, cette thèse expose 512 théorèmes de géométrie prouvés par ces méthodes de démonstration automatique.

Ces théorèmes sont issus pour la plupart de College Geometry de N. Altschiller-Court, ainsi que d'autres ouvrages d'enseignement de la géométrie de culture anglo-saxonne, française et chinoise : Introduction to geometry de Coxeter, Leçons de géométrie élémentaire de Hadamard, A survey of geometry de Howard, Geometry Transformation de Yaglom, et A collection of problems in high school mathematics (en chinois, édité à Pékin). D'autres théorèmes célèbres issus de la culture européenne sont aussi pris en compte : théorèmes de Pappus, de Simson, de Steiner, de Pascal, de Gauss, de Morley, de Napoléon, cercle des neufs points, etc...

Cet ensemble très diversifié de plus de cinq cents théorèmes constitue un ensemble non négligeable de situations géométriques étudiées traditionnellement, et nous permet d'évaluer la complexité arbitraire d'un énoncé courant de géométrie.

L'intérêt présenté par la collection de situations géométriques de Shang-Ching Chou est double. D'une part, son volume est le garant d'une certaine représentativité et donne une légitimité à une étude épistémique sur les situations géométriques. D'autre part, la forme de ces théorèmes rend possible une étude exhaustive des situations sans pour autant avoir à fournir un travail aussi important que celui fourni par Chou et son équipe.

En effet, chaque théorème étudié par Shang-Ching Chou est proposé dans son livre sous la forme (cf. figure I.4-d.) :

- d'un texte anglais décrivant la situation géométrique et la propriété géométrique observée dans cette situation,
- d'une description informatique d'une situation du théorème étudié avec :

Example 107 : (M34-121: 1.0s, 21). Let ABC be a triangle with $AC=AB$. D is a point on BC, Line AD meets the circumcircle of ABC at E. Show that $AB^2=AD.AE$

Points A, B are arbitrarily chosen. Points C, O, E, D are constructed (in order) as follows: $CA = AB$; $OA = OC$; $OA = OB$; $EO = OA$; D is on line AE; D is on line BC. The conclusion : $AB.AB = AD . AE$

$A = (0,0)$, $B = (u_1,0)$, $C = (x_1,u_2)$, $O = (x_3,x_2)$, $E = (x_4,u_3)$, $D = (x_3,x_5)$.

The **nondegenerate** conditions: Points A, B and C are not collinear; Line BC intersects line AE.

Figure I.4-d : Situation géométrique telle que présentée par Shang Ching Chou.

- le choix d'un ensemble de paramètres décrivant la figure, le plus souvent les coordonnées de quelques points intervenant dans la figure, accompagné

- d'une procédure de construction de la figure reposant sur ces paramètres ou ces points "de base". Cette construction est proposée sous la forme d'une suite de pseudo-constructions élémentaires, en fait plus exactement sous la forme d'une suite d'équations algébriques qui fournissent un système d'équations presque triangulaire, permettant le calcul des coordonnées des points intervenant dans la situation.

- Enfin, Chou présente les résultats de la démonstration automatique du théorème, c'est à dire les conditions de non dégénérescence,

- Un schéma de la situation est également proposé.

Complexité algébrique de 512 théorèmes.

L'ensemble de ces 512 théorèmes a été étudié sous l'angle de la complexité algébrique des figures. Auparavant, une première évaluation de la complexité géométrique des figures est donnée. Elle prend en considération le nombre de paramètres et de variables à déterminer pour chaque figure.

n	1 ou 2	3	4	5	6	7	8	9	10	11	12	13	14	15	16 et +
P(n)				16	266	94	80	30	16	5	2		3		
V(n)	9	38	20	55	57	57	51	44	35	28	25	23	23	25	45

Tableau I.4-a : complexité géométrique des 512 situations

Le tableau I.4-a ci-dessus reprend :

- P(n) le nombre de situations parmi les 512 qui comportent n paramètres et
- V(n) le nombre de situations parmi les 512 qui comportent n variables.

En raccourci, une figure ayant $2i$ paramètres et $2j$ variables correspond à une figure ayant $(i+j)$ points dont i points sont donnés (points de base) et j points sont construits.

Passons maintenant à l'étude de la complexité algébrique des 512 situations Géométriques. Quatre caractéristiques représentatives de la complexité algébrique des figures ont été réunies dans le tableau I.4-b ci-dessous. Ces quatre critères d'analyse sont les suivants :

- Nb(n) : le nombre de situations parmi les 512 qui comportent la résolution de n équations du second degré lors de la construction

- Max(n) : le nombre de situations parmi les 512 qui nécessitent n extensions algébriques lors de la construction de la figure en partant de paramètres rationnels arbitraires. Il est entendu que le cas général correspond au nombre maximal d'extensions Nb, et qu'il y a une probabilité d'obtenir une complexité inférieure dans un cas particulier; mais sur l'ensemble des figures, cette probabilité est nulle.

- Loc(n) : le nombre de situations parmi les 512 qui nécessitent n extensions algébriques lors d'une construction modifiée localement.

- Glob(n) : le nombre de situations parmi les 512 qui nécessitent n extensions algébriques lors d'une construction modifiée globalement.

Les deux évaluations citées en dernier (Loc(n) et Glob(n)) prennent en compte des modifications locales ou globales de la procédure de construction de Chou qui diminuent la complexité de la figure initiale. Ces modifications sont définies et étudiées plus précisément en III-2. Elles portent sur la recherche de constructions rationnelles, et sur le (re-)formulation de la situation.

En bref, un exemple de construction rationnelle est donné à la figure I.2.c. Il y a modification locale quand cette procédure de construction est préférée à la prise d'un point arbitraire non rationnel sur un cercle comme figure I.4-d pour les points C et E.

Un exemple de modification globale est donné dans une situation comportant le centre I du cercle inscrit dans un triangle ABC donné. Cette situation peut être re-formulée comme la construction du triangle ABC à partir de A,B et I. La première situation (avec A, B, C données) introduit deux extensions algébriques quadratiques, la seconde (avec A, B, I) est rationnelle.

Voici les résultats regroupés par classe de complexité. La liste des 512*4 complexités pour chaque théorème est donnée en annexe 3. Elle est longue de 3 pages sur 3 colonnes

n :	0	1	2	3	4	5	6	7	8	9	10	11	12
Nb(n)	202	28	80	80	55	35	11	2	8	5	5	0	1
Max(n)	354	96	42	15	2	2	1						
Loc(n)	441	57	11	3									
Glob(n)	492	20											

Tableau I.4-b : Complexité algébrique des 512 situations.

L'obtention de ces résultats est issue d'un double travail :

- le premier "à la main" a eu pour conséquences de séparer les 300 situations de toute évidence rationnelles des 180 autres et successivement, d'écrêter ces 180 problèmes non rationnels pour trouver des figures équivalentes localement ou globalement qui soient rationnelles.

- Enfin, pour la trentaine de problèmes restant, une étude numérique sur ordinateur a été effectuée pour confirmer la complexité trouvée, ou éventuellement découvrir l'apparition d'une rationalité insoupçonnée. Ce dernier travail a été entrepris en collaboration avec Patrick Lixi lors de son stage de Magistère de Mathématiques-2ème année pendant l'été 96, et a donné lieu à l'écriture d'un rapport portant en particulier sur ce sujet, cf.[Lixi-96].

Il va sans dire que les résultats "à la main" sont susceptibles de contenir quelques erreurs dues à la grande quantité de données à étudier. Une double étude à la main de ces 512 théorèmes a, je l'espère, limité le nombre d'erreurs.

Analyse des résultats.

L'analyse de ces résultats est très importante comme il est dit déjà plus haut. Les résultats de cette analyse sont positifs, même excessivement et étonnamment positifs.

Si l'on s'attache à la première évaluation $Nb(n)$ de ce tableau I.4-b, les résultats sont en accord avec ceux du tableau préliminaire I.4-a concernant la simple complexité géométrique des figures en termes de nombre d'éléments. Si l'on met de côté les figures relevant de la géométrie projective $Nb(0)$, le nombre d'extensions suit une courbe qui ressemble à celle empruntée par $V(n)$, correspondant au nombre de constructions. La proportion de constructions quadratiques est à peu près uniforme.

Au vu de $Nb(n)$, nous pourrions nous inquiéter. En effet, dans les conditions données par $Nb(n)$, si une équation du second degré signifie une extension quadratique supplémentaire, une représentation formelle pour les nombres constructibles est fort coûteuse. Son emploi est restreint aux situations comportant moins de 4 extensions algébriques.

Le critère de restriction employé correspond à limiter le coût de la gestion des nombres algébriques à un coût arbitrairement 1000 fois supérieur au simple coût de gestion de nombres rationnels. Comme le chapitre II-2 le montre, le coût relatif d'une multiplication dans une suite de n extensions quadratiques successives est de l'ordre de 5^n fois le coût d'une multiplication entre rationnels. Par suite, comme $5^4=5*5*5*5=25*5*5=125*5*5=625*5=3125$, au delà de 4 extensions quadratiques successives, la gestion des nombres algébriques est trop coûteuse. Parmi les 512 situations proposées, $Nb(4,5,\dots) = 132$ situations dépassent le critère. Soit, si les 202 figures qui relèvent de la géométrie projective ne sont pas prise en compte, près de la moitié des situations qui apparaissent usuellement¹⁵.

Heureusement, l'hypothèse de départ "une équation du second degré = une extension algébrique quadratique" est fautive dans notre situation, comme nous le montre la suite du tableau.

En effet, l'analyse des résultats relatifs à $Max(n)$ permet d'avancer comme un préliminaire que les situations géométriques sont très particulières. La comparaison entre $Nb(n)$ et $Max(n)$ montre qu'un nombre très important d'équations du second degré trouvent leurs solutions parmi les rationnels. Ce comportement va à l'encontre du comportement général. L'emploi des méthodes formelles générales pour la résolution de ces équations risque donc d'introduire des

¹⁵ en considérant les 512 théorèmes comme représentatif.

nombres rationnels sous la forme de racines carrées réductibles. Ce qui nous ramène aux conclusions négatives du paragraphe précédent, ou nous oriente vers la recherche de réductions possibles.

Nous parvenons au résultat important qui suit. Dans l'hypothèse d'une arithmétique exacte évoluée, l'analyse de la ligne Max(n) mène à la conclusion suivante :

Résultat expérimental I.4-a : Avec une arithmétique sur les nombres constructibles réalisant des réductions algébriques afin d'assurer que chaque nombre est représenté sous une forme exacte et minimale, l'essentiel des situations géométriques peut être pris en compte avec un surcoût de 1000.

En effet, avec un maximum autorisé de 1000 pour le surcoût de gestion des nombres constructibles, seulement 5 situations sur 512 sont trop complexes pour être traitées.

Rappelons, pour bien comprendre l'intérêt de ce premier résultat que : l'arithmétique usuelle des ordinateurs est une arithmétique approchée (à virgule flottante). Par conséquent, la majorité des figures géométriques est usuellement erronée. Pour avoir des figures "justes", il faut une arithmétique exacte. Mais une arithmétique exacte est coûteuse, elle n'est envisageable que si la complexité n'est pas trop élevée. Avec ce résultat, l'emploi d'une arithmétique exacte est autorisé car la complexité est limitée, à condition, tout de même, qu'une forme normale puisse être obtenue.

Notons cependant qu'il ne suffit pas d'avoir une arithmétique mixte : en précision infinie sur les nombres rationnels, et en virgule flottante pour les réels. Puisque dans 310 situations sur 512 des nombres non rationnels sont nécessaires.

Notons également que l'emploi d'une arithmétique exacte "classique" (comme celle proposée par Derive, Mathematica ou Maple) risque de ne pas être suffisant. En effet les réductions algébriques nécessaires à ce premier résultat sont souvent optionnelles, voire ignorées. Et sans ces réductions, nous retrouvons la situation du paragraphe précédent. Usuellement, les arithmétiques utilisées pour les nombres algébriques réduisent les racines carrées simples (non imbriquées) et s'arrêtent là, même s'il existe des algorithmes de simplification. Comme les cas qui nous intéressent portent sur des racines imbriquées, ces réductions "élémentaires" ne sont pas suffisantes.

Les résultats concernant les lignes Loc(n) et Glob(n) sont aussi très importants. Nous les regroupons car ils portent sur le travail de recherche d'une figure rationnelle. Ils nous amènent à formuler la proposition suivante :

Résultat expérimental I.4-b : Avec un effort de recherche de procédures de construction géométriques rationnelles, l'essentiel des situations géométriques peut être pris en compte avec une arithmétique exacte sur les rationnels.

En effet, il s'avère au regard de la dernière ligne du tableau que la quasi totalité des situations géométriques courantes est rationnelle. Et la faible minorité qui nécessite des extensions algébriques n'en nécessite qu'une seule et unique, de type quadratique. Bien sûr,

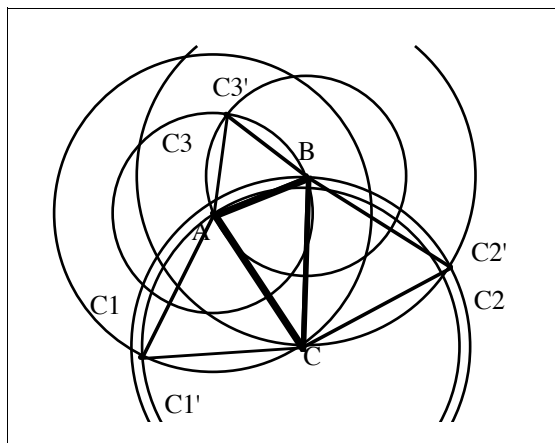


Figure I.4-c: Théorème de Napoléon.

Théorème de Napoléon :

Étant donné un triangle ABC, trois triangles équilatéraux étant érigés sur les cotés extérieurs du triangle ABC, les centres de ces trois triangles équilatéraux forment un triangle équilatéral.

tout cela s'entend à partir du moment où la procédure de construction utilisée pour produire ces situations est bien choisie.

Dans les cas où des modifications locales et globales sont envisagées, une arithmétique exacte évoluée est absolument nécessaire pour avoir des figures géométriques justes. Mais dans le cas des modifications globales, il suffit d'une arithmétique exacte sur les rationnels effectuant des réductions élémentaires pour 492 cas sur 512. Pour les 20 situations restantes, les réductions élémentaires ne suffisent toujours pas, mais des réductions limitées et une représentation naïve des nombres constructibles comme dans [Guillot, Jolivet, Leperd, Moulin-91] conviennent.

L'analyse des conséquences de ces résultats nous amène à entreprendre la recherche d'une représentation exacte sophistiquée des nombres constructibles, car ces résultats ne seront complètement valides que sous les deux conditions que :

- 1) la complexité minimale algébrique puisse être atteinte lors de l'emploi d'une procédure de construction algébrique et géométrique "simple",
- 2) les représentations des nombres constructibles soient aussi performantes avec les réductions promises et restent compatibles avec les méthodes de travail usuelles.

Éléments d'explication

Maintenant que nous avons retiré quelques profits de cette étude, nous pouvons essayer de donner une explication à ces résultats. Nous pouvons trouver un premier élément de réponse dans l'analyse du théorème de Napoléon.

Le cas du théorème de Napoléon est représentatif de ce qui se passe dans beaucoup de théorèmes. Un dessin de la situation concernée est donné par la figure I.4-c. Il indique la présence de 6 cercles, et 3 intersections entre ces cercles. Dans l'hypothèse défavorable citée plus haut, cette situation doit comporter 6 extensions algébriques introduites par les intersections entre les cercles C_1 & C_1' , C_2 & C_2' , C_3 & C_3' . En fait, les cercles ne sont pas indépendants entre eux, et les extensions algébriques s'effondrent les unes sur les autres. Chaque intersection fait intervenir la même extension algébrique comme je l'ai déjà annoncé au

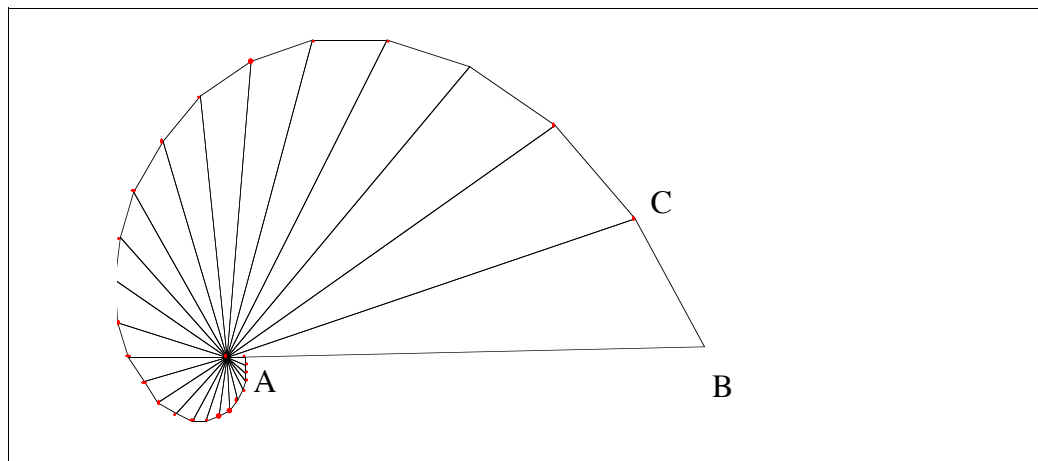


Figure I.4-e : Spirale à partir de triangles similaires.

chapitre I.2, et comme le calcul de simplification permet de le vérifier. Il n'y a là qu'une seule extension quadratique qui est $Q[\sqrt{3}]$ (celle qui apparaît avec les triangles équilatéraux)

Un second élément de réponse peut venir des remarques sur le non-déterminisme -au sens de Prolog- des figures non-rationnelles. Un théorème géométrique est d'autant plus délicat à prouver qu'il comporte des cas de figures distincts. Les situations où cela arrive sont toujours non rationnelles. Il se peut donc qu'il y ait eu une sélection "naturelle" au cours des siècles des situations géométriques non ambiguës. Le cas du théorème de Napoléon est exemplaire, en fait 8 situations analogues à ce théorème existent selon que les trois triangles sont construits à l'intérieur ou à l'extérieur. Cette sélection correspondrait à des situations qui peuvent être rendues rationnelles. Finalement, sur les 512 situations, seulement 20 ont plusieurs cas de figures. Les autres sont parfaitement déterministes.

c) Quelques cas de figures complexes, et arbitrairement complexes d'un point de vue algébrique, Spirales.

Bien sûr, l'étude précédente des 512 théorèmes de géométrie révélant que ceux-ci sont à 96 % rationnels et que les 4 % restant ne nécessitent qu'une seule extension, ne signifie pas que la géométrie dans sa totalité se satisfasse d'une seule extension algébrique quadratique. Pour rappel, les polygones réguliers à 5, 12, 17 cotés constituent quelques figures nécessitant plus d'une extension quadratique, cf. chapitre I.2-c. Cette section est dédiée à des figures arbitrairement complexes du point de vue géométrique, c'est à dire comportant un nombre arbitraire de points tous distincts. Elle contient également quelques figures qui sont arbitrairement complexes du point de vue algébrique .

Comme paradigme de figures arbitrairement complexe, nous avons choisi les spirales. Toutes les spirales données ici sont constituées de triangles. Excepté la première spirale, toutes sont constituées de triangles rectangles et la construction du $n+1^{\text{ième}}$ triangle de la spirale s'effectue sur l'hypoténuse du $n^{\text{ième}}$ triangle. Éventuellement, la construction du $n+1^{\text{ième}}$

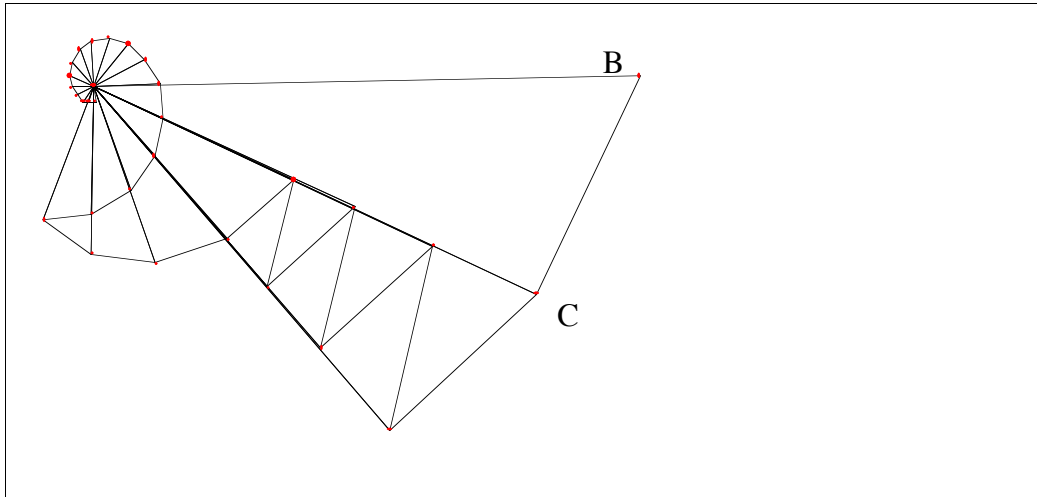


Figure I.4-f : Spirale à développement anarchique.

triangle à partir du précédent est donnée en vis à vis de la spirale elle-même. Le centre de ces spirales est noté A, le premier coté est noté AB, le premier triangle est noté ABC.

Par souci esthétique, toutes ces spirales sauf la seconde pour l'exemple, tournent dans le même sens. Une construction aléatoire d'un coté ou de l'autre de l'hypoténuse, si elle modifie l'aspect esthétique, ne change en rien la complexité algébrique de la figure. En fait, à chaque spirale de n triangles correspondent $2^n - 1$ autres spirales au développement anarchique ayant la même complexité algébrique.

Spirales Rationnelles

Le premier exemple, figure I.4-e, de figure géométrique arbitrairement complexe est donné par une spirale de points aux coordonnées rationnelles. Cet exemple de spirale rationnelle est obtenu par construction de triangles similaires à un premier triangle rationnel ABC. Notons de plus que l'éloignement au centre est une distance non rationnelle mais qui appartient toujours à la même et unique extension quadratique.

Un second exemple, figure I.4-f, identique au précédent, est donné avec un triangle initial rectangle en B. C'est un exemple de développement anarchique, pseudo-aléatoire dans le sens des positifs ou négatifs à chaque nouveau triangle. La complexité de cette figure est la même que pour la précédente.

Les spirales partant d'un triangle quelconque peuvent être "convergentes" (dessin de la figure I.4-e) ou "divergentes" (dessin de la figure I.4-f) :

Proposition I.4-a : Dans le cas de triangle quelconque, la divergence ou la convergence de la construction d'une spirale, faite de triangles similaires accolés, tient dans le rapport $[AC]/[AB]$ de deux cotés accolés consécutifs.

Par conséquent, les spirales partant d'un triangle rectangle sont toujours strictement divergentes (si $B \neq C$ bien sûr) car l'hypoténuse $[AC]$ est strictement plus grande que le coté

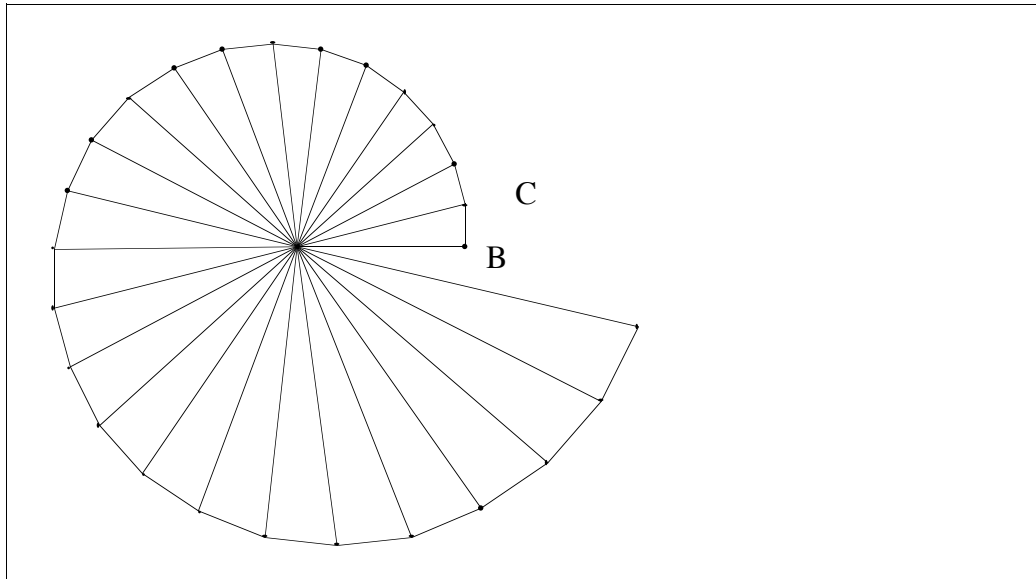


Figure I.4-g : Spirale sur des triangles similaires rectangles.

initial (la base de la figure) [ab]. C'est que nous observerons avec les spirales suivantes, toutes obtenues à partir de triangles rectangles construits sur l'hypoténuse.

Spirale rationnelle sur triangles rectangles similaires : Dans le cas d'une spirale, figure I.4-g, obtenue à partir de triangles similaires, les coordonnées des points de la spirale sont données par les formules ci dessous :

Si le triangle de départ est

A:(0,0), B:(p,0), C:(p,q),

par récurrence, le point de la spirale étant noté $M_n:(X_n, Y_n)$,

alors $M_{n+1}:$ $(X_n - q/p Y_n, Y_n + q/p X_n)$,

et les longueurs $[AM_{n+1}]$ valent $1/p \cdot \sqrt{p^2 + q^2} [AM_n]$.

Ces longueurs appartiennent donc toutes à une même extension algébrique de Q par $\sqrt{p^2 + q^2}$, et une longueur sur deux est rationnelle.

Spirales irrationnelles

Spirale nécessitant une infinité d'extensions par des racines simples (figure I.4-h): Elle est obtenue par ajout d'une longueur constante à la perpendiculaire du point courant. En particulier, la spirale obtenue à partir d'un segment [AB] de longueur égale à la distance constante ajoutée, introduit les racines carrées simples $\sqrt{2}$, $\sqrt{3}$, $\sqrt{4}$, $\sqrt{5}$.

Cependant, cette figure contient des objets dont les coordonnées sont de complexité algébrique arbitrairement grande, car $\sqrt{2} + \sqrt{3} + \sqrt{5} + \sqrt{7} + \sqrt{11} + \dots + \sqrt{P_n}$ est d'ordre 2^n (P_n est le $n^{\text{ième}}$ nombre premier) . Une preuve particulière de ce théorème est donnée en annexe 4. Elle reprend le schéma de démonstration du théorème central de cette thèse qui se trouve chapitre II.2 (la preuve est laissée au lecteur. Une idée pour une démonstration classique est donnée par le calcul des puissances $n^{\text{ième}}$ de $\sqrt{2} + \sqrt{3} + \sqrt{5} + \sqrt{7} + \sqrt{11} + \dots + \sqrt{P_n}$: ce sont des polynômes avec

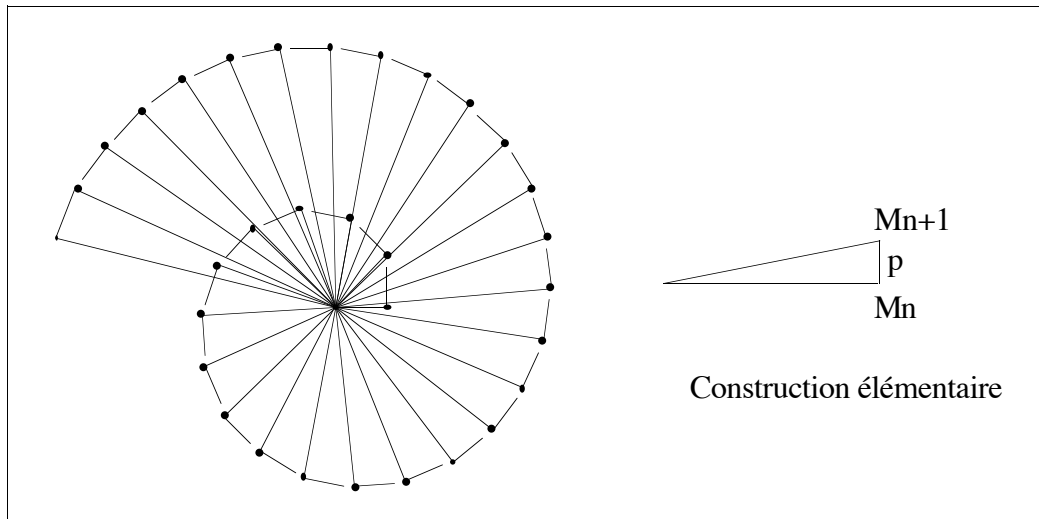


Figure I.4-h : Construction de $\sqrt{1}$, $\sqrt{2}$, $\sqrt{3}$, ..., $\sqrt{32}$.

des racines carrées de produits arbitraires des n premier nombres premiers, et par conséquent $\sqrt{P_{n+1}}$ n'y apparaît pas.)

Dans le cas de la spirale obtenue à partir d'un segment initial de longueur égal à la longueur ajoutée, figure I.4-h, les coordonnées des points de la spirale sont données par les formules suivantes :

Si le triangle de départ est

A:(0,0), B:(p,0),

par récurrence, le point de la spirale étant noté $M_n:(X_n, Y_n)$,

alors le point construit M_{n+1} a pour coordonnées :

$$(X_n - Y_n \cdot \sqrt{X_n^2 + Y_n^2}, Y_n + X_n \cdot \sqrt{X_n^2 + Y_n^2}),$$

et les longueurs $[AM_{n+1}]$ valent $\sqrt{([AM_n]^2 + 1)}$.

Ces longueurs sont donc $\sqrt{2}$, $\sqrt{3}$, $\sqrt{4}$, $\sqrt{5}$, $\sqrt{6}$, ... qui décrivent un corps $Q[\sqrt{2}, \sqrt{3}, \dots]$ de dimension algébrique infinie comme espace vectoriel sur Q .

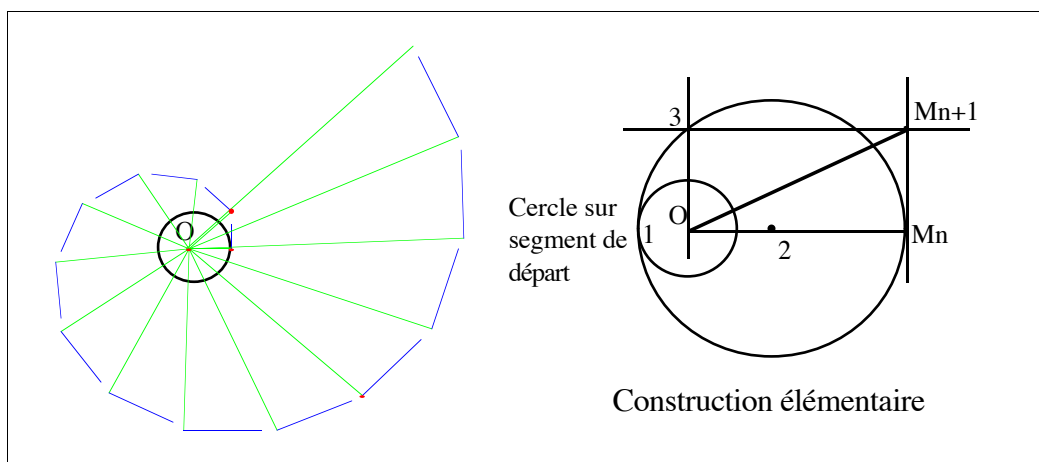


Figure I.4-i : Construction d'une spirale géométrique avec racines carrées imbriquées.

Conclusion de la première partie :

La géométrie à la règle et au compas utilisée en classe est riche d'une longue histoire. Elle comporte encore d'intéressants problèmes. Elle révèle des orientations de recherches nouvelles en informatique (par exemple, chercher une construction rationnelle) qui pourraient être très utiles en démonstration automatique et impose de trouver la solution à de nouveaux problèmes : comment "bien" représenter formellement les coordonnées des figures géométriques obtenues à la règle et au compas ? La solution à ce problème semble abordable sur le plan de la complexité mais nécessite encore une réalisation effective. C'est le sens de la seconde partie.

II. Le paradigme CLP(Géométrie).

Introduction de la seconde partie :

Cette seconde partie porte sur la définition d'un environnement de programmation logique avec contraintes (PLC, CLP pour Constraint Logic Programming en anglais) dédié à la géométrie de la règle et du compas. Par commodité, nous appellerons cet environnement CLP(Géométrie).

Cette partie est introduite par un court rappel de la notion de programmation logique avec contraintes (chapitre II-1), et se poursuit avec la définition d'un paradigme CLP(Géométrie) (chapitre II-2 : domaine, et chapitre II-3 : algorithme de satisfaction des contraintes qui reprend en partie [Bouhineau-95b] et [Bouhineau-96]). Quelques résultats d'utilisation de ce paradigme sont donnés à la fin de cette partie (chapitre II-4). Ce dernier chapitre comporte des "résultats" et exemples d'utilisation relativement immédiats du paradigme CLP(Géométrie) introduit dans cette partie. La troisième et dernière partie de ce manuscrit porte sur des applications conséquentes construites sur ce paradigme CLP(Géométrie).

II-1. CLP - définition d'un paradigme de programmation.

Introduction :

La programmation logique avec contraintes est un paradigme de programmation de haut niveau, récent, dont la bonne santé est démontrée par la richesse et la variété des sujets de recherche qui lui sont liés :

- hiérarchie de contraintes (HCLP),
- environnement et interface visuels avec contraintes,
- classe des problèmes réellement difficiles (NP-difficiles),
- coopération entre solveurs,

et naturellement :

- définition,
- architecture,
- sémantique de nouveaux langages appartenant au paradigme CLP.

L'organisation de conférences régulières et nombreuses sur le domaine, et l'importance de la place prise par ce paradigme dans les conférences de Logique, d'Intelligence Artificielle, de Programmation Logique, de méthodologie de programmation, et de langage de programmation en témoignent aussi.

L'origine de ce paradigme de programmation est à retrouver dans la série de communication : [Cohen-90], [Colmerauer-90], [Older et Vellino-90] parus au même moment et [Jaffar et Lassez-87] et [Dincbas, Van Hentenryck, Simonis, Aggoun, Graf, et Berthier-88] parus quelques années plus tôt.

Nous retraçons rapidement l'histoire de ce paradigme au cours de la première section (II.1-a) de ce chapitre. Puis, en II.1-b, le paradigme CLP est explicité. Enfin, quelques exemples de langages utilisant ce paradigme sont donnés en dernière section (II.1-c).

a) Contraintes et Programmation Logique : *petite histoire des contributions à la Programmation Logique avec Contraintes.*

L'histoire de la programmation logique avec contraintes est bien sûr beaucoup plus courte, et peut-être moins instructive que celle de la géométrie. Elle prend ses sources au confluent de deux paradigmes de programmation des années 1970 : la programmation logique et la programmation par contraintes.

Programmation Logique (Prolog).

À l'origine de la programmation logique et des langages et environnements de programmation de la famille de Prolog, se trouve l'analyse des langues naturelles. Mais ce domaine assez limité n'est pas le seul lieu d'expérimentation des langages de programmation proposés. Une utilisation des environnements de programmation obtenus au cours des travaux sur les langues naturelles est envisagée dans de nombreux autres domaines. Cette utilisation montre l'adéquation de ces environnements à une classe de problèmes beaucoup plus large que l'analyse des langues naturelles. En particulier, elle montre leur adéquation à toute une classe de problèmes que nous nommerons "combinatoires". En même temps, l'emploi de Prolog reste limité à des applications ayant une matière très "formelle", faisant intervenir peu de calculs numériques.

Sur le plan théorique, la programmation logique puise ses fondements dans la logique, en particulier dans la théorie des clauses de Horn (clauses où tous les littéraux sont négatifs, sauf un et un seul, qui est positif) et des algorithmes de satisfaction d'ensembles de clauses qui leur sont appliqués. En 1965, c'est l'algorithme de résolution de Robinson qui initie le mouvement. Il permet de décider de la satisfiabilité d'un ensemble de clauses de Horn que représente aujourd'hui un programme Prolog.

La première réalisation d'une implémentation d'un langage Prolog est due à Alain Colmerauer et Philippe Roussel (en 72) à Marseille (Groupe d'Intelligence Artificielle). Peu après, Robert Kowalsky démontre la complétude de la stratégie Prolog (unification entre les têtes de règle) pour les clauses de Horn, et donne une sémantique des langages Prolog en termes de point fixe. Sur le plan technique, en 1975, des améliorations sont apportées par David Warren en vue de la définition d'une machine Prolog efficace, et d'un compilateur associé. Des modèles de machines Prolog parallèles voient le jour avec le projet "5ième génération d'ordinateurs" développé au Japon dans les années quatre-vingts.

D'une certaine manière, l'approche Prolog vis à vis de la logique est exemplaire. En quelques mots seulement, cette approche consiste à aborder avec modestie ou réalisme des problèmes difficiles. Dans le cas de Prolog, elle consiste à avouer son incapacité à prendre la logique dans sa globalité comme langage d'expression; par défaut Prolog considère un sous-ensemble suffisamment expressif de la logique, les clauses de Horn, sur lequel peuvent s'effectuer des calculs efficaces.

Cette approche "modeste" se retrouve plus tard dans les choix des premiers langages CLP comme Prolog III, et dans notre approche. Dans le cas de Prolog III, elle consiste à restreindre les contraintes acceptables aux contraintes "linéaires", dans notre cas, à favoriser les constructions rationnelles en géométrie. (Nous nommons "linéaires" les expressions et équations où n'interviennent que des formes polynomiales avec des monômes de degré total 1. Par exemple, $3x+y-5$ est linéaire, et m^2x+px ne l'est pas. La notion de linéaire se révèle être une notion clé de la programmation, elle est abordée plus systématiquement à la section II.1-b).

A l'opposé de la programmation par contraintes que nous abordons à la section suivante, Prolog a initialement un domaine principal ou privilégié, celui des arbres, et un seul mécanisme

```

/* retourner(tuple,tuple) vrai si %1, %2 sont en miroir */
retourner(l,l') ->
  palindrome(l,l');

/palindrome(tuple) vrai si %1 est un palindrome de longueur paire */
palindrome(<>) -> ;
palindrome(<e>.l.<e>) -> ;

requête :
  retourner(<1,2,3,4>,l);
réponse : {l=<4,3,2,1>}

requête :
  retourner(k,<1,2,3,4>);
réponse : {k=<4,3,2,1>}

```

Programme II.1.a : Exemple de Prolog pur (avec tuple).

de résolution, l' "unification". Pour l'essentiel des moteurs Prolog, celle-ci est associée à une stratégie particulière : l'exploration "en profondeur d'abord" de l'arbre de recherche, avec retours en arrière. Le schéma d'une machine Prolog est très simple. Un programme Prolog est constitué de deux éléments,

-1- un ensemble de buts (des arbres, ou termes Prolog) $b_1(e_{11}, \dots, e_{1n}) \dots b_p(e_{p1}, \dots, e_{pq})$ appelé requête (ou résolvente), et

-2- un ensemble de règles de la forme $P(a_1, \dots, a_n) \leftarrow P_1(a_{11}, \dots, a_{1k}) \dots P_p(a_{p1}, \dots, a_{pk})$, que nous appellerons l'environnement.

L'exécution du programme s'effectue de la manière suivante : soit $b_1(e_{11}, \dots, e_{1n}) \dots b_p(e_{p1}, \dots, e_{pq})$ la requête. La première étape consiste à sélectionner une règle parmi l'ensemble des règles constituant l'environnement, où le but b_1 apparaît en partie gauche, par exemple $b_1(a_1, \dots, a_n) \leftarrow P_1(a_{11}, \dots, a_{1k}) \dots P_p(a_{p1}, \dots, a_{pk})$. Puis le moteur Prolog effectue l'unification de $b_1(e_{11}, \dots, e_{1n})$ avec $b_1(a_1, \dots, a_n)$, c'est à dire en particulier l'unification des a_i avec les e_i . La requête, ou résolvente, est remplacée par $P_1(a_{11}, \dots, a_{1k}) \dots P_p(a_{p1}, \dots, a_{pk}) b_2(e_{21}, \dots, e_{2q}) \dots b_p(e_{p1}, \dots, e_{pq})$. Le moteur recommence comme précédemment, jusqu'à ce que la requête soit un ensemble vide de buts à satisfaire.

La machine Prolog peut être ainsi vue comme un processus de réécriture, ou comme un algorithme de recherche de satisfiabilité d'un ensemble de clauses d'un type particulier (d'où la notion de résolvente). Du point de vue logique, un moteur Prolog recherche la satisfiabilité de l'ensemble suivant : un ensemble de clauses de Horn auquel est adjointe une clause positive. L'ensemble de clauses de Horn est l'environnement précédent, la clause positive est la requête. Cette recherche de satisfiabilité s'effectue selon l'algorithme de résolution de Robinson. Le tout s'effectue pour Prolog (pur) en logique du premier ordre.

Un exemple de programme Prolog est donné figure II.1.a. Il consiste à retourner un tuple. Il utilise un prédicat "palindrome" qui vérifie qu'un tuple est un palindrome.

Programmation par contraintes.

À l'origine de la programmation par contraintes se trouvent en premier lieu les travaux de Alan Borning pour la réalisation de ThingLab. En 1980, ThingLab est un laboratoire de simulation d'objets physiques et graphiques basé sur une définition des problèmes à modéliser à base d'"objets" (au sens informatique) et de contraintes mathématiques. Le cœur de ThingLab est constitué par un algorithme de satisfaction de contraintes (domaine de recherche relatif aux CSP, pour Constraints Satisfaction Problem dans la littérature anglo-saxonne). Des exemples sont donnés dans [Borning-81] ou dans [Fron-94].

Le travail de Borning est précurseur dans de nombreux domaines, la programmation par contraintes (qui nous intéresse ici), la programmation par objets, et les recherches sur les interfaces graphiques. Les recherches dans ce dernier domaine se poursuivent en utilisant toujours des préceptes issus de la programmation par contraintes, de plus en plus la programmation logique, et naturellement CLP. Notons au passage qu'il y a dans ce domaine peu d'utilisation de la géométrie, au sens où nous l'entendons (règle-compas). Ce constat se retrouve dans le cas de la conception assistée par ordinateur (architecture, mécanique).

Chaque langage ou environnement de programmation à base de contraintes est ainsi défini par trois éléments :

- 1- un domaine pour les variables (par exemple des ensembles finis, très souvent des valeurs numériques, ou des intervalles de réels, parfois des valeurs rationnelles, entières ou booléennes, et des arbres, listes, suites éventuellement),
- 2- un ensemble de contraintes possibles portant sur ces variables (par exemple contraintes linéaires ou algébriques, ou quelconques pour les variables numériques, contraintes sur les ensembles, précédence, etc) et
- 3- un algorithme de résolution de ces contraintes.

Cet algorithme de résolution de contraintes est le plus souvent une boîte noire (à l'opposé de la définition claire du domaine des variables et des contraintes autorisées) dont la complétude et dans une moindre mesure, la correction, sont parfois délicates à définir. Parmi les algorithmes classiques utilisés, nous citerons rapidement des algorithmes d'énumération, des algorithmes de recherche de cohérence locale, de relaxation de contraintes, et de propagation des solutions locales. L'ensemble des possibles est grand, comparé à l'algorithme de résolution presque unique de Prolog.

Prémisses de la programmation logique avec contraintes.

Précurseur de la fusion entre les langages Prolog et les langages à base de contraintes, se trouve Prolog II. Par deux mécanismes, Prolog II introduit des contraintes dans l'algorithme de résolution de Robinson, utilisé dans les machines Prolog :

- D'une part, au cours de l'unification, il doit se dérouler un test coûteux, dit "test d'occurrence" (ou "occur-check") pour vérifier que l'unification n'est pas en train de "boucler" sur une structure infinie (à cause de l'unification). Sur le plan théorique, ce test est inévitable. Selon les implantations des machines Prolog, ce test est ou non effectué du fait de son coût et de son importance moindre sur le plan pratique (dans la pratique, sauf erreur de

programmation, ce test est inutile). Pour concilier les aspects pratiques et théoriques, la machine Prolog II remplace l'unification sur les arbres finis par une unification sur les arbres infinis rationnels. Dans ce cadre où le domaine des termes Prolog est plus grand, le test d'occurrence est inutile. Mais le remplacement de ce test par une unification sur les arbres infinis impose que le mécanisme d'unification soit élargi à un mécanisme de résolution de système de contraintes sur les arbres infinis.

- D'autre part, un prédicat prédéfini fut ajouté à Prolog II. Ce prédicat est le \neq ($\text{dif}(\text{terme}, \text{terme})$ réalisé si %1 est différent de %2). Il est traité comme une contrainte de différence au sens strict, c'est à dire qu'elle peut s'exercer sur des variables. Une contrainte \neq imposée sur deux variables limite par la suite les unifications possibles portant sur ces deux variables. Ce prédicat permet d'améliorer également la rapidité d'exécution de certains programmes Prolog. Au passage, nous noterons que l'importance théorique du \neq n'est pas négligeable; sur des ensembles finis de variables, il permet de définir n'importe quelle contrainte (même l'égalité).

D'une certaine manière, programmation par contraintes et programmation logique sont complémentaires et poursuivent un même but : programmer avec une approche essentiellement déclarative, c'est à dire établir des relations entre des objets. Par ailleurs, ces approches sont complémentaires. L'algorithme d'unification est une forme primaire d'algorithme de satisfaction de contraintes. La plus grande portée des algorithmes de satisfaction de contraintes prolonge l'unification des clauses pour les domaines d'utilisation des contraintes. De même, le processus de résolution des machines Prolog, généralise l'équivalent primaire chez les CSP, et fournit un langage de programmation de haut niveau. Depuis le début, Prolog est reconnu comme très agréable et permettant un prototypage aisé (ce qui est un indice d'ergonomie pour un langage de programmation).

Observons donc comment cette fusion s'opère entre programmation logique et programmation par contraintes.

b) Le paradigme de programmation : CLP.

Histoire contemporaine.

La programmation logique avec contraintes est née après Prolog II dans la fin des années 80. À l'origine se trouvent plusieurs définitions générales d'un nouveau cadre de programmation : la programmation logique avec contraintes, et des langages réalisant des projets théoriques très similaires. Pour citer les trois principaux¹⁶ : CLP(R) proposé par Jaxon Joffar et Jean-Louis Lassez [Jaffar, Michaylov, Stuckey, et Yap-92], Prolog III proposé par Alain Colmerauer [Colmerauer-90] et CHIP proposé par Mehmet Dincbas et son équipe [Dincbas, Van Hentenryck, Simonis, Aggoun, Graf, et Berthier-88]. Ces trois projets ont des particularités fortes qui les distinguent sans équivoque les uns des autres :

¹⁶ Dans une moindre mesure, on peut aussi citer CAL [Sakai et Aiba] et BNR-Prolog [Older et Vellino-90], [Older et Benhamou-93b].

-CLP(R) est principalement orienté vers les contraintes numériques, en ayant une approche aussi large que possible dans ce domaine,

- Prolog III a une approche beaucoup plus générale, avec des contraintes sur les listes et tuples, les nombres rationnels, les expressions et équations linéaires et enfin les booléens, et

- CHIP est principalement orienté vers les contraintes sur les domaines finis.

Les cadres de ces projets sont relativement similaires, ce sont tous des extensions naturelles de Prolog, et l'évolution a montré que les projets théoriques coïncidaient effectivement.

Depuis, en dix ans, ces langages/machines ont remplacé la machine Prolog classique, et ont donné lieu à d'autres réalisations. La communauté scientifique qui travaillait sur Prolog s'est en grande partie déplacée vers ce nouveau domaine de recherche que sont les CLP et leurs extensions.

Programmation CLP.

Sur le plan théorique, la programmation logique avec contraintes emprunte à la programmation logique le mécanisme de résolution par buts, et au domaine de la satisfaction de contraintes les algorithmes de satisfaction de contraintes. Le cœur du moteur d'inférence est constitué par un algorithme de satisfaction de contraintes plus puissant que l'algorithme d'unification qu'il remplace.

Un programme CLP se présente toujours sous la forme de paquets de règles (l'environnement) et d'une requête, ou résolvante. L'exécution du programme s'effectue en trois temps par réduction de la résolvante et accumulation de contraintes selon le schéma de résolution suivant : soient $b_1(e_{11}, \dots, e_{1n}) \dots b_p(e_{p1}, \dots, e_{pq})$ la requête, et S un ensemble de contraintes initiales (S est vide si l'utilisateur n'a rien fixé),

-1- le moteur CLP sélectionne une règle parmi l'ensemble de règles constituant l'environnement où le but b_1 apparaît en partie gauche, par exemple $b_1(a_1, \dots, a_n) \leftarrow P_1(a_{11}, \dots, a_{1k}) \dots P_p(a_{p1}, \dots, a_{pk})$. Puis il ajoute la contrainte $\{b_1(e_1, \dots, e_n) = b_1(a_1, \dots, a_n)\}$ à l'ensemble de contraintes déjà présentes S ,

-2- si le nouveau système est satisfiable, la nouvelle requête à réduire devient (quitte à renommer les paramètres si nécessaire) $P_1(a_{11}, \dots, a_{1k}) \dots P_p(a_{p1}, \dots, a_{pk}) b_2(e_{21}, \dots, e_{2q}) \dots b_p(e_{p1}, \dots, e_{pq})$ et

-3- le moteur recommence comme précédemment, jusqu'à ce que la résolvante soit un ensemble vide de buts à satisfaire. Alors S est le résultat du programme.

Comme pour le moteur Prolog classique, le moteur CLP effectue le choix des règles selon une stratégie particulière, qui assure l'évaluation de toutes les solutions possibles. En cas d'erreur, le dernier choix de règle est remis en cause, et une autre solution est recherchée. Le plus souvent, cette stratégie correspond à l'exploration "en profondeur d'abord" de l'arbre de recherche, avec retours en arrière.

Sur le plan algorithmique, la programmation CLP repose sur quatre principes :

- raisonnement par cas (élargi, car c'est plus qu'une simple étude de cas, grâce à l'unification, et à la résolution de systèmes de contraintes),

- réversivité (très naturelle, essentiellement structurelle, il faut se rappeler que le domaine principal de Prolog est constitué des arbres),

- non-déterminisme (au sens de Prolog), c'est à dire recherche combinatoire exhaustive des solutions; ce non-déterminisme permet d'effectuer des recherches combinatoires très facilement, enfin

- réversibilité : c'est peut-être le plus important, et le plus difficile à manipuler, c'est la déclarativité de Prolog qui s'exprime ici, et en même temps un indice de la bonne définition d'un prédicat.

Contraintes numériques.

En ce qui concerne les contraintes numériques, les inconvénients de Prolog sont dépassés. Plusieurs solutions ont été apportées. La classification suivante des contraintes numériques possibles est importante :

- contraintes linéaires : expressions où n'interviennent que des formes polynomiales avec des monômes de degré total maximum 1. Exemple : $3*x+y$.

- contraintes pseudo-linéaires : expressions où n'interviennent que des formes polynomiales avec des degrés partiels maximum 1. Exemple : $y+m*x$.

- contraintes quadratiques : expressions où n'interviennent que des formes polynomiales avec des monômes de degré total maximum 2. Exemple : $(x-y)^2$. Parmi les contraintes quadratiques se trouvent les équations du second degré en une seule variable. Exemple : $3.x^2+4x-2$.

- contraintes pseudo-quadratiques : expressions où n'interviennent que des formes polynomiales avec des degrés partiels maximum 2.

- contraintes non linéaires ou algébriques, c'est à dire contraintes arithmétiques générales. Par exemple x^3*y^2 .

Les contraintes linéaires appartiennent depuis longtemps au domaine de la recherche opérationnelle. Elles sont associées à des algorithmes efficaces, complets, corrects, et répandus : élimination de Gauss, et algorithme du simplexe. Mais l'emploi de ces contraintes est limité. A l'aide de gels intelligents, le domaine des contraintes pseudo-linéaires est aussi abordé avec efficacité par des algorithmes corrects mais incomplets (ex : Prolog III). Ensuite vient, presque directement, le domaine des contraintes non linéaires ou algébriques. Pour ce domaine, il existe deux types d'algorithmes : les algorithmes formels, type base de Gröbner ou Collins, qui sont coûteux, et les algorithmes approchés à base d'intervalles. Le domaine des contraintes quadratiques, ou pseudo-quadratiques (celui de la géométrie) est peu abordé, il ne possède pas d'algorithmes spécifiques satisfaisants, même si tout le monde sait résoudre les équations du second degré.

La notion de "linéaire" se révèle être une notion clé de la programmation, car c'est un domaine où il existe des algorithmes performants.

Notons au passage que la satisfiabilité de l'ensemble de contraintes S est comprise dans une structure éventuellement élargie, car les algorithmes de satisfaction de contraintes sont parfois incomplets.

Environnement.

```

/* orthocentre(point,point,point,point) est vrai si %4 est l'orthocentre du
triangle (%1,%2,%3) */
orthocentre(a,b,c,h) ->
  perpendiculaire(a,b,c,h)
  perpendiculaire(b,c,a,h)
  perpendiculaire(c,a,b,h);

/*perpendiculaire(point,point,point,point) est vrai si la droite (%1,%2) est
perpendiculaire à la droite (%3,%4) */
perpendiculaire(a,b,c,d) ->
  { a=<x_a,y_a>, b=<x_b,y_b>, c=<x_c,y_c>, d=<x_d,y_d>,
  (x_a-x_b)*(x_c-x_d)+(y_a-y_b)*(y_c-y_d)=0};

exemple de requêtes et de résultats :
> orthocentre(<-1,0>,<2,0>,<1,2>,h);
résultat { h= <1,1> }
> orthocentre(a,<1,2>,<-3,5>,<-1,3>);
résultat {a= <-10,-9> }
> orthocentre(<-10,-9>,b,<-3,5>,<-1,3>);
résultat {b= <1,2> }
> orthocentre(<-10,-9>,<1,2>,c,<-1,3>);
résultat {c= <-3,5> }

```

Programme : II.1.b: Définition de l'orthocentre d'un triangle.

Bien sûr, comme pour Prolog pur, d'autres mécanismes sont associés à l'exécution de programmes. Nous nous contentons de les citer ici rapidement : possibilité d'utiliser la logique d'ordre supérieur, ajout et suppression de règles à la volée, coupure, gel et méta programmation (par exemple prédicat "findall" en Prolog III)

Enfin, sur le plan algorithmique et théorique, il est important de signaler que l'obtention d'une preuve de correction partielle est particulièrement aisée pour un programme CLP donné. Prolog est un langage de spécification tout aussi bien qu'il est un langage de programmation. Par contre, même si l'obtention "humaine" des preuves d'arrêt est facilitée par la structure des programmes Prolog, le problème de l'obtention de telles preuves reste indécidable

Sur le plan des langages et environnements de programmation, les langages CLP sont reconnus pour leur ergonomie et leur facilité d'emploi (comme Prolog). L'efficacité du code obtenu est améliorée par rapport à Prolog, reprenant les techniques de recherche opérationnelle (RO) comme résolveur de contraintes. Ils deviennent des concurrents importants dans bon nombre de types d'applications industrielles (par exemple dans les problèmes de planification).

En ce qui concerne les différences, qu'en est-il par rapport aux autres langages de programmation ? : typage (c'est possible, mais pas encore de consensus), compilation (en cours), ordinateurs parallèles (MIMD, SIMD, scalaire), aspect fonctionnel (lambda-Prolog), génie logiciel (il reste du travail, mais il y a déjà des débogueurs).

Environnement	Domaine	Référence
Prolog III	chaînes de caractères	[Colmerauer-90]
	arbres infinis	
	booléens	
	numérique (pseudo-linéaire) gauss + simplexe	
Prolog IV	numérique (intervalle)	[Benhamou et Touraivane-95]
	numérique (pseudo-linéaire) gauss + simplexe	
	arbres infinis	
	chaîne de caractères	
Chip	booléens	[Dincbas et al.-88]
	domaine finis	
	numériques (rationnels)	
CLP(R)	numérique (Gröbner)	[Jaffar, Michaylov, Stuckey, et Yap-92]
Risc-CLP	numérique (Gröbner)	
CAL	numérique (Gröbner)	[Sakai et Aiba]
CLP(BNR)	booléen	[Older et Benhamou-93b]
	intervalle	
Quad-CLP(R)	numérique (quadratique)	

Tableau II.1-a : Langages CLP.

Parmi les sujets de recherches actuels qui nous intéressent particulièrement au sein de la communauté CLP, nous citons un travail important dans le domaine des contraintes numériques basées sur des arithmétiques par intervalle. Nous n'abordons ce domaine que très succinctement. En particulier, les articles suivants [Benhamou-94], [N'Dong et Van Caneghem-95], [Older et Vellino-93] présentent des travaux très intéressants.

De même, un travail important est effectué pour la recherche de coopérations entre différentes approches [Marti-95] ou l'utilisation de l'approche intervalle dans d'autres domaines que les réels, [Older et Benhamou-93a].

Exemple.

Un exemple de petit programme utilisant les contraintes linéaires en Prolog III impossible à écrire (aussi fluide) avec un Prolog pur : définition de la relation orthocentre entre quatre points. cf. figure I.1.b. L'avantage de la programmation CLP dans ce cas vient de ce que le prédicat obtenu est parfaitement déclaratif, il est complètement réversible. Avec un Prolog pur, pour obtenir le même résultat, l'écriture de 4 programmes est nécessaire.

c) Quelques exemples de langages CLP.

Les exemples du tableau II.1-a sont tirés de [Van Hentenryck-92] et [Van Hentenryck-95].

En particulier, nous citons Quad-CLP(R) qui s'intéresse au même domaine que nous : les contraintes géométriques. Basé sur CLP(BNR), l'approche de Quad-CLP(R) et l'apport des travaux de Gilles Pesant sont étudiés dans la suite pour observer en quoi ces travaux et les nôtres sont complémentaires. En fait cette complémentarité est faible, bien que nos deux travaux soient sur des axes orthogonaux, et qu'ils partent de la même analyse : comment modéliser les contraintes quadratiques en utilisant l'emploi des contraintes linéaires ? [Pesant et Boyer-94]

La solution de Gilles Pesant est orthogonale à la notre. Elle propose de prendre une approximation des contraintes quadratiques par des contraintes linéaires. Malheureusement ces deux travaux ne peuvent pas être rapprochés. Le côté "approximation" de Quad-CLP(R) ne permet pas d'utiliser l'aspect "exact" de notre travail CLP(Géométrie).

Conclusion de ce chapitre:

Au cœur de la programmation logique par contraintes, se trouve le plus souvent un compromis entre l'expressivité du langage et le couple complétude/correction des algorithmes de satisfaction de contraintes. Ce compromis s'établit en fonction de la complexité des algorithmes de satisfaction de contraintes d'un domaine particulier.

Pour ce qui concerne la géométrie, son fort contenu formel place naturellement la géométrie dans les applications de Prolog et de la programmation logique avec contraintes. Cette place est connue depuis longtemps, depuis les essais de démonstration automatique en géométrie.

Traitée formellement, la géométrie est à mi chemin entre deux domaines connus par certains langages appartenant au paradigme CLP. Elle se situe entre le domaine des équations pseudo-linéaires sur les nombres rationnels de Prolog III, Prolog IV, CHIP, CLP(R) (les principaux langages de CLP) et le domaine des équations algébriques sur les nombres réels de Cal, Risc-CLP.

Traitée de façon informelle, la géométrie est pleinement décrite par la plupart des langages de CLP possédant un calcul sur les intervalles et ils sont déjà nombreux !

À mi-chemin entre tous ces domaines, se trouve déjà le travail de [Pesant-95] qui a proposé Quad-CLP(R). Il emploie une représentation approchée du domaine et combine des algorithmes d'approximation avec le traitement formel du domaine des équations linéaires.

Notre objectif est d'apporter un second travail à la croisée des mêmes chemins, en offrant une représentation formelle du domaine, qui soit compatible avec les algorithmes usuels de résolution de contraintes numériques linéaires (type Gauss).

II-2. CLP (Géométrie) : définition du domaine.

Introduction :

La définition d'un nouveau domaine CLP(Géométrie) nécessite tout d'abord la définition d'une représentation du domaine numérique utilisé en géométrie règle-compas. C'est à dire une représentation de l'ensemble des nombres constructibles, cf. Partie I.

La notion clé de ce manuscrit se situe dans la définition de cette représentation des nombres constructibles et d'une arithmétique associée. Cette représentation, pour être utilisée dans un langage CLP(Géométrie), doit supporter la définition des opérateurs arithmétiques et logiques $+$, $-$, $*$, $/$, $\sqrt{\quad}$, $=$, $>$. Elle doit permettre de plus l'élaboration d'algorithmes de résolution de systèmes de contraintes quadratiques. Les points relatifs à la résolution de systèmes de contraintes sont abordés dans le chapitre suivant II-3.

Ce chapitre se décompose ainsi : la section II.2-a envisage plusieurs représentations possibles pour les nombres constructibles. Parmi ces représentations classiques, la section II.2-b développe la plus simple et montre ses limites. La section II.2-c dépasse ces limites en proposant une représentation normale des nombres constructibles. C'est le cœur de ce chapitre. Les dernières sections abordent deux questions importantes : est-ce que la représentation est incrémentale ?, en II.2-d, et quelle est sa complexité ?, en II.2-e.

a) Définition d'un domaine et tentative de représentation.

Le domaine visé par CLP(Géométrie) est composé naturellement des réels dits nombres constructibles présentés dans la partie I, et d'une arithmétique étendue comportant les 4 opérations élémentaires, la racine carrée, et les prédicats d'égalité et d'ordre.

Il s'agit donc essentiellement d'un domaine sur des nombres algébriques, avec des racines de polynômes d'ordre 2 au maximum. Plusieurs idées peuvent mener à des représentations de ce domaine. Elles ne sont pas étudiées ici de manière exhaustive, mais suffisamment pour voir apparaître quelques inconvénients.

La première idée, la représentation naïve " $2+3\sqrt{5}$ " (avec une gestion formelle de la racine carrée) que nous utilisons pour nos calculs à la main, est présentée dans la section II.2-b.

Le domaine considéré est celui des nombres algébriques. Chaque élément peut être défini comme racine d'un polynôme réduit. L'addition, la soustraction, la multiplication, la division

de deux nombres sont alors définis comme la production d'un polynôme réduit ayant cet élément comme racine, cf. [Mutafian-80]. Ce n'est pas immédiat, mais c'est réalisable.

Le calcul d'une racine carrée est alors extrêmement simple. Par exemple, pour le calcul de la racine carrée de a , il s'agit de produire un polynôme dont cette racine carrée est solution. Le polynôme $P(X):X^2-a$ convient.

Cependant, cette représentation comporte certains inconvénients, elle ne permet pas aisément de distinguer les nombres algébriques conjugués (a et $-a$ dans le cas précédent). Par suite, la définition du prédicat d'égalité est malaisée, il faut recourir à un codage à la Thom. Quant au prédicat d'ordre, sa définition est encore plus difficile. De plus la structure linéaire d'espace vectoriel n'est pas conservée.

Comme la structure de l'espace des réels définie au cours des calculs va être de la forme $Q[\sqrt{a_1}] \dots [\sqrt{a_n}]$, il est assez simple de définir un élément primitif β de ce corps et d'identifier $Q[\sqrt{a_1}] \dots [\sqrt{a_n}]$ avec $Q[\beta]$, cf. [Mutafian-80].

Par suite, tous les éléments du domaine sont des polynômes en β . L'addition, la soustraction sont extrêmement aisées. Elles s'effectuent en ajoutant les coefficients des polynômes terme à terme. L'égalité est aussi une égalité terme à terme. La multiplication et la division, bien que de coût plus important, sont définies sans difficulté. La structure linéaire d'espace vectoriel est conservée ce qui avantage l'emploi du résolveur linéaire de Prolog III.

Seulement deux inconvénients apparaissent. Lors du calcul d'une nouvelle racine carrée, le corps $Q[\sqrt{a_1}] \dots [\sqrt{a_n}]$ devient $Q[\sqrt{a_1}] \dots [\sqrt{a_n}][\sqrt{a_{n+1}}]$ et β devient β' . Les expressions des éléments en fonction de β doivent être reconstruites pour s'exprimer en fonction de β' . Ce n'est pas difficile, mais coûteux. Enfin, la définition du prédicat d'ordre n'est pas aisée.

b) Représentation à l'aide d'une structure naïve.

Qu'en est-il d'une gestion naïve du domaine s'appuyant sur la structure $Q[\sqrt{a_1}] \dots [\sqrt{a_n}]$ obtenue par adjonction successive de racines carrées ?

Cette tentative est étudiée plus profondément, non pas parce qu'elle ne comporte aucun inconvénient, mais parce qu'une modification de cette représentation est envisagée dans la section suivante II.2-c, qui nous permet de l'adopter avec profit. Par ailleurs, cette tentative constitue un exemple didactique pour la définition de structures récursives ne comportant pas de difficultés mathématiques importantes.

Mais que signifie "représentation naïve du corps $Q[\sqrt{a_1}] \dots [\sqrt{a_n}]$ " ? C'est ce que nous allons voir ci-dessous immédiatement.

Un nombre réel x de $Q[\sqrt{a}]$ où a est rationnel, s'écrit $x=p+q.\sqrt{a}$, avec p, q rationnels; il suffit donc de prendre deux nombres rationnels p, q , pour représenter x . Cette représentation avec deux nombres s'applique chaque fois que l'on veut représenter des éléments d'un corps de degré 2.

Par conséquent, pour représenter les éléments de $K_n=Q[\sqrt{a_1}] \dots [\sqrt{a_n}]$, il suffit de considérer la structure comme récursive, c'est à dire prendre K_n comme corps de degré au plus 2 sur $K_{n-1} = Q[\sqrt{a_1}] \dots [\sqrt{a_{n-1}}]$. Un élément X de K_n est représenté par (P,Q) , où P et Q sont des

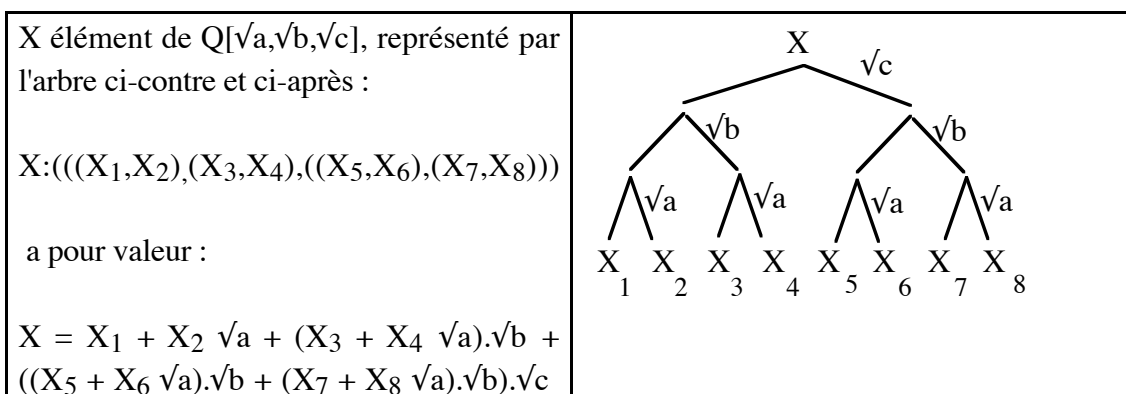


Figure II.2-a: Représentation d'un nombre.

éléments de K_{n-1} , avec $X=P+Q.\sqrt{a_n}$. Il n'y a pas de restriction sur a_n , excepté bien sûr que a_n soit aussi un élément de K_{n-1} .

Un élément x de $Q[\sqrt{a_1}]...[\sqrt{a_n}]$ est donc représenté par un arbre binaire complet (cf. figure II.2-a). Les feuilles de cet arbre sont des coefficients rationnels. Les branches sont étiquetées alternativement pour la branche gauche par 1, et pour la branche droite par $\sqrt{a_i}$ pour la profondeur i . La valeur de x est obtenue en prenant le poids total de l'arbre, les coefficients portés par les branches sont des coefficients multiplicatifs.

Avant de passer à l'expression des opérations arithmétiques sur cette structure, notons l'un des inconvénients immédiats de cette représentation. Cette représentation est de complexité exponentielle en espace selon le nombre de racines carrées employées. Nous pouvons faire ce commentaire pour les deux représentations au II.2-a. Si nous nous limitons à un surcoût d'un facteur 1000 pour la complexité spatiale par rapport aux rationnels, l'emploi de cette représentation est limité à des situations où interviennent moins de 10 racines imbriquées. En effet, la place mémoire nécessaire à la représentation d'un nombre défini avec 10 racines distinctes correspond au stockage d'un arbre binaire de hauteur 10 avec des rationnels comme feuilles. Ce qui nécessite une place mémoire de $2^{10} = 1024$ coefficients rationnels.

Cette structure récursive naturelle permet une définition naturelle de l'addition, de la soustraction, de la multiplication et de la division.

Au passage, les définitions de la soustraction et de la division peuvent être obtenues directement et sans effort à partir des définitions de l'addition et de la multiplication. La structure d'espace vectoriel et l'utilisation d'un résolveur de contraintes linéaires permettent la définition de la soustraction et de la division comme opérations inverses de l'addition et de la multiplication car les contraintes qui dépendent de ces dernières sont linéaires. Ce que nous établissons ci-après :

- Dans cette structure, l'addition est une application linéaire, comme dans tout espace vectoriel. Elle s'effectue membre à membre. Par suite, comme l'addition entre nombres rationnels introduit des contraintes linéaires, l'addition entre nombres constructibles introduit des contraintes linéaires aussi.

```

/* ng_plus(L_Ext, ?nc,?nc,?nc): %3 = %1 + %2 dans l'extension %0 */
ng_plus(<>,_A,_B,_A+_B) -> ;
ng_plus(<_E>.L,<_A1,_A2>,<_B1,_B2>,<_C1,_C2>) ->
  ng_plus(L,_A1,_B1,_C1)
  ng_plus(L,_A2,_B2,_C2);

/* ng_minus(L_Ext, ?nc,?nc,?nc): %3 = %1 - %2 dans l'extension %0 */
ng_minus(L,A,B,C) ->
  ng_plus(L,B,C,A);

```

Figure II.2-b : Définition de l'addition et de la soustraction.

- Dans cette structure, la multiplication entre une constante et une variable est une application linéaire également ! (la démonstration est donnée plus loin) En fait c'est un résultat paradoxal puisque la multiplication entre deux variables n'est pas une application linéaire. Entre deux structures a priori complexes il risque d'en être de même, or ce n'est pas le cas. C'est même un résultat relativement courant : de même pour la définition d'une arithmétique sur les polynômes, sous certaines conditions portant sur la représentation des polynômes, la multiplication entre un polynôme et une variable représentant un polynôme est elle aussi une application linéaire. La linéarité de la multiplication de deux nombres constructibles dont l'un est une variable permet de définir la division comme son inverse car les contraintes qui dépendent de la multiplication avec une constante sont linéaires. Bien sûr cette définition de la division utilise le résolveur linéaire.

Notation.

Nous considérons dans la suite un corps $K_n = Q[\sqrt{a_1}] \dots [\sqrt{a_n}]$ défini comme extension du corps $K_{n-1} = Q[\sqrt{a_1}] \dots [\sqrt{a_{n-1}}]$ par adjonction de $\sqrt{a_n}$, où a_n est un élément positif de K_{n-1} .

K_0 est le corps Q des rationnels.

Un élément X de K_n est noté $X:(P_x, Q_x)$ où P_x , et Q_x sont des éléments de K_{n-1} . Un élément X de K_0 est un rationnel noté tout simplement X .

Addition et soustraction.

L'addition de $X:(P_x, Q_x)$ avec $Y:(P_y, Q_y)$, tous deux éléments de K_n , donne $Z:(P_z, Q_z)$ tel que $P_z = P_x + P_y$ et $Q_z = Q_x + Q_y$.

[[preuve]] Soient $X = P_x + Q_x \cdot \sqrt{a_n}$, $Y = P_y + Q_y \cdot \sqrt{a_n}$,

alors $X + Y = (P_x + P_y) + (Q_x + Q_y) \cdot \sqrt{a_n}$ dans R [[fin de la preuve]]

Nous avons donc, en distinguant les additions dans K_n (notées $+^n$), de celles de K_{n-1} (notées $+^{n-1}$), et de celles dans Q (notées $+^0$) : $X +^n Y : (P_x +^{n-1} P_y, Q_x +^{n-1} Q_y)$.

La soustraction est l'inverse de l'addition. La soustraction de Y à X , tous deux éléments de K_n , donne Z tel que $X = Y +^n Z$. Quand X , Y sont connus et Z est variable, l'addition précédente $Y + Z$ définit un système de contraintes linéaires entre les coefficients de X , Y et Z . Par conséquent, si X et Y sont instanciés, le résolveur linéaire de Prolog III donne Z .

```

/* ng_mult(L_Ext, ?nc,?nc,?nc): %3 = %1 * %2 dans l'extension %0 */
ng_mult(<>, _A, _B, _A*_B) -> ;
ng_mult(<_E>.L, <_A1, _A2>, <_B1, _B2>, <_C1, _C2>) ->
  ng_mult(L, _A1, _B1, _T1)
  ng_mult(L, _A1, _B2, _T2)
  ng_mult(L, _A2, _B1, _T3)
  ng_mult(L, _A2, _B2, _T4)
  ng_mult(L, _E, _T4, _T5)
  ng_plus(L, _T1, _T5, _C1)
  ng_plus(L, _T2, _T3, _C2);

```

Figure II.2-c : Définition de la multiplication

Les prédicats Prolog correspondants à ces définitions sont donnés ci-après, figure II.2-b. L'addition s'effectue récursivement selon la profondeur de l'arbre. Le premier argument du prédicat "ng_plus" représente K_n sous la forme de la liste des $\sqrt{a_i}$, la récurrence s'effectue en consommant cette liste. La soustraction est définie comme l'inverse de l'addition.

Multiplication et division.

La multiplication de $X:(P_x, Q_x)$ avec $Y:(P_y, Q_y)$, tous deux éléments de K_n , donne $Z:(P_z, Q_z)$ tel que $P_z = P_x * P_y + Q_x * Q_y * a_n$ et $Q_z = P_x * Q_y + Q_x * P_y$.

[[preuve]] Soient $X = P_x + Q_x \cdot \sqrt{a_n}$, et $Y = P_y + Q_y \cdot \sqrt{a_n}$,

alors le produit $X * Y$ vaut $(P_x * P_y + Q_x * Q_y * a_n) + (P_x * Q_y + Q_x * P_y) \cdot \sqrt{a_n}$ dans R

[[fin de la preuve]]

La multiplication par a_n est apparue avec le produit des $\sqrt{a_n}$. Cette multiplication est licite dans K_{n-1} , car a_n est un élément de K_{n-1} .

Nous avons donc, en distinguant les additions et multiplications dans K_n (notées $+^n$ et $*^n$), de celles de K_{n-1} :

$X *^n Y : (P_x *^{n-1} P_y +^{n-1} Q_x *^{n-1} Q_y *^{n-1} a_n, P_x *^{n-1} Q_y +^{n-1} Q_x *^{n-1} P_y)$; avec $*^0$ qui correspond à la multiplication des rationnels.

Sur le plan de la complexité algorithmique, cette définition "naïve" de la multiplication n'est peut-être pas optimale. L'utilisation d'une représentation matricielle des nombres constructibles doit permettre l'emploi d'algorithmes de multiplication (matricielle) plus performant.

La division est l'inverse de la multiplication. La division de X par Y , tous deux éléments de K_n , donne Z tel que $X = Y *^n Z$. Quand X, Y sont connus et Z est variable, la multiplication précédente $Y * Z$ définit un système de contraintes linéaires entre les coefficients de X, Y et Z .

[[preuve]]

En effet cela peut s'observer par récurrence :

dans $K_0 = Q : Z * Y = X$, la seule variable est Z , la contrainte est linéaire. Le système possède une variable, une équation, il a donc une solution unique si $Y \neq 0$.

```

/* ng_sqrt(L_Ext, nc,->L_Ext,->nc): %3 dans l'extension %2 est égale à √%1 dans
l'extension %0 */
ng_sqrt(L,A,<A>.L,<Z,O>) ->
zero(L,Z)
one(L,O);

```

Figure II.2-d : Définition de la racine carrée (provisoire).

Dans K_n , la multiplication $Z*Y$ correspond à deux équations $P_y*P_z+Q_y*Q_z*a_n = P_x$ et $P_y*Q_z+Q_y*P_z = P_x$, les seules variables sont P_z, Q_z , les deux contraintes sont linéaires. Il y a autant d'équations que d'inconnues et par récurrence le système est linéaire.

[[fin de la preuve]]

Par conséquent, si X et Y sont instanciés, le résolveur linéaire de Prolog III donne Z .

Le prédicat Prolog correspondant à la multiplication est donné figure II.2-c. La multiplication s'effectue récursivement selon la profondeur de l'arbre en consommant le premier argument du prédicat "ng_mult" qui représente K_n .

Extraction d'une racine carrée (provisoire).

La définition du calcul de la racine carrée est immédiate. La racine carrée de $X:(P_x, Q_x)$, élément positif de K_n , donne $Z:(P_z, Q_z)$, élément de $K_{n+1}=K_n[\sqrt{X}]$ tel que $P_z=0$ et $Q_z=1$. Le prédicat Prolog correspondant à cette définition est donné figure II.2-d.

Ordre et égalité.

L'égalité et l'ordre entre deux variables ne sont pas définis directement, mais par le biais d'un prédicat "positif ou nul" :

Un élément $X:(P_x, Q_x)$ de K_n est positif ou nul dans les trois cas suivants seulement :

- a) $P_x \geq 0$ et $Q_x \geq 0$,
- b) $P_x \geq 0$ et $Q_x \leq 0$ et $(P_x)^2 - (Q_x)^2 \cdot a_n \geq 0$
- c) $P_x \leq 0$ et $Q_x \geq 0$ et $(Q_x)^2 \cdot a_n - (P_x)^2 \geq 0$.

Le prédicat Prolog correspondant à cette définition est donné figure II.2-e. L'étude de cas s'effectue récursivement selon la profondeur de l'arbre en consommant le premier argument du prédicat "ng_positive" qui représente K_n .

Par suite :

$X \geq Y$ si et seulement si $X - Y \geq 0$,

$X = 0$ si et seulement si $X \geq 0$ et $X \leq 0$, et enfin

$X = Y$ si et seulement si $X - Y = 0$, c'est à dire $X - Y \geq 0$ et $X - Y \leq 0$.

Commentaires.

L'arithmétique obtenue possède plusieurs défauts. Comme il a déjà été dit : la représentation utilisée est de complexité exponentielle en espace selon le nombre de racines carrées employées. Le coût des opérations est lui aussi exponentiel. Ce point est étudié au II.2-e. De manière informelle, en notant n le nombre d'extensions, l'addition et la soustraction sont

```

%ng_positive(L_Ext,nc) vrai si %1 est positif dans %0
ng_positive(<>,_X) -> { _X >= 0 };
ng_positive(<_E>.L,<_P,_Q>) ->
  ng_positive(L,_P) / %coupure verte
  ng_positive_P(L,_E,_P,_Q);
ng_positive(<_E>.L,<_P,_Q>) ->
  ng_positive(L,_Q)
  ng_mult(L,_P,_P,_X)
  ng_mult(L,_Q,_Q,_Y)
  ng_mult(L,_Y,_E,_Z)
  ng_minus(L,_Z,_X,_S)
  ng_positive(L,_S);

ng_positive_P(L,_E,_P,_Q) ->
  ng_positive(L,_Q);
ng_positive_P(L,_E,_P,_Q) ->
  ng_mult(L,_P,_P,_X)
  ng_mult(L,_Q,_Q,_Y)
  ng_mult(L,_Y,_E,_Z)
  ng_minus(L,_X,_Z,_S)
  ng_positive(L,_S);

```

Figure II.2-e : Définition de positif ou nul.

de complexité algorithmique linéaire selon le nombre de coefficients, donc en $O(2^n)$. La multiplication est en $O(5^n)$, et le calcul de l'égalité est de complexité au moins aussi importante. Dans l'optique d'une complexité limitée à 1000 fois celle de l'arithmétique rationnelle, cette représentation naïve est limitée à 4 extensions quadratiques. Car $5^4 = 625$ et $5^5 = 3125$.

Dans cette optique, cette représentation est impropre à l'utilisation au delà de 4 extensions, mais ce n'est pas son principal inconvénient.

L'égalité ainsi définie entre deux nombres X et Y, outre sa complexité, a pour principal défaut de ne pas correspondre à l'égalité sur les arbres représentant X et Y. L'unification entre termes Prolog (entre les arbres ici) ne correspond pas à l'unification entre les nombres. C'est un point très important dans le contexte qui est le notre, car l'unification est à la base du moteur d'inférence Prolog. L'égalité sur ces arbres est utilisée implicitement dans les définitions de la soustraction et de la division.

Par exemple, les deux éléments X et Y qui valent respectivement $\sqrt{2}.\sqrt{3}$ et $\sqrt{6}$ sont égaux dans R, alors que leurs représentations respectives sont différentes dans $Q[\sqrt{2}][\sqrt{3}][\sqrt{6}]$:

X:(((0,0),(0,1)),((0,0),(0,0))) et
Y:(((0,0),(0,0)),((1,0),(0,0)))

Le prédicat d'égalité défini plus haut assure que X et Y sont effectivement égaux dans $Q[\sqrt{2}][\sqrt{3}][\sqrt{6}]$, et c'est heureux car nous reconnaissons aisément l'égalité dans R. Mais les représentants ne sont pas unifiables.

NB. Toutes les égalités ne sont pas aussi aisées à découvrir. Par exemple, dans $Q[\sqrt{2}][\sqrt{3}][\sqrt{6}][\sqrt{5+2\sqrt{6}}]$, les nombres $X = \sqrt{2} + \sqrt{3}$ et $Y = \sqrt{5+2\sqrt{6}}$ sont égaux comme permet de

le vérifier le prédicat d'égalité, ce qui n'est pas évident à priori, et qui apparaît encore moins évident avec leur représentation :

$$X:(((0,1),(1,0)),((0,0),(0,0))),(((0,0),(0,0)),((0,0),(0,0))) \text{ et}$$

$$Y:(((0,0),(0,0)),((0,0),(0,0))),(((1,0),(0,0)),((0,0),(0,0)))$$

D'une manière générale, l'égalité d'expressions faisant intervenir des racines carrées imbriquées, et la simplification de ces expressions est un problème difficile. Les logiciels de calcul formel n'ont pas tous complètement résolu ces problèmes. Sur ce point, des précisions sont données au chapitre III.1.

c) Représentation normale du domaine et arithmétique associée.

La structure naïve présentée au paragraphe précédent doit être améliorée pour que l'égalité corresponde à l'unification. Il faut que l'égalité entre deux nombres constructibles soit définie par l'égalité entre les arbres binaires qui représentent ces deux nombres. Ensuite, vient la définition de l'arithmétique associée.

Représentation normale.

Les objectifs que nous nous sommes fixés peuvent être atteints si le choix de l'extension algébrique peut se faire de façon canonique, et entraîne une représentation unique¹⁷. Mais d'une part le choix d'une extension n'est pas canonique. Pourquoi ? Par exemple, prenons $A = \sqrt{10} + \sqrt{21}$ et $B = \sqrt{2} \cdot \sqrt{5} + \sqrt{3} \cdot \sqrt{7}$; il est clair que $A = B$. Ces deux représentations correspondent donc au même nombre. A est donné dans $Q[\sqrt{10}, \sqrt{21}]$, B dans $Q[\sqrt{2}, \sqrt{3}, \sqrt{5}, \sqrt{7}]$. Ces deux extensions sont aussi respectables l'une que l'autre. Chaque extension présente des qualités (en particulier l'unicité de représentation des éléments), aucune ne possède ce caractère d'évidence que l'usage accorde à une représentation canonique. D'autre part, dans $Q[\sqrt{2}, \sqrt{3}, \sqrt{5}, \sqrt{7}, \sqrt{10}, \sqrt{21}]$, A et B s'expriment simultanément, ce nombre possède donc deux représentations. Par suite le choix d'une extension particulière (par exemple maximale) n'assure pas forcément une représentation unique.

Cependant, si l'égalité entre extensions ne permet pas d'en choisir une, il est possible, étant donnée une extension particulière, d'avoir une représentation unique des éléments de cette extension. Nous nommerons "normale" ces extensions algébriques quadratiques et les représentations des nombres associées. C'est la solution à notre problème. Nous choisissons arbitrairement une extension à condition qu'elle soit normale, et exprimons tous les nombres dans cette extension. Parmi les extensions précédentes : $Q[\sqrt{2}, \sqrt{3}, \sqrt{5}, \sqrt{7}, \sqrt{10}, \sqrt{21}]$, $Q[\sqrt{2}, \sqrt{3}, \sqrt{5}, \sqrt{7}]$, $Q[\sqrt{10}, \sqrt{21}]$, et $Q[\sqrt{10} + \sqrt{21}]$, toutes, exceptée la première, sont normales. Le nombre représenté par A ou B possède donc une unique représentation dans ces extensions algébriques, A dans $Q[\sqrt{10}, \sqrt{21}]$ et B dans $Q[\sqrt{2}, \sqrt{3}, \sqrt{5}, \sqrt{7}]$.

[[preuve à construire d'après ce qui suit laissée au lecteur.]]

¹⁷Une autre solution consiste à choisir une représentation canonique des nombres constructibles, ce qui est possible avec polynôme minimal + codage à la Thom, mais difficilement compatible avec l'emploi des résolveurs linéaires usuels des langages représentant l'approche CLP (cf. II.2-a).

La caractérisation formelle et une direction pour l'obtention de la représentation normale d'un nombre constructible sont données par la définition suivante :

Définition : 1. Il y a équivalence entre les définitions suivantes :

- une extension $K=Q[\sqrt{a_1}]...[\sqrt{a_n}]$ est normale si les éléments de K possèdent une écriture unique en fonction des $\sqrt{a_i}$.
- une extension $K=Q[\sqrt{a_1}]...[\sqrt{a_n}]$ est normale si les $(\sqrt{a_i})_{i:1...n}$ sont linéairement indépendants, on dit aussi que les $(\sqrt{a_i})_{i:1...n}$ sont libres.

2. L'écriture que nous employons $r = \sum_i r_i \cdot \left(\prod_{k \in E_i} \sqrt{a_k} \right)$ d'un nombre constructible est normale si $Q[\sqrt{a_1}]...[\sqrt{a_n}]$ est normale.

Il s'ensuit la propriété immédiate : si $K_n = Q[\sqrt{a_1}]...[\sqrt{a_n}]$ est normal, alors $K_{n-1} = Q[\sqrt{a_1}]...[\sqrt{a_{n-1}}]$ est normal, et de même $K_i = Q[\sqrt{a_1}]...[\sqrt{a_i}]$ pour $i=1...n$. La "normalité" d'une extension est une propriété transmise aux corps intermédiaires. Cette propriété est utilisée implicitement pour les récurrences sur n qui suivent.

Le lemme suivant sera utile pour l'obtention d'une extension normale :

Lemme II.a : Il y a équivalence entre les énoncés :

1. \sqrt{a} est libre dans $Q[\sqrt{a_1}]...[\sqrt{a_n}]$ normal.
2. a n'est pas un carré dans $Q[\sqrt{a_1}]...[\sqrt{a_n}]$ normal.

[[preuve]]

Par contraposée montrons que $\neg 1 \Leftrightarrow \neg 2$ en deux étapes $\neg 1 \Rightarrow \neg 2$, et $\neg 2 \Rightarrow \neg 1$.

1^{ère} étape : $\neg 1$ signifie que \sqrt{a} s'exprime en fonction de $(\sqrt{a_k})_{k:1...n}$. Cette expression, mise au carré, donne a fonction des $(\sqrt{a_k})_{k:1...n}$, c'est à dire $\neg 2$.

2^{nde} étape : $\neg 2$ signifie que a est le carré E^2 d'une expression E fonction des $(\sqrt{a_k})_{k:1...n}$. Donc cette expression ou son opposé $(-E)$, est \sqrt{a} , c'est à dire $\neg 1$.

[[fin de la preuve]]

Comment obtenir $Q[\sqrt{a_1}]...[\sqrt{a_n}]$ normale ? Il suffit de réduire la liste des $(\sqrt{a_i})_{i:1...n}$ à un sous-ensemble linéairement indépendant $(\sqrt{a_j})_{j:1...k}$. Nous obtenons une extension normale par reconstructions de $Q[\sqrt{a_1}]...[\sqrt{a_n}]$ à partir de Q en adjoignant successivement chaque racine carrée $\sqrt{a_i}$. A chaque adjonction d'une racine carrée $\sqrt{a_i}$, il faut vérifier que $\sqrt{a_i}$ est libre dans $Q[\sqrt{a_1}]...[\sqrt{a_{i-1}}]$. C'est le cas si a_i n'est pas un carré dans $Q[\sqrt{a_1}]...[\sqrt{a_{i-1}}]$ d'après le lemme II.a. Si les $(\sqrt{a_j})_{j:1...i}$ sont linéairement indépendants, le processus d'adjonction peut continuer. Sinon, $\sqrt{a_i}$ s'exprime en fonction des $(\sqrt{a_j})_{j:1...i-1}$, et par conséquent, $\sqrt{a_i}$ est inutile, elle peut être remplacée par une combinaison linéaire de $(\sqrt{a_j})_{j:1...i-1}$, et le processus d'adjonction peut continuer avec $(\sqrt{a_j})_{j:1...i-1}$. Le processus de réduction par adjonction est donné par le prédicat informel de la figure II.2-f.

```

/* reduit_base(l_exp_const,[],->l_exp_const): Q[%1]=Q[%3], %3 est normal. */
reduit_base(<>,1,1) ->
reduit_base(<a>.l,e,f)->
  carre(a,e)
  reduit_base(l,e,f);
reduit_base((<a>.l,e,f)->
  not(carre(a,e))
  reduit_base(l,<a>.e,f);

```

figure II.2-f : Réduction d'une liste de racines.

Le principal reste encore à écrire, la recherche de racine carrée dans une extension quadratique. C'est le résultat formel le plus important de ce chapitre. Pour y arriver, nous avons besoin du lemme suivant :

Lemme II-b : Soit $\mathbf{a} = u + v.\sqrt{a_n}$ un élément de $K_n = Q[\sqrt{a_1}] \dots [\sqrt{a_n}]$, une extension normale; u, v, a_n sont des éléments de $K_{n-1} = Q[\sqrt{a_1}] \dots [\sqrt{a_{n-1}}]$ tels que $v \neq 0$. Notons $\Delta = u^2 - a_n v^2$ et Y la solution positive de (e) : $a_n.Y^2 - u.Y - \frac{v^2}{4} = 0$ ($Y = \frac{u + \sqrt{\Delta}}{2.a_n}$). Alors, il y a équivalence entre les énoncés suivants :

1. \mathbf{a} est un carré dans K_n .
2. Δ et Y sont des carrés dans K_{n-1} .

[[preuve]] (les manipulations utilisées rapidement ci-après sont explicitées "calmement" dans la proposition II-C)

1. \Rightarrow 2. est immédiat car 1. implique l'existence de x et y dans K_{n-1} tels que $(x+y.\sqrt{a_n})^2 = \mathbf{a}$. C'est à dire, en développant et en utilisant le fait que K_n est normal, (1) $u = x^2 + y^2$ et (2) $v = 2.x.y$. En remplaçant x par $v/2y$ selon (2) dans (1), il vient (1') $a_n.y^4 - u.y^2 - \frac{v^2}{4} = 0$.

Donc y^2 est solution de l'équation au carré de (e), donc y est racine de Y . Comme $\Delta = (2.a_n.Y - u)^2$, 2. en découle.

2. \Rightarrow 1. vient de ce que y , la racine carrée de Y dans K_{n-1} , permet de définir $x = v/2.y$, élément de K_{n-1} tel que $x+y\sqrt{a_n}$ dans K_n soit une racine carrée de \mathbf{a} .

[[fin de la preuve]]

Nous avons donc le résultat suivant qui montre que la recherche de racine carrée dans une suite d'extensions algébriques quadratiques est décidable. C'est le résultat principal de ce chapitre.

Proposition II.C : Soit \mathbf{a} un élément de $K_n = Q[\sqrt{a_1}] \dots [\sqrt{a_n}]$, une extension normale. Alors, il peut être décidé si \mathbf{a} est un carré dans K_n et si c'est le cas, la racine carrée positive de \mathbf{a} peut être exhibée.

[[preuve]] Nous démontrons le résultat par récurrence sur l'entier n représentant le nombre d'extensions quadratiques successives.

Cas $n=0$, c'est le problème de recherche d'une racine carrée rationnelle d'un rationnel.

Si a est un entier naturel, un calcul satisfaisant est proposé dans [Rolfe-87].

Si a est un rationnel non entier, soit p/q un représentant de a (pas nécessairement canonique). Alors pq/q^2 est un autre représentant de a . La recherche d'une racine carrée de a se ramène à la recherche d'une racine carrée de pq . Si b est une racine carrée positive rationnelle de pq , alors $b/|q|$ est une racine carrée rationnelle de a . Si pq n'a pas de racine carrée rationnelle, alors a non plus.

Soit $n>0$, nous supposons la propriété vérifiée pour tout entier strictement inférieur à n , montrons la propriété pour n . Comme a est un élément de $K_n = Q[\sqrt{a_1}] \dots [\sqrt{a_n}]$, a s'écrit $a = u + v \cdot \sqrt{a_n}$ avec u et v deux éléments de $K_{n-1} = Q[\sqrt{a_1}] \dots [\sqrt{a_{n-1}}]$.

a possède une racine carrée dans $K_n \iff$ il existe b dans K_n tel que $b^2 = a$.

- \iff il existe x, y dans K_{n-1} tels que $(x + y \cdot \sqrt{a_n})^2 = a$, (en notant $b = x + y \cdot \sqrt{a_{n-1}}$).

- \iff il existe x, y dans K_{n-1} tels que $u + v \cdot \sqrt{a_n} = x^2 + y^2 \cdot a_n + 2 \cdot x \cdot y \cdot \sqrt{a_n}$ (en développant).

- \iff il existe x, y dans K_{n-1} tels que $u = x^2 + y^2 \cdot a_n$ et $v = 2 \cdot x \cdot y$ (en utilisant l'hypothèse K_{n-1} est normale).

Nous démontrons la proposition précédente par disjonction des cas. Deux cas sont possibles : $v = 0$ ou $v \neq 0$.

Cas où $v = 0$.

- \iff il existe x, y dans K_{n-1} tels que $u = x^2 + y^2 \cdot a_n$ et $0 = 2 \cdot x \cdot y$.

- \iff il existe x, y dans K_{n-1} tels que $u = x^2 + y^2 \cdot a_n$ et ($x = 0$ ou $y = 0$).

- \iff il existe x, y dans K_{n-1} tels que $u = x^2$ ou $u = y^2 \cdot a_n$.

- \iff il existe x, y dans K_{n-1} tels que $u = x^2$ ou $u/a_n = y^2$.

L'hypothèse de récurrence s'applique dans les deux branches du "ou" pour déterminer si u et u/a_n (éléments de K_{n-1} , car $a_n \in K_{n-1}$) possèdent une racine dans K_{n-1} et pour la déterminer. Ce qui permet de conclure pour a .

Ce qui clôt cette branche de la disjonction de cas.

Cas où $v \neq 0$. Alors, si x et y existent, ils sont non nuls.

- \iff il existe x, y dans K_{n-1} tels que $u = (v/2 \cdot y)^2 + y^2 \cdot a_n$ et $v = 2 \cdot x \cdot y$

(substitution de x par $v/2 \cdot y$).

- \iff il existe x, y dans K_{n-1} tels que $a_n \cdot y^4 - u \cdot y^2 - v^2/4 = 0$ et $v = 2 \cdot x \cdot y$.

- \iff il existe x, y dans K_{n-1} tels que $a_n \cdot Y^2 - u \cdot Y - v^2/4 = 0$, $v = 2 \cdot x \cdot y$, et $Y = y^2$

(renommage de y^2 par Y).

L'existence de y , racine carrée de Y , est assurée par le lemme II-b. Cela assure l'équivalence entre les deux propositions précédentes.

L'équation du second degré en Y obtenue se résout de manière classique :

Le calcul du discriminant donne $\Delta = u^2 - 4 \cdot a_n \cdot v^2/4 = u^2 - a_n \cdot v^2$. Le lemme II-b assure que Δ possède une racine carrée positive δ dans K_{n-1} . L'hypothèse de récurrence permet de calculer δ .

Il vient $Y = \frac{u \pm \delta}{2.a_n}$. La solution positive est $\frac{u+\delta}{2.a_n}$, car a_n est positif. Le lemme II-b assure l'existence de y , racine carrée de Y . L'hypothèse de récurrence permet de calculer y .

Par conséquent cette branche de la disjonction par cas se conclut avec le calcul par récurrence d'une racine carrée positive δ de $u^2 - a_n.v^2$, et d'une racine positive y de $\frac{u+\delta}{2.a_n}$.

Ce qui clôt la démonstration par disjonction des cas.

Ce qui conclut la démonstration de la proposition pour n .

Par suite, la proposition est démontrée par récurrence, ce qui termine la démonstration.
[[fin de la preuve]]

Dans la suite, nous utilisons une représentation normale, sauf mention contraire. Ceci assure que l'égalité correspond à l'unification.

Arithmétique.

L'arithmétique correspondant à une représentation normale est composée pour l'essentiel des prédicats définis à la section II.2-b précédente. Cependant, il y a deux modifications :

- le calcul de la racine carrée est remplacé par l'algorithme présenté dans la preuve de la proposition II-C. Attention, l'élément sur lequel porte l'extraction de racine carrée doit être connu. La proposition II-C n'a pas de sens pour une variable.
- l'égalité entre deux nombres constructibles correspond à l'égalité entre les arbres représentant ces deux nombres. Elle s'effectue donc via l'algorithme d'unification.

Notons que précédemment la représentation naïve donnait des définitions ambiguës pour la soustraction et de la division. L'ambiguïté venait de l'absence d'unicité de représentation des nombres. Avec une représentation normale, les définitions élégantes de ces deux opérations sont justifiées. La beauté de ces deux prédicats, qui s'exprime par leur simplicité, assure leur correction. Ceci n'est pas négligeable.

Avec une représentation normale il est possible d'utiliser l'algorithme d'unification et le résolveur d'équations linéaires sur les arbres Prolog comme une boîte noire. Ceci est démontré plus loin en II.3-d, dans le chapitre sur l'algorithme de résolution. Cette possibilité est utilisée pour les définitions de l'inverse de l'addition et de la multiplication.

L'utilisation de résolveur/boîtes noires est parfois dangereux. Le danger vient de ce qu'il se peut qu'une égalité entre deux nombres doive être vérifiée de manière interne par le résolveur. C'est le cas lors d'introduction d'équations géométriques redondantes. Avec une représentation normale, le danger est évité.

L'égalité est vérifiée implicitement par le résolveur de contraintes, car elle correspond à l'égalité entre arbres. Avec une représentation non normale ce n'est pas le cas, il est cependant

bien sûr pensable de "gérer" cette vérification "à la main". Mais cela contrarie l'emploi du résolveur de contraintes comme boîte noire.

Notre démarche étant fondée sur l'emploi d'un résolveur linéaire cette arithmétique normale est nécessaire.

d) Extensibilité du domaine.

La représentation du domaine présentée au II.2-c est statique, au sens où la connaissance de la liste d'extensions quadratiques est supposée acquise. Supposons que les extensions apparaissent dynamiquement. C'est le cas potentiellement à chaque extraction de racine carrée. Comment prendre ceci en compte ?

Propositions

Plusieurs possibilités sont envisageables :

-1- Avoir des couples (nombre *dans* extension correspondante), et savoir faire les opérations arithmétiques entre ces différents couples. Par exemple, savoir additionner $((0,1),(2,3))$ dans $Q[\sqrt{2},\sqrt{1+\sqrt{2}}]$ avec $((-1,1)$ dans $Q[\sqrt{3}]$.

-2- Avoir une seule liste d'extensions qui se prolonge dynamiquement, et des nombres constructibles ayant des représentations sur les préfixes de cette liste. Il faut alors savoir faire des opérations comme l'addition de $((0,1),(2,3))$ et de $(-1,1)$ dans $Q[\sqrt{2},\sqrt{1+\sqrt{2}},\dots]$.

-3- Avoir une seule liste d'extensions qui se prolonge dynamiquement et des nombres constructibles dont les représentations dépendent de cette liste et se modifient dynamiquement. Dans ce dernier cas, l'arithmétique doit aussi comporter une part dynamique.

Chacune des trois possibilités envisagées a été mise en oeuvre. Elles sont donc réalisables. Les plus élaborées, celles qui consomment le moins d'espace en mémoire et qui demandent l'effort de programmation le plus important sont les deux premières. La dernière est "gloutonne". Mais c'est la plus efficace du fait du faible nombre d'extensions observées dans les cas concrets. L'intelligence des deux premières coûte plus qu'elle ne rapporte, à cause de la faible complexité des situations observées. Il est plus que probable que dans des cas plus complexes l'efficacité, tant en espace qu'en temps, soit dans l'ordre inverse.

Mise en place.

Comment mettre en place -3- ? L'idée consiste à geler sur la liste des racines carrées l'extension des structures des représentants des nombres et des opérations qui portent sur ces nouveaux éléments. Cela est-il possible pour les représentants et toutes les opérations ?

Dans le cas de l'ajout d'une nouvelle extension quadratique :

- Pour les données :

- pour une variable : il faut remplacer l'arbre V représentant cette variable par un noeud ayant pour fils gauche V , et pour fils droit un arbre binaire complet de même hauteur que V ayant des nouvelles variables pour feuilles.

- pour un arbre C représentant un nombre défini par une constante : il faut remplacer cet arbre par un noeud ayant pour fils gauche C, et pour fils droit un arbre binaire complet de même hauteur que C, ayant la même valeur 0 pour toutes ses feuilles.

-Pour l'arithmétique :

- pour une addition/soustraction : l'extension est facile, il faut faire l'addition/soustraction terme à terme sur les nouvelles données.

- pour la multiplication/division: c'est plus délicat. L'extension ne peut se faire que si l'un des opérateurs est constant. Notons que l'on retrouve ici la même contrainte que pour le traitement des équations pseudo-linéaires dans Prolog III : il faut que l'un des facteurs soit connu pour que la multiplication soit étendue. Alors, l'extension s'effectue en multipliant les éléments nouveaux avec l'opérateur constant dans l'extension définie par cet élément. Cela s'effectue pour chaque sous-arbre correspondant aux racines de l'opérateur constant.

- pour l'extraction de racine carrée : après le calcul initial, la racine carrée est définie comme une constante. Par conséquent, ce calcul doit être étendu comme celui pour une constante.

-Pour les relations :

- pour l'égalité, elle s'étend tout simplement comme l'addition.

Exemple.

Soit initialement l'équation (e) : $-x+(3+5.\sqrt{2}).y=7-11.\sqrt{2}$ exprimée dans l'extension $Q[\sqrt{2}]$. Notons que $Q[\sqrt{2}]$ est normale.

Avec l'écriture complète des constantes $-1+0.\sqrt{2}$, $3+5.\sqrt{2}$, $7-11.\sqrt{2}$, et des variables $x=x_1+x_2.\sqrt{2}$ et $y=y_1+y_2.\sqrt{2}$, l'équation (e) s'écrit :

$$(-1+0.\sqrt{2}).(x_1+x_2.\sqrt{2})+(3+5.\sqrt{2}).(y_1+y_2.\sqrt{2})=7-11.\sqrt{2}$$

c'est à dire, avec deux équations (e₁) : $-x_1+3y_1+10y_2=7$ et (e₂) : $-x_2+5y_1+3y_2=-11$.

Supposons que l'on ajoute la racine de 13 à l'extension $Q[\sqrt{2}]$. L'extension $Q[\sqrt{2},\sqrt{13}]$ est toujours une extension normale. L'équation (e) et les expressions qu'elle contient subissent les extensions suivantes :

pour les constantes :

$$-1+0.\sqrt{2} \text{ devient } -1+0.\sqrt{2}+0.\sqrt{13}+0.\sqrt{2}.\sqrt{13},$$

$$3+5.\sqrt{2}, \text{ devient } 3+5.\sqrt{2}+0.\sqrt{13}+0.\sqrt{2}.\sqrt{13},$$

$$7-11.\sqrt{2} \text{ devient } 7-11.\sqrt{2}+0.\sqrt{13}+0.\sqrt{2}.\sqrt{13},$$

pour les variables :

$$x=x_1+x_2.\sqrt{2} \text{ devient } x=x_1+x_2.\sqrt{2}+x_3.\sqrt{13}+x_4.\sqrt{2}.\sqrt{13},$$

$$y=y_1+y_2.\sqrt{2} \text{ devient } y=y_1+y_2.\sqrt{2}+y_3.\sqrt{13}+y_4.\sqrt{2}.\sqrt{13},$$

pour les multiplications :

Les nouveaux éléments du produit initial $(-1+0.\sqrt{2}).(x_1+x_2.\sqrt{2})$ sont $(-1+0.\sqrt{2}).(x_3+x_4.\sqrt{2})\sqrt{13}$ c'est à dire $-1.x_3.\sqrt{13}$ et $-1.x_4.\sqrt{2}.\sqrt{13}$. L'opérateur constant est ici -1.

Les nouveaux éléments du produit initial $(3+5.\sqrt{2}).(y_1+y_2.\sqrt{2})$ sont $(3+5.\sqrt{2}).(y_3+y_4.\sqrt{2})\sqrt{13}$ c'est à dire $(3.y_3+10.y_4).\sqrt{13}$ et $(5.y_3+3.y_4).\sqrt{2}.\sqrt{13}$. L'opérateur constant est ici $3+5.\sqrt{2}$.

```

complexite_add(0,1) -> ;
complexite_add(_k,2*_add) ->
  complexite_add(_k-1,_add),{_k>0};

complexite_mul(0,0,1) -> ;
complexite_mul(_k,5*_addm+2*_adda,5*_mulm) ->
  complexite_mul(_k-1,_addm,_mulm)
  complexite_add(_k-1,_adda),{_k>0};

complexite_pos(0,0,0,1) -> ;
complexite_pos(_k,_aa+3*( _mm+_pp),3*( _mm+_pp),3*_pp) ->
  complexite_add(_k-1,_aa)
  complexite_mul(_k-1,_am,_mm)
  complexite_pos(_k-1,_ap,_mp,_pp), {_k>0};

```

Figure II.2-g : Calcul de complexité.

pour l'addition :

Les nouveaux éléments de l'addition initiale $-1+x_1+x_2\sqrt{2}+(3+5\sqrt{2})(y_1+y_2\sqrt{2})$ sont $-1.x_3.\sqrt{13}+(3.y_3+10.y_4).\sqrt{13}$ et $-1.x_4.\sqrt{2}.\sqrt{13}+(5.y_3+3.y_4).\sqrt{2}.\sqrt{13}$

pour le symbole d'égalité :

à l'égalité (e) s'ajoutent deux égalités supplémentaires

$$(e_3) : -x_3+3y_3+10y_4=0 \text{ et}$$

$$(e_4) : -1.x_4+5.y_3+3.y_4 = 0.$$

Après extension, l'équation (e) s'exprime avec quatre équations, celles attendues a priori.

e) Complexité de la représentation et de l'arithmétique associée.

Pour être complet, évaluons la complexité des opérations arithmétiques. Formellement, en notant n le nombre d'extensions algébriques quadratiques, nous savons que la représentation d'un nombre constructible de K_n nécessite 2^n rationnels, et pour les opérations :

- l'égalité, l'addition et la soustraction de deux nombres de K_n nécessitent le calcul de 2^n égalités, additions ou soustractions entre rationnels, (ces opérations s'effectuent terme à terme)

- la multiplication de deux nombres de K_n nécessite de l'ordre de 5^n multiplications entre rationnels. (pour rappel la multiplication de (P_x, Q_x) avec (P_y, Q_y) donne (P_z, Q_z) tel que $P_z=P_x*P_y+Q_x*Q_y*a_n$ et $Q_z=P_x*Q_y+Q_x*P_y$.)

La figure II.2-g correspond à des programmes pour calculer précisément la complexité des opérations. Ils illustrent deux idées :

- Ils montrent pédagogiquement ce que signifie une complexité exponentielle.

n	0	1	2	3	4	5
A/a(n)	1	2	4	8	16	32
M/a(n)	0	5	25	125	625	3125
M/m(n)	1	2	14	78	406	2062
P/a(n)	0	7	26	106	464	2134
P/m(n)	0	6	24	102	456	2118
P/p(n)	1	3	9	27	81	243

Tableau II.2-a : complexité des opérations

- Ils sont dérivés directement de programmes de calculs sur les nombres constructibles et constituent des abstractions de ces programmes. Au lieu d'effectuer les opérations sur les nombres constructibles, ces prédicats "comptent" récursivement le nombre d'opérations nécessaires.

Les coûts précis des opérations sont donnés dans le tableau II.2-a dont les lignes comportent :

- A/a(n) : le nombre d'additions entre rationnels pour une addition entre deux nombres constructibles dans une extension avec n racines carrées.

- M/a(n), M/m(n) : le nombre d'additions, de multiplications entre rationnels pour une multiplication entre deux nombres constructibles dans une extension avec n racines carrées.

- P/a(n), P/m(n), P/p(n) : le nombre d'additions, de multiplications et de tests ≥ 0 entre rationnels pour un test ≥ 0 entre deux nombres constructibles dans une extension avec n racines carrées.

L'étude de ce tableau indique que dans l'optique d'une complexité limitée à 1000 fois celle de l'arithmétique rationnelle, cette représentation est limitée à 4 extensions quadratiques.

Il faut ajouter un point encore. La complexité donnée ici est exprimée en nombre d'opérations entre rationnels. Il faudrait considérer en plus la complexité de ces opérations. Pour les arithmétiques en virgule flottante, ces opérations se font en temps et espace constant. Dans le cas présent, l'arithmétique sur les rationnels ne peut utiliser une arithmétique en virgule flottante. Elle nécessite une arithmétique en précision infinie, dont les opérations ne s'effectuent plus en temps et espace constant. Le calcul de la factorielle 1997! à la fin de la partie I donne un exemple de la complexité en espace que peut prendre un calcul "simple". Le temps de calcul est aussi dépendant du nombre de chiffres significatifs intervenant dans les calculs (les premières multiplications entre petits entiers prennent moins de 1 ms, les dernières multiplications entre un petit entier et un grand entier prennent plus de 10 ms).

Conclusion de ce chapitre :

La définition du domaine repose sur deux résultats relativement simples mais originaux :

- l'existence d'un premier algorithme pour décider si un nombre constructible est positif,
et

- l'existence d'un second algorithme de réduction d'une extension algébrique quadratique en extension normale ou, ce qui est équivalent, d'un algorithme de calcul explicite de racine carrée dans une extension quadratique.

Ces deux résultats associés à un choix d'une représentation naïve du domaine permettent de disposer d'une représentation normale qui soit effective.

Notons que, dans le cas proche d'extensions algébriques par des racines cubiques, la recherche du signe d'une expression et le calcul de racines cubiques explicites n'ont pas, à ma connaissance, d'algorithmes simples. Pour le calcul d'une racine cubique dans un corps $\mathbb{Q}[\sqrt[3]{a}]$, en utilisant une méthode similaire à celle employée pour le calcul des racines carrées après triangularisation du système d'équations il reste une équation algébrique de degré 8 en une seule variable dont la résolution n'est pas "simple" a priori. J'entends par "simple" : "qui s'effectue à l'aide de quelques formules", comme dans le cas d'une équation de degré 1, 2, 3 et 4. L'équation générale de degré 5 n'est pas simple comme l'a montré Evariste Galois.

Notons aussi un essai pour échapper à la complexité de la représentation des rationnels en précision infinie. La notion de racine carrée existe pour les corps finis $\mathbb{Z}/n\mathbb{Z}$, cf. [Naudin, Quitté-85]. Il s'agit des résidus quadratiques. Différents résultats intéressants portent sur les résidus quadratiques :

- Dans $\mathbb{Z}/(2n+1)\mathbb{Z}$ il y a n résidus quadratiques. Autrement dit, la moitié des nombres possèdent une racine carrée. Dans $\mathbb{Z}/(4n+3)\mathbb{Z}$, le calcul des racines carrées est simple.

- Pour A, B, \dots donnés, il est possible expérimentalement de choisir n tel que A, B, \dots soient des carrés dans $\mathbb{Z}/n\mathbb{Z}$.

L'ensemble de ces résultats sont prometteurs pour des développements futurs de la représentation normale.

Plus généralement le corps de base \mathbb{Q} peut être remplacé par un corps $K = \mathbb{Q}[a_1, \dots, a_2]$ à condition que le corps K possède deux propriétés : 1) dans K il doit exister un algorithme pour calculer la racine carrée d'un élément de K et 2) dans K il doit y avoir un algorithme pour décider si un élément est positif. Si ces deux conditions sont remplies, notre arithmétique peut s'adapter sans problème aux nombres constructibles sur K . Avec cette extension, il se peut qu'une partie du domaine considéré par la CAO soit accessible. Avec la représentation initiale ce n'est pas le cas car un grand nombre de paramètres pouvant intervenir dans les dessins ne sont pas constructible à la règle et au compas (par exemple l'angle de 1°).

II-3. CLP (Géométrie) : algorithme de satisfaction de systèmes de contraintes.

Introduction :

Les algorithmes utilisés couramment dans les langages CLP pour assurer la satisfiabilité de l'ensemble de contraintes numériques sont de deux types, formels ou approchés.

Reprenons la classification donnée au II.1-b. Du côté des approches formelles se trouvent les algorithmes portant sur le domaine restreint des contraintes linéaires : algorithme d'élimination de Gauss et algorithme du simplexe, Pour le domaine des contraintes non linéaires : les algorithmes fondés sur une décomposition selon des bases de Gröbner, ou à la Collins. Ces derniers algorithmes généraux sont caractérisés par une complexité exponentielle. La correction et la complétude de ces algorithmes vis à vis de la satisfiabilité des systèmes de contraintes considérés sont assurées au prix de la complexité.

Parmi les algorithmes approchés, se trouvent tous les algorithmes de cohérence d'arc, de chemin pour les arithmétiques sur intervalle [Older, Vellino-90]. Se trouvent aussi les algorithmes d'approximations successives, type Newton, [Van Hentenryck, Michel-95]. Avec une position à part, l'algorithme de Gilles Pesant dans [Pesant-95] nous a particulièrement intéressé, puisqu'il s'intéresse aux contraintes quadratiques. C'est un cas particulier où l'approximation n'est pas structurelle mais sémantique. Rappelons que pour l'essentiel, dans ces algorithmes, la complétude et la correction ne sont pas assurées.

Les travaux actuels de la communauté CLP portent en particulier sur la définition de résolveurs coopératifs associant des résolveurs formels, booléens, approchés, et sur des domaines finis.

Pour notre part, dans le cadre d'un paradigme CLP(Géométrie), nous avons choisi un algorithme de type formel¹⁸ pour s'attaquer aux contraintes quadratiques. Le cœur de la résolution est donné par l'algorithme d'élimination de Gauss. Cet algorithme implanté dans le résolveur linéaire de Prolog III, et dans la plupart des langages CLP, est caractérisé par son efficacité et sa faible complexité algorithmique. Aujourd'hui encore, cette faible complexité est un argument en faveur de ce choix. Notre algorithme s'inspire de la résolution naïve de systèmes d'équations pseudo-linéaires tel que ce dernier est implanté dans Prolog III. Notre algorithme prolonge celui de Prolog III pour les contraintes quadratiques tout en gardant la complétude et la correction de l'algorithme initial.

¹⁸Initialement l'idée de cet algorithme se trouve dans [Allen, Bouhineau, Trilling-93]

```

exemple 1 > { x_a=-1, y_a=0, x_b=2, y_b=0, x_c=1, y_c= 2,
(x_a-x_b)*(x_c-x)+(y_a-y_b)*(y_c-y)=0,
(x_b-x_c)*(x_a-x)+(y_b-y_c)*(y_a-y)=0,
(x_c-x_a)*(x_b-x)+(y_c-y_a)*(y_b-y)=0};
résultat : {x_a=-1,y_a=0,x_b=2,y_b=0,x_c=1,y_c=2, x=1,y=1}

exemple 2 > {x_a' = 5, y_a' = -1, x_b' = 1, y_b' = 2, x_c' = -3, y_c' = 6,
x_a + x = 2*x_a', y_a + y = 2*y_a',
x_b + x = 2*x_b', y_b + y = 2*y_b',
x_c + x = 2*x_c', y_c + y = 2*y_c',
(x_a-x_b)*(x_c-x)+(y_a-y_b)*(y_c-y)=0,
(x_b-x_c)*(x_a-x)+(y_b-y_c)*(y_a-y)=0,
(x_c-x_a)*(x_b-x)+(y_c-y_a)*(y_b-y)=0};
résultat : {x_a' = 5, y_a' = -1, x_b' = 1, y_b' = 2, x_c' = -3, y_c' = 6,
x_a=58, x=-48, y_a=52, y=-54, x_b=50, y_b=58, x_c=42, y_c=66}

```

Figure II.3-a: Exemples de résolutions de systèmes de contraintes non-linéaires selon Prolog III.

Le présent chapitre se présente ainsi : la section II.3-a porte sur l'algorithme de résolution de contraintes pseudo-linéaires implanté dans Prolog III. Vient ensuite notre propre algorithme, section II.3-b, qui s'inspire de l'algorithme précédent. Les sections II.3-c, II.3-d et II.3-e portent sur les "à coté" de cet algorithme : définition d'un prédicat nécessaire à notre algorithme, compatibilité de cet algorithme avec la représentation normale des nombres constructibles, complétude de l'algorithme.

Notons l'existence d'algorithmes de construction automatique puissants dans le domaine de la géométrie projective, cf.[Balbiani, Dugat, Fariñas del Cerro, Lopez-94]. De même, dans le domaine de la CAO, des résultats intéressants ont été obtenus par [Kramer-92]. Mais leur application au domaine de la géométrie ne nous semblent pas immédiat.

a) Algorithme naïf de résolution de systèmes d'équations non-linéaires.

L'algorithme de résolution de systèmes pseudo-linéaires donné pour Prolog III dans [Colmerauer-93] constitue la source d'inspiration de notre propre algorithme naïf de résolution de systèmes de contraintes quadratiques. Nous présentons succinctement cet algorithme. Sur quelques exemples, nous montrons les limites de cette approche en général, et pour le domaine de la géométrie.

Le cœur de la résolution de systèmes de contraintes non-linéaires adopté par Prolog III est l'algorithme d'élimination de Gauss. Un système d'équations non-linéaires est résolu de la manière suivante, parmi les équations, celles qui sont linéaires sont mises à part, et sont résolues selon l'algorithme de Gauss. Les résultats de cette résolution, c'est à dire l'instanciation de certaines variables et les relations linéaires qui s'établissent entre variables, sont propagés au reste du système. Si cela a pour conséquence de linéariser de nouvelles équations auparavant non-linéaires, le processus est repris à son point initial; sinon, le système d'équations est dit résolu.

exemple 3 > $\{y_a=4, y_b=-3, x_c=1, y_c=2, x_d=5, y_d=3,$
 $(x_a-x_b)*(x_c-x_d)+(y_a-y_b)*(y_c-y_d)=0,$
 $(x_b-x_c)*(x_a-x_d)+(y_b-y_c)*(y_a-y_d)=0,$
 $(x_c-x_a)*(x_b-x_d)+(y_c-y_a)*(y_b-y_d)=0\};$
résultat : $\{y_a=4, y_b=-3, x_c=1, y_c=2, x_d=5, y_d=3, x_b = x_a + 7/4,$
 $(x_a-13/4)*(-x_a + 1) = -12, (x_a-5)*(x_a + 3/4) = 5\}$

exemple 4 > $\{y_a=4*x_a+5, y_b=3*x_b+1, x_c=1, y_c=2, x_d=5, y_d=3,$
 $(x_a-x_b)*(x_c-x_d)+(y_a-y_b)*(y_c-y_d)=0,$
 $(x_b-x_c)*(x_a-x_d)+(y_b-y_c)*(y_a-y_d)=0,$
 $(x_c-x_a)*(x_b-x_d)+(y_c-y_a)*(y_b-y_d)=1\};$
résultat : $\{x_c=1, y_c=2, x_d=5, y_d=3,$
 $x_a=y_a/4-5/4, y_b=-6y_a+19, x_b=2y_a-6,$
 $(-6y_a+16)*(-y_a+2)=X\$9, (2y_a-11)*(-(1/4)y_a+9/4)=-X\$9+1,$
 $((1/4)y_a-25/4)*(2y_a-7)=-X\$11, (y_a-3)*(-6y_a+17)=X\$11\}$

Figure II.3-a (suite): Exemples de résolutions de systèmes de contraintes non-linéaires selon Prolog III.

L'article [Colmerauer-93] définit une structure étendue des variables réelles pour laquelle cet algorithme est complet et correct. Cette structure considère le signe relationnel "≠" qui remplace (implicitement) et étend le signe égal habituel "=". Une contrainte où apparaît ce signe est satisfaite notamment si, d'un côté ou l'autre du symbole, il se trouve un produit entre variables. Chaque équation non-linéaire est donc satisfaite par structure. Dans cette structure, où les produits de variables entraînent que les contraintes d'égalité et d'inégalités sont satisfaites, l'algorithme de résolution de systèmes d'équations non-linéaires et de vérification de la satisfiabilité de tels systèmes est complet et correct. (exemple de système satisfiable dans cette structure $\{x.y=0, x.x=1, y.y=1\}$)

Exemples.

Sur quelques exemples donnés figure II.3-a, nous présentons les possibilités et limites de cet algorithme. Les exemples proposés sont géométriques. Ils concernent la construction d'un triangle et de son orthocentre à partir de différents éléments. Chaque exemple comporte de plus une équation redondante, qu'il faut vérifier. Dans la construction classique à partir des sommets du triangle, cette équation redondante exprime la propriété de concours des hauteurs.

Voici dans un premier temps l'énoncé des problèmes :

- L'exemple 1 concerne la construction de l'orthocentre en connaissant les sommets du triangle. Les équations utilisées sont directement issues du programme figure II.1-a. La forme de ces équations montre, au passage, que cette construction est équivalente à la construction d'un triangle connaissant deux de ses sommets et son orthocentre (ce qui explique les succès de la section II.1-a).

- L'exemple 2 correspond à la construction d'un triangle à partir de son cercle des neuf points. Plus précisément, cet exemple correspond à la construction d'un triangle à partir de trois points, milieu entre l'orthocentre et un sommet. L'orthocentre lui-même et les sommets ne sont

répéter tant que la taille de $\Sigma = (L+Q)$ décroît :
pour chaque élément s de Σ **faire** :
 si équation_linéaire(s) **alors** (Gauss(s), $\Sigma = \Sigma - s$)
 sinon si équation_second_degré(s) **alors** (Résolution_trinome(s), $\Sigma = \Sigma - s$).

Figure II.3-b : Algorithme de résolution (abstraction)

pas donnés. Ils sont solutions du problème. Les trois points donnés définissent le cercle des neuf points, qui n'est pas construit ici.

- L'exemple 3 correspond à la construction d'un triangle connaissant l'un de ses sommets, son orthocentre, et deux verticales sur lesquelles se situent les deux autres sommets.

- L'exemple 4 généralise l'exemple 3 : il s'agit de trouver une construction d'un triangle connaissant l'un de ses sommets, son orthocentre, et deux droites où se situent les deux autres sommets. Cet exemple comporte de plus une erreur dans l'expression de la contrainte redondante. Cet exemple ne correspond à aucun triangle dans la structure "classique" des réels.

Commentaires.

La résolution des problèmes précédents amène les commentaires suivants :

- Les résultats des exemples 1 et 2 mettent en évidence la puissance de l'algorithme de résolution des équations pseudo-linéaires mis en place dans Prolog III. Les deux exemples sont parfaitement résolus. Ces exemples montrent, par l'exemple, la correction de l'algorithme : la contrainte redondante introduite est vérifiée.

- De plus près, l'exemple 1 est relativement immédiat, puisque les instanciations proposées pour x_a, \dots, y_c linéarisent directement l'ensemble des équations.

- L'exemple 2 est plus délicat. Il faut plusieurs passes pour linéariser et résoudre le problème. Une première passe propage les instanciations des variables et les relations linéaires issues des 6 premières équations. Cette passe a pour résultat de linéariser les trois équations finales. Une deuxième passe prend en compte ces équations. Elle permet la résolution du problème et la vérification de la contrainte redondante.

- Les résultats des exemples 3 et 4 montrent les limites de l'algorithme. Ces deux exemples ne sont pas résolus parfaitement.

- L'exemple 3 ne donne pas la solution. Au terme de la résolution, la solution semble pourtant relativement immédiate : il suffit de résoudre une équation du second degré. De plus, la solution ignore la redondance entre les deux équations restantes.

- L'exemple 4 donne des solutions, alors que le problème n'en a pas. Ceci peut choquer le lecteur. En fait, l'existence de solution exprime que le problème est satisfiable dans la structure élargie avec le "≠" présenté plus haut. Par ailleurs, l'introduction de variables auxiliaires ne facilite pas la lecture.

Dans les exemples 3 et 4, il manque essentiellement un mécanisme de résolution d'équation du second degré. C'est ce que nous apportons dans la suite.

b) Algorithme de résolution de systèmes de contraintes quadratiques.

```

réduction = vrai
propage les instanciations de I dans Q et L
. (cette étape peut modifier dynamiquement Q et L)
Tant que (réduction) faire :
Début Boucle 0
. Tant que ( $L \neq \emptyset$  et réduction) faire :
. Début Boucle 1
. . réduction = faux

. . Pour chaque élément  $\ell$  de L de la forme  $\sum_{i=1}^n m_i * x_i + p = 0$  où p et les
. .  $m_i$  sont des constantes faire :
. . . supprime  $\ell$  de L
. . . remplace  $x_1$  par  $(-p - \sum_{i=2}^n m_i * x_i) / m_1$  dans Q et L
. . . . (c'est l'élimination de Gauss)
. . . . (cette étape peut modifier dynamiquement Q et L)
. . . réduction = vrai
. . . gèle l'introduction de  $x_1$  dans I jusqu'à ce que les  $x_i$  ( $2 \leq i \leq n$ ) soient dans I
. . Fin Pour
. Fin Boucle 1
. Si ( $Q \neq \emptyset$ ) faire :
. Début Bloc 2
. . réduction = faux
. . Pour chaque élément q de Q de la forme  $a * x^2 + b * x + c = 0$  où a, b, c
. . sont des constantes faire :
. . . supprime q de Q
. . . résout l'équation : remplace x par  $\frac{-b \pm \sqrt{b^2 - 4.a.c}}{2.a}$  dans Q et L
. . . . (cette étape peut modifier dynamiquement Q et L)
. . . réduction = vrai
. . . introduit la valeur de x dans I
. . Fin Pour
. Fin Bloc 2
Fin Boucle 0

```

Figure II.3-c : Algorithme de résolution de systèmes de contraintes quadratique.

L'idée de l'algorithme que nous présentons est la même que celle employée en Prolog III. Elle est donnée à la figure II.3-b. Dans une forme plus complète, l'algorithme est donné à la figure II.3-c : les instanciations, équations linéaires et équations du second degré pour une variable sont résolues, aussitôt que possible, dans cet ordre. Nous accordons une priorité à la résolution des équations linéaires. Ainsi, nous retardons le surcoût de gestion des nombres constructibles, ou l'imprécision des calculs avec les réels introduits par les équations du second degré (dans l'hypothèse d'une représentation des nombres constructibles en virgule flottante). Les équations qui n'appartiennent à aucun des types cités précédemment, c'est à dire les équations pseudo-linéaires et les contraintes quadratiques générales, sont gelées tant qu'elles appartiennent à ces classes.

La résolution des équations linéaires est donnée par l'algorithme d'élimination de Gauss. Nous le considérons comme une boîte noire ayant les propriétés suivantes : résolution correcte et complète de systèmes d'équations linéaires sur des rationnels. La résolution de systèmes d'équations linéaires sur les nombres constructibles en découle, II.3-d. De plus, nous admettons que l'algorithme de Gauss permet de vérifier la satisfiabilité de systèmes d'équations linéaires redondantes.

Algorithme

Pour décrire notre algorithme, nous introduisons la notation suivante : un système de contraintes est un triplet $\langle Q, L, I \rangle$ d'ensemble d'équations. Q est l'ensemble des équations quadratiques. L est l'ensemble des équations pseudo-linéaires. I est l'ensemble des instanciations. L'algorithme est constitué de deux boucles, la première concerne les équations linéaires, la seconde concerne les équations du second degré en une variable. Il est présenté en détail figure II.3-c.

Plus précisément, l'élimination de Gauss et l'évolution dynamique de Q et L sont effectuées en deux étapes qu'il faut décrire :

- L'élimination de Gauss qui est exprimée explicitement dans l'algorithme de la figure II.3-c, est effectuée implicitement lors de l'appel au solveur de contraintes linéaires de Prolog III. Celui-ci opère effectivement cette réduction à l'intérieur de ce que l'on a appelé une boîte noire. Idem pour le remplacement d'une variable par sa valeur dans la résolution d'une équation du second degré : elle est réalisée en appelant le solveur de contraintes.

- L'évolution dynamique de Q et L se fait via un retour d'informations de la boîte noire constituée par le solveur de contraintes :

- Si une instanciation est détectée dans le solveur, soit qu'elle soit produite par notre algorithme, soit qu'elle provienne de la résolution d'un système d'équations linéaires, celle-ci est propagée dans les éléments de Q et L. Cette propagation prend effet au moment où les contraintes sont analysées.

- De même, le remplacement d'une variable par une expression linéaire lors de l'élimination de Gauss prend effet au moment où les contraintes sont analysées. Ce remplacement est effectué via un prédicat "Relation_Lineaire". Ce prédicat, étudié à la section II.3-c, interroge le solveur boîte noire pour savoir si deux variables X, Y sont telles que $X = a.Y + b$, avec a et b constants. Dans ce cas, X est effectivement remplacé par Y dans la contrainte analysée.

Nous donnons des exemples d'application de cet algorithme en géométrie dans le chapitre II.4 suivant.

Complexité, correction et complétude.

Avant de poursuivre l'étude de quelques points précis de l'algorithme aux sections suivantes :

- recherche de relations linéaires II.3-c, pour le prédicat "Relation_Lineaire" qui interroge la boîte noire,

- compatibilité avec la représentation normale des nombres constructibles II.3-d,

- complétude géométrique de l'algorithme,
- nous finissons cette section avec les points théoriques :
- complexité,
 - correction et
 - complétude algébrique de l'algorithme.

En comparaison avec la complexité exponentielle des algorithmes généraux pour la résolution des systèmes algébriques quelconques, notre algorithme possède une complexité polynomiale faible de l'ordre de $l^3 \cdot q^3$ où $l=|L|$ et $q=|Q|$. La complexité en n^3 correspond à la complexité de la résolution de Gauss.

La correction de cet algorithme est a priori évidente. Si l'algorithme se termine avec une solution et $Q=L=\emptyset$, alors la solution est juste. Si l'algorithme déclare que le système est insatisfiable, c'est correct. Cependant, il faut signaler : d'une part la correction de cet algorithme est soumise à la correction de l'élimination de Gauss sur les nombres constructibles donnés selon notre représentation normale. Ce point est étudié section II.3-d. D'autre part, si l'algorithme se termine avec $Q \neq \emptyset$ ou $L \neq \emptyset$, les équations restant dans Q et L ne sont pas prises en compte. Dans la structure élargie de [Colmerauer-93], la correction est assurée.

La complétude de cet algorithme est beaucoup plus délicate à décrire. La classe des problèmes pour lesquels cet algorithme est décrit est celle des systèmes de contraintes quadratiques. Pour la sous-classe des équations linéaires et du second degré en une seule variable, notre algorithme est complet. Cependant, la complétude de notre algorithme est soumise à la complétude de la résolution de Gauss sur les nombres constructibles.

Avec le mécanisme d'attente de modification des équations pseudo-linéaires et quadratiques, la classe des problèmes où l'algorithme est complet est, en fait, plus large. Nous nous trouvons devant la même difficulté que [Colmerauer-93] pour définir la classe de problèmes pour lesquels cet algorithme est complet. En quelque sorte, c'est la classe des problèmes qui deviennent mixtes linéaire/2nd-degré après résolution locale et propagation des solutions locales.

Dans une structure élargie du même type que celle présentée dans [Colmerauer-93], la complétude est assurée pour la classe des contraintes non linéaire. Mais cela a-t-il un intérêt ? Nous essaierons dans la section II.3-e de définir une notion de complétude en termes géométriques.

c) Recherche de relations linéaires.

L'un des points laissés en suspens au II.3-b concerne la recherche de relations linéaires entre deux variables selon le prédicat `Relation_Lineaire` qui interroge la boîte noire. Sur le plan théorique, il s'agit de trouver la projection du système de contraintes linéaires sur deux variables. Cette information est connue du résolveur de Prolog III, de manière interne. Mais comme nous utilisons le résolveur linéaire comme une boîte noire et que cette information n'est pas donnée, il nous faut la trouver.

```

%Relation_Lineaire(->num,->num,?num,?num) s'efface si %4=%3*%1+%2
Relation_Lineaire(_v-_b,_b,_x,_y) ->
  trouve_val(_x,0,_y,_b)
  trouve_val(_x,1,_y,_v);

trouve_val(_x,_x,_y,_) ->
  known(_y)
  assign(trouve,1')
  assign(valeur,_y)
  fail;
trouve_val(_x,_,_y,_v) ->
  val(trouve,1')
  assign(trouve,0')
  val(valeur,_v);

```

Figure II.3-d : Prédicat Relation_Lineaire.

D'une manière générale, il peut sembler intéressant que des prédicats pour informer sur l'état du système de contraintes soient ainsi fournis dans les langages CLP. C'est le paradigme ask/tell que nous observons parfois dans la communauté Prolog et qui, nous l'espérons, verra des réalisations pratiques.

Ici, il semble intéressant d'obtenir un prédicat Relation_Lineaire(v,k) où v est un ensemble de variables (V_i), et k un ensemble de coefficient (m_i,p), tel que Relation_Lineaire(v,k) soit réalisé si $\sum m_i * V_i = p$. Si k est connu, alors ce prédicat ajoute la contrainte linéaire, sinon une relation linéaire entre les éléments de v donnés est recherchée, qui donne la liste des coefficients k (c'est le point qui nous intéresse).

En l'absence du prédicat Relation_Lineaire(v,k), nous avons écrit un prédicat réalisant la recherche de relations linéaires entre deux variables seulement. Il repose sur les deux principes suivant :

- les seules relations algébriques qui peuvent lier x et y via le résolveur d'équations linéaires, sont des relations linéaires,
- si une relation linéaire existe entre deux variable x et y, l'instanciation de x provoque via le résolveur d'équations linéaires l'instanciation de y. Par suite :
 - si l'instanciation $x=0$ entraîne $y=p$ et
 - si l'instanciation $x=1$ entraîne $y=k$ alors
 - x et y sont liés par la relation linéaire suivante : $y=(k-p)x+p$.

Concrètement le prédicat est écrit avec des effets de bords peu glorieux, cf. figure II.3-d.

De même, que pour l'établissement de la correction et de la complétude de l'algorithme naïf de résolution de systèmes d'équations mixtes, ce prédicat assume la compatibilité de l'algorithme d'élimination de Gauss avec la représentation normale des nombres constructibles.

d) Algorithme d'élimination de Gauss et nombres constructibles.

Les sections précédentes dont II.2-c, II.3-b et II.3-c ont révélé la nécessité d'une preuve de la compatibilité entre l'algorithme d'élimination de Gauss, et la représentation normale des nombres constructibles. L'algorithme d'élimination de Gauss mis en place dans Prolog III porte sur des systèmes de contraintes linéaires sur les rationnels. La représentation normale d'un système d'équations linéaires sur des nombres constructibles induit un certain nombre d'équations linéaires sur les rationnels. Comment ces deux points s'enchaînent-ils pour donner un algorithme de résolution complet et correct ?

Un premier point est de montrer comment s'établit la résolution d'un système d'équations linéaires sur les nombres constructibles en faisant appel au résolveur linéaire de Gauss. Nous le faisons d'abord sur un exemple. Une démonstration formelle vient s'ajouter ensuite.

Un second point est de montrer que la correction et la complétude du résolveur linéaire sur les rationnels sont étendues aux nombres constructibles. Nous le faisons en nous appuyant sur la correction et la complétude de l'algorithme général de Gauss sur les réels. L'algorithme général de Gauss sur les réels n'est pas implanté dans Prolog III. Pour information, la gestion des réels se fait dans Prolog III avec des nombres en virgule flottante. Un algorithme d'élimination type Gauss sur les nombres flottants est utilisé. Nous ne l'utilisons pas car il donne des résultats approchés.

Résolution d'un système linéaire sur les nombres constructibles.

Considérons l'exemple suivant pour montrer comment l'écriture de contraintes linéaires sur des nombres constructibles se ramène à la résolution de systèmes linéaires sur les rationnels. Supposons donc que l'on ait à résoudre le système :

$$\begin{cases} 6\sqrt{2}.x - (4 - \sqrt{2}).y + z = \sqrt{6\sqrt{2} - 4} \\ x - \sqrt{2}.y + (25 - \sqrt{2} - \sqrt{3\sqrt{2} - 2}).z = 2\sqrt{2} - \sqrt{3\sqrt{2} + \sqrt{24\sqrt{2} - 16}} \\ -5.x + 4.y - \sqrt{8}.z = 1 + \sqrt{24\sqrt{2} - 16} \end{cases}$$

La liste des racines carrées apparaissant dans ce système est :

$\sqrt{2}$, $\sqrt{6\sqrt{2}-4}$, $\sqrt{3\sqrt{2}-2}$, $\sqrt{3\sqrt{2}+\sqrt{24\sqrt{2}-16}}$, $\sqrt{8}$ et $\sqrt{24\sqrt{2}-16}$. Après réductions, il apparaît que ces nombres constructibles sont définis dans l'extension algébrique $Q[\sqrt{2},\sqrt{3\sqrt{2}-2}]$ et que :

$$\begin{aligned} \sqrt{6\sqrt{2}-4} &= \sqrt{2} * \sqrt{3\sqrt{2}-2}, \\ \sqrt{3\sqrt{2}+\sqrt{24\sqrt{2}-16}} &= \sqrt{2} + \sqrt{3\sqrt{2}-2}, \\ -\sqrt{8} &= -2\sqrt{2}, \\ \sqrt{24\sqrt{2}-16} &= 2 * \sqrt{2} * \sqrt{3\sqrt{2}-2}. \end{aligned}$$

Par conséquent, le système d'équations s'écrit avec la structure d'arbre :

$$\begin{cases} ((0,6)(0,0))^2x + ((-4,1)(0,0))^2y + ((1,0)(0,0))^2z = ((0,0)(0,1)) \\ ((1,0)(0,0))^2x + ((0,-2)(0,0))^2y + ((25,-2)(1,0))^2z = ((0,0)(0,1)) \\ ((-5,0)(0,0))^2x + ((4,0)(0,0))^2y + ((0,-2)(0,0))^2z = ((1,0)(0,2)) \end{cases}$$

```

mise en place de la première équation :
> float_anc(2,c2,1)      % c2=2
float_anc(3,c3,1)      % c3=3
float_anc(6,c6,1)      % c6=6
float_anc(4,c4,1)      % c4=4
sqrt_anc(c2,r2,1)      % r2=sqrt(2) 1ere racine
mult_anc(r2,c3,e0,1)   % e0=3*sqrt(2)
plus_anc(e1,c2,e0,1)   % 3*sqrt(2) - 2
sqrt_anc(e1,r',1)     % sqrt(3*sqrt(2) - 2) 2nde racine
mult_anc(r2,c6,e2,1)   % 6*sqrt(2)
plus_anc(e3,c4,e2,1)   % 6*sqrt(2) - 4
sqrt_anc(e3,e4,1)     % sqrt(6*sqrt(2) - 4) partie droite
mult_anc(x,e2,e5,1)    % x*6*sqrt(2)
plus_anc(e6,c4,r2,1)  % sqrt(2) - 4
mult_anc(y,e6,e7,1)   % y*(sqrt(2) -4)
plus_anc(z,e5,e8,1)   % x*6*sqrt(2)+z
plus_anc(e8,e7,e9,1)   % x*6*sqrt(2)+y*(sqrt(2) -4)+z
;                       % partie gauche de l'équation
résultat de Prolog (après renommage et quelques simplifications):
{ ..., e4 = <0,0,0,1>, x = <x1,x2,x3,x4>,
  y = <y1,y2,y3,y4>, z = <z1,z2,z3,z4>,
  e9 = <z1+2y2-4y1+12x2, z2-4y2+y1+6x1,
        z3+2y4-4y3+12x4, z4-4y4+y3+6x3>, ... }

```

Figure II.3-e : Contrainte linéaire sur des nombres constructibles.

Les variables s'expriment elles-mêmes sous forme d'arbres $x=((x_0,x_1),(x_2,x_3))$, $y=((y_0,y_1),(y_2,y_3))$, et $z=((z_0,z_1),(z_2,z_3))$. Le système final correspond au système de contraintes effectivement transmis au résolveur de Prolog III, qui est le suivant. Il correspond aux équations induites par la figure II.3-e:

$$\left\{ \begin{array}{l} z_1+2y_2-4y_1+12x_2=0 \\ z_2-4y_2+y_1+6x_1=0 \\ z_3+2y_4-4y_3+12x_4=0 \\ z_4-4y_4+y_3+6x_3=1 \\ -6z_4+2z_3-2z_2+25z_1-2y_2+x_1=0 \\ 2z_4-3z_3+25z_2-z_1-y_1+x_2=1 \\ -2z_4+25z_3-z_1-2y_4+x_3=0 \\ 25z_4-z_3-z_2-y_3+x_4=-1 \\ -4z_2+4y_1-5x_1=1 \\ -2z_1+4y_2-5x_2=0 \\ -4z_4+4y_3-5x_3=0 \\ -2z_3+4y_4-5x_4=2 \end{array} \right.$$

Ce système, comme nous le voyons facilement, est linéaire. Il comporte autant d'équations que de variables. Son déterminant est sûrement non nul. Le calcul du déterminant 12×12 n'est pas immédiat, il vaut : 395647413048. Il est bien non nul ! Donc le système linéaire possède une solution unique.

La solution de ce système sur les rationnels avec Prolog III est :

$$\begin{aligned} x &= ((387746087/23995517493 , 680772792/7998505831) , \\ &\quad (11432448752/23995517493 , 818805942/7998505831)) \\ y &= ((7622469556/23995517493 , 2771963413/23995517493) , \\ &\quad (29649065933/47991034986 , 14989420003/23995517493)) \\ z &= ((438130886/23995517493 , 379635858/7998505831) , \\ &\quad (157722052/23995517493 , 355981351/15997011662)) \end{aligned}$$

La solution du système sur les nombres constructibles est donc :

$$\begin{aligned} x &= 387746087/23995517493 + 680772792/7998505831 \cdot \sqrt{2} + \\ &\quad 11432448752/23995517493 \cdot \sqrt{3\sqrt{2}-2} + \\ &\quad 818805942/7998505831 \cdot \sqrt{2} \cdot \sqrt{3\sqrt{2}-2} \\ y &= 7622469556/23995517493 + 2771963413/23995517493 \cdot \sqrt{2} + \\ &\quad 29649065933/47991034986 \cdot \sqrt{3\sqrt{2}-2} +, \\ &\quad 14989420003/23995517493 \cdot \sqrt{2} \cdot \sqrt{3\sqrt{2}-2} \\ z &= 438130886/23995517493 + 379635858/7998505831 \cdot \sqrt{2} + \\ &\quad 157722052/23995517493 \cdot \sqrt{3\sqrt{2}-2} + \\ &\quad 355981351/15997011662 \cdot \sqrt{2} \cdot \sqrt{3\sqrt{2}-2} \end{aligned}$$

Correction et complétude.

Sur le plan formel, considérons un système de n équations linéaires sur n variables, avec des coefficients qui sont des nombres constructibles d'une extension K_i normale de degré 2^i , ayant i racines carrées distinctes.

L'algorithme général de Gauss pour les réels nous assure que si une solution existe, elle est unique. La correction (évidente) de notre représentation assure que si une solution est trouvée, elle satisfait bien les équations. Donc, d'après ce qui précède, c'est la solution.

Il reste à montrer qu'un tel système a bien une solution. Le décompte du nombre de variables rationnelles et des équations linéaires transmises au résolveur linéaire s'effectue comme suit :

- chaque variable constructible correspond à 2^i variables rationnelles,
- chaque équation linéaire sur des variables constructibles correspond à 2^i équations sur des variables rationnelles.

Le point délicat est de démontrer que ces équations sont linéaires. La linéarité des équations sur les nombres constructibles est transmise aux rationnels. En effet, comme nous l'avons vu en II.2-c, la multiplication d'une variable constructible par une constante donne une variable dont les coordonnées sont fonctions linéaires.

Donc, au niveau du système sur les nombres rationnels, il y a autant de variables que d'équations. Ce système rationnel est soit incohérent, soit admet une infinité de solutions, soit a une seule solution.

- Le cas où le système rationnel aurait une infinité de solution est impossible car la représentation normale ne donne qu'une seule représentation à la solution réel. C'est ici aussi qu'il est important d'utiliser une représentation normale !

- Le cas où le système rationnel serait incohérent est exclu, puisque l'on sait que le système réel possède une solution.

Donc, le système rationnel a une solution et une seule. La correction et la complétude de la résolution de Gauss sur les rationnels sont transmises à la résolution sur les nombres constructibles représentés par une forme normale.

Note : Il y a une démonstration directe par double récurrence selon le nombre d'équations et le nombre d'extensions en effectuant le calcul du déterminant du système rationnel, mais elle doit être très compliquée. Pour le cas de base une équation, une extension, elle fait apparaître clairement la nécessité de la normalité de l'extension, et pour la suite, cela devient vite inextricable à la main.

Rappelons en ce qui concerne la vérification des contraintes redondantes, que la forme normale de l'extension quadratique nous assure la complétude et la correction de cette vérification.

e) Complétude géométrique.

Il a déjà été dit au II.3-b qu'il y a complétude et correction de notre algorithme pour des systèmes d'équations linéaires et du second degré en une variable. L'objectif de ce travail est la définition d'un paradigme de programmation adapté à la géométrie. La complétude et la correction de notre algorithme de résolution doit s'exprimer en termes géométriques. Nous avons vu en I-4 que nous ne pouvions, pour des raisons de complexité algorithmique, prendre en compte toute spécification géométrique.

Nous pouvons imaginer deux classes de problèmes pour lesquelles notre algorithme soit correct et complet :

- la première classe de problème nous semble minimale, c'est l'ensemble de spécifications "constructives" : celles qui contiennent la spécification complète d'une construction à la règle et au compas.

- La seconde classe, celle que nous espérons atteindre, concerne l'ensemble de spécifications qui permettent la construction à la règle et au compas d'une figure respectant la spécification sans l'ajout d'éléments supplémentaires à la spécification.

Complétude pour les spécifications constructives

En ce qui concerne la première classe de problèmes, celle des spécifications constructives, notre algorithme est correct mais incomplet ! En effet, toutes les constructions proposées

vérifieront bien la spécification puisque les contraintes redondantes sont vérifiées correctement. Mais notre algorithme est incomplet si l'on observe les trois constructions élémentaires :

- intersection droite-droite : $\begin{cases} y=m.x+p \\ y=m'.x+p' \end{cases}$
- intersection droite-cercle : $\begin{cases} y=m.x+p \\ (x-x_0)^2+(y-y_0)^2=K \end{cases}$
- intersection droite-cercle : $\begin{cases} (x-x_0)^2+(y-y_0)^2=K \\ (x-x_0')^2+(y-y_0')^2=K' \end{cases}$

($m, m', p, p', x_0, y_0, K, x_0', y_0', K'$ sont des constantes connues; x, y , les variables inconnues que nous cherchons).

De ces trois systèmes d'équations, seuls les deux premiers sont résolus par notre algorithme :

- Le premier système est résolu directement par l'algorithme de Gauss puisque les deux équations sont linéaires.

- Le second système est résolu en deux temps. Le premier temps consiste à introduire la première équation dans le résolveur linéaire. Ensuite, cette relation linéaire entre x et y est propagée dans le système restant. Dans un second temps, la deuxième équation devient du second degré en y . Elle est donc résolue. Ce qui achève la résolution du système.

- En ce qui concerne le troisième système, les deux équations ne sont ni linéaires, ni du second degré en une variable. Le résolveur naïf n'aboutit pas.

Pour effectuer la résolution du troisième système "à la main", nous notons que la différence terme à terme des deux équations donne l'équation supplémentaire :

$$(2x-x_0-x_0')(x_0'-x_0)+(2y-y_0-y_0')(y_0'-y_0)=K-K'$$

Cette équation est linéaire. En fait, nous retrouvons l'équation de la corde commune aux deux cercles. Avec cette équation supplémentaire, nous retrouvons le cas du second système d'équations, que nous pouvons résoudre.

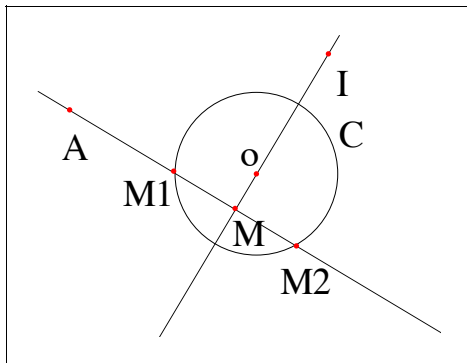
L'ajout des cordes communes est convenable d'un point de vue pédagogique car cela a du sens en géométrie. Et du point de vue algébrique, cela a du sens aussi. Ici nous pouvons observer l'adéquation simple (ce qui est rare) entre la géométrie et l'algèbre et de plus sur une transformation immédiate.

En conséquence, nous ajoutons à notre algorithme un pré-traitement géométrique qui ajoute à la spécification toutes les équations de cordes communes. Et par suite, nous obtenons une complétude pour la première classe de problèmes envisagée :

Proposition II.3-a : Notre algorithme est complet pour la classe des spécifications constructives.

[[preuve]] Si une construction existe, elle s'effectue à chaque étape selon l'une des trois constructions élémentaires (intersections). Chacune de ces constructions est résolue par l'algorithme.

[[fin de la preuve]]



Soient A, I, O trois points du plan et C un cercle de centre O.

Chercher M, M1, M2 trois points tels que :

A, M1, M2 soient alignés

M1 et M2 soient sur C

M milieu de M1, M2

$AM \perp MI$

Figure II.3-f : Spécification non constructive.

Complétude pour les constructions sans ajouts.

En ce qui concerne la deuxième classe de problèmes, c'est à dire les figures qui peuvent être construites à la règle et au compas à partir d'une spécification sans ajout d'éléments supplémentaires, il est plus difficile de voir si notre algorithme est complet. Et s'il ne l'est pas, il est délicat de voir comment le rendre complet. Naturellement, pour les mêmes raisons que précédemment, il est correct.

Un contre-exemple montre que l'algorithme amélioré n'est pas complet pour la seconde classe de problème. La spécification de la figure II.3-f n'est pas résolue par notre algorithme, même avec l'ajout des équations des cordes communes (qui sont ici des médiatrices).

Nous ne développerons pas ici les raisons de l'impossibilité de la résolution naïve du système. Cependant, nous pouvons observer sur cet exemple que l'ajout de la spécification supplémentaire $O \in (MI)$ permet la construction de la figure. Cette spécification est redondante par rapport au système algébrique initial. Elle permet la résolution algébrique du système d'équations.

L'ajout effectué dans l'exemple précédent est significatif. Il donne l'idée de deux algorithmes, l'un combinatoire et complet, l'autre heuristique et incomplet, mais meilleur que le présent pour atteindre la complétude que nous cherchons :

- Le premier algorithme consiste, à l'aide d'une recherche combinatoire, à essayer d'ajouter toutes les propriétés géométriques à la figure pour produire sa construction. Si au terme de cette recherche la figure n'a pas été construite, c'est qu'elle n'appartient pas à la classe considérée.

- L'autre algorithme consiste à chercher une construction, même partielle, de la figure à partir d'un autre ensemble de points de bases. A partir de cette construction partielle, des propriétés supplémentaires peuvent être extrapolées. Sous condition qu'une figure particulière ait été trouvée, l'algorithme est complet. Si une figure approchée seulement a été trouvée, cela peut aider à une recherche combinatoire limitée aux propriétés plausibles.

Pour la suite, nous nous contenterons de l'algorithme initial modifié par l'ajout des cordes communes de la section précédente. Les deux idées d'algorithmes présentées ici ne sont pas employées. La complétude pour les constructions sans ajout est accessible mais n'est pas atteinte.

Conclusion :

L'approche naïve proposée dans ce chapitre répond aux exigences de nos objectifs initiaux :

- résolveur de type formel, correct,
- compatibilité avec la représentation normale,
- explication de la complétude en termes géométriques.

Elle comporte de nombreuses perspectives d'améliorations :

- améliorations algébriques : propagation des relations quadratique, résolution d'équation de degré supérieur, renommage des parties non-linéaire cf. [Colmerauer-93] pour le dernier point,
- améliorations géométriques : ajout de plus de redondances automatiquement cf. [Schreck-92], ou mise en place des algorithmes de la dernière section.

Il reste à tester cette approche sur des exemples usuels de la géométrie. C'est l'objet du chapitre II-4.

II-4. Résultats.

Introduction :

Pour clore cette partie sur la définition d'un environnement de programmation CLP(Géométrie) adapté à la géométrie, nous présentons quelques résolutions de systèmes de contraintes correspondant à des figures géométriques courantes. Parmi ces résolutions, certaines correspondent à des problèmes de construction réputés difficiles. D'autres correspondent à des variantes de situations classiques, par exemple en instanciant les éléments qui sont habituellement construits, et non données du problème. Ces constructions "à l'envers" constituent parfois des problèmes de construction géométrique compliqués, même si elles correspondent à des situations géométriques simples et courantes.

L'ensemble de ces constructions est étudié en vue de la mise en place d'un environnement informatisé d'apprentissage humain de la géométrie utilisant une approche déclarative. Le niveau de complexité des figures est choisi en correspondance.

Ce chapitre commence section II.4-a par la définition rapide du cadre où s'effectuent les résolutions formelles : définition d'un langage géométrique, définition des contraintes géométriques correspondantes, définition de la grille où les points de départ sont pris. La section II.4-b reprend les résultats de complétude du chapitre II-3. Aux sections II.4-c et II.4-d, les figures géométriques sont réparties en deux classes selon que l'algorithme est suffisant ou non à la construction. Enfin, en II.4-e, un récapitulatif est entrepris pour les constructions possibles dans un triangle classique comportant hauteurs, médianes, médiatrices.

a) Représentation des figures géométriques.

Poursuivant l'approche adoptée en première partie, nous proposons un langage de définition des figures géométriques qui s'appuie sur des constructions géométriques à la règle et au compas. Le chapitre II.3, plus formel, s'appuie sur trois constructions seulement, intersections droite-droite, droite-cercle, cercle-cercle. Ce chapitre, plus général, propose un plus grand nombre de constructions. Les contraintes linéaires sont privilégiées, même si celles-ci ne correspondent pas à des constructions immédiates à la règle et au compas. Par exemple, cela mène à favoriser la définition du milieu de deux points par rapport à la construction d'une bissectrice (ce qui rejoint les conclusions de la partie I)

Langage géométrique.

Le langage géométrique utilisé comporte les éléments suivants :

- les objets géométriques sont :
 - + les points : point(a),
 - + les droites : (d) ou droite(d) ou (a,b) si a et b sont des points,
 - + les cercles : cercle c(o) où "c" est le nom du cercle, "o" son centre;
- les relations géométriques entre objets sont :
 - + l'appartenance d'un point à une droite ou un cercle : $\dots \in \dots$,
 - + le parallélisme : $\dots // \dots$,
 - + la perpendicularité : $\dots \perp \dots$,
 - + le milieu : a milieu de [b,c];
- deux relations sont ajoutées pour distinguer les différents cas de figure lors de constructions d'intersection avec un cercle
 - + a et b sont du même côté de (d),
 - + a et b sont de chaque côté de (d).

Une droite définie en utilisant la notation (a,b) est une droite ayant un nouveau nom (un nom interne comme #ab) sur laquelle portent deux contraintes $a \in \#ab$, et $b \in \#ab$. Les contraintes d'appartenance, de parallélisme, de perpendicularité portant sur une droite définie en utilisant cette notation (a,b) portent en fait sur la droite #ab.

Comme nous l'avons dit en II.3-e, si un point M appartient à deux cercles C1(O1) et C2(M2), nous effectuons un enrichissement de la spécification pour être complet. La corde commune des deux cercles C1(O1) et C2(M2) est ajoutée avec un nouveau nom (un nom interne comme #C1C2). La contrainte linéaire qui définit #C1C2 en fonction de C1(O1) et C2(M2) est ajoutée. Enfin, la contrainte $M \in \#C1C2$ est ajoutée à la spécification de la figure.

Représentation algébrique.

En reprenant les conventions données en I.2-c pour la représentation des objets nous avons :

- les objets représentés comme suit :
 - + un point P par deux coordonnées (P_x, P_y) ,
 - + une droite D est représentée par les deux coefficients D_m, D_p qui interviennent dans l'équation $y = D_m \cdot x + D_p$ (la section suivante sur le choix de la grille résout le problème des droites verticales),
 - + un cercle C(o) est représenté par les coordonnées de son centre o, et un coefficient C_k (carré du rayon) qui intervient dans l'équation $(x - O_x)^2 + (y - O_y)^2 = C_k$;
- les relations géométriques entre objets sont :
 - + $M \in D : M_y = D_m \cdot M_x + D_p$,
 - + $D // D' : D_m = D'_m$,
 - + $D \perp D' : D_m \cdot D'_m = -1$,
 - + A milieu de [B,C] : $2 \cdot A_x = B_x + C_x, 2 \cdot A_y = B_y + C_y$.

Le choix d'un représentant algébrique unique pour les objets géométriques est important. Il justifie le choix de la représentation unique par l'équation $y=m.x+p$ pour les droites, même si les verticales ne sont pas représentables. Les équations plus générales, mais non uniques de type $a.x+b.y+c=0$, posent le problème de cette unicité. Ajouter la contrainte $a^2+b^2=1$ résout ce problème, mais introduit une contrainte non linéaire dont le coût est trop élevé. Ajouter la contrainte ($a=1$ ou $b=1$) résout également le problème, mais introduit une prévisible explosion combinatoire dans la recherche des représentants des droites. Ces deux solutions n'ont pas été adoptées.

Choix de la grille.

Pour rappel, la grille correspond à l'ensemble des coordonnées que peut prendre un point donné par l'utilisateur. La grille donnée par les coordonnées des pixels à l'écran d'un ordinateur n'est pas utilisable avec notre représentation. Elle accorde une place prépondérante aux droites verticales qui ne sont pas représentables.

La grille que nous choisissons ne donne pas cette place prépondérante aux verticales. C'est la grille classique des écrans d'ordinateurs (1000 points par 500 points (en gros)) mais les coordonnées de ces points sont $Rotation_{\theta}.(x,y)$ pour le pixel (x,y) . La matrice $Rotation_{\theta}$ est une matrice rationnelle approchée de la matrice de rotation d'angle θ , et de centre de rotation le centre de l'écran. L'angle θ est choisi très faible, de telle sorte que l'horizontale de l'écran qui correspond à un presque horizontale des coordonnées, donne à l'écran une droite horizontale au pixel près.

Cette grille n'interdit pas les horizontales ou les verticales dans l'absolu. Cependant, elle assure que deux points pris au hasard à l'écran ne sont pas sur une même horizontale ni même sur une verticale. Par conséquent, l'obtention d'une vraie horizontale nécessite une construction. Pour celui qui ne connaît pas les caractéristiques précises définissant $Rotation_{\theta}$, cette construction n'est possible. Par suite, les horizontales ne peuvent apparaître que suite à un effort important et délibéré. Le choix des équations $y=m.x+p$ se justifie donc.

Dans le même ordre d'idées, la grille permet d'éviter la plupart des alignements fortuits entre points quelconques. De même, les points cocycliques sur la grille sont très rares. L'irrégularité de la grille raréfie les configurations un peu particulières entre points (milieu, perpendicularité, etc...).

Si la grille le pouvait, elle interdirait que deux points choisis au hasard (sur le même pixel) soit identiques. L'idée sous-jacente est que ces propriétés, pour être réalisées doivent être spécifiées, ou dépendre des spécifications de la figure. Sur une feuille blanche, ces propriétés ne peuvent apparaître que si d'une certaine manière elles ont été spécifiées.

Spécification d'une figure.

Nous considérons une figure comme un ensemble constitué de :

- une spécification (comportant un ensemble d'objets et des contraintes géométriques liant ces objets), et
- un ensemble de points de base (sous-ensemble des points de la spécification qui recevront une instantiation).

Exemple 1 : Figure représentant la construction de l'orthocentre d'un triangle donné :

- spécification : point(a), point(b), point(c), point(h), (a,h) \perp (b,c), (a,b) \perp (c,h), (a,c) \perp (b,h);
- points de base : point(a), point(b), point(c).

Exemple 2 : Figure représentant la construction d'un triangle à partir d'un coté et de son orthocentre :

- spécification : point(a), point(b), point(c), point(h), (a,h) \perp (b,c), (a,b) \perp (c,h), (a,c) \perp (b,h);
- points de base : point(a), point(b), point(h).

Les deux exemples précédents montrent que deux figures géométriques différentes peuvent coïncider sur la spécification et se distinguer seulement sur l'ensemble des points de base.

Il y a analogie dans cette présentation d'une figure géométrique avec la notion de programme Prolog. En effet, la notion de programmes en Prolog ne coïncide pas exactement avec ce que l'on a l'habitude de définir comme programme pour d'autres langages de programmation. Un programme Prolog est un ensemble constitué de :

- un paquet de règles définissant l'environnement où sont interprétées des requêtes,
- une forme de requête définissant ce qui est "connu" dans l'appel des requêtes et ce qui est variable.

Deux programmes Prolog distincts peuvent coïncider sur l'ensemble des paquets de règles qui constituent l'environnement et se différencier seulement sur la forme de la requête. Par exemple, considérons un paquet de règles définissant usuellement un tri. Le programme de tri comporte ce paquet et la forme de requête suivante : tri(liste!,->liste) où %1 est une liste connue, et %2 une variable qui sera instanciée par le programme. Avec ce même paquet de règles, nous pouvons donner un programme de permutation. Il comporte ce même paquet de règles et la forme de requête suivante : tri(->liste,list!) où %1 une variable qui sera instanciée par le programme et %2 est une liste connue. En effet, si %2 est triée alors l'ensemble des %1 qui satisfassent tri(%1,%2) recouvre l'ensemble des permutations de %2.

b) Figures géométriques où l'algorithme est complet.

Le chapitre II.3 montre la complétude de l'algorithme de résolution pour deux classes de problèmes : d'une part la classe des problèmes linéaires et des équations du second degré en une variable, d'autre part la classe des spécifications constructives. Quelles figures géométriques correspondent à ces classes ?

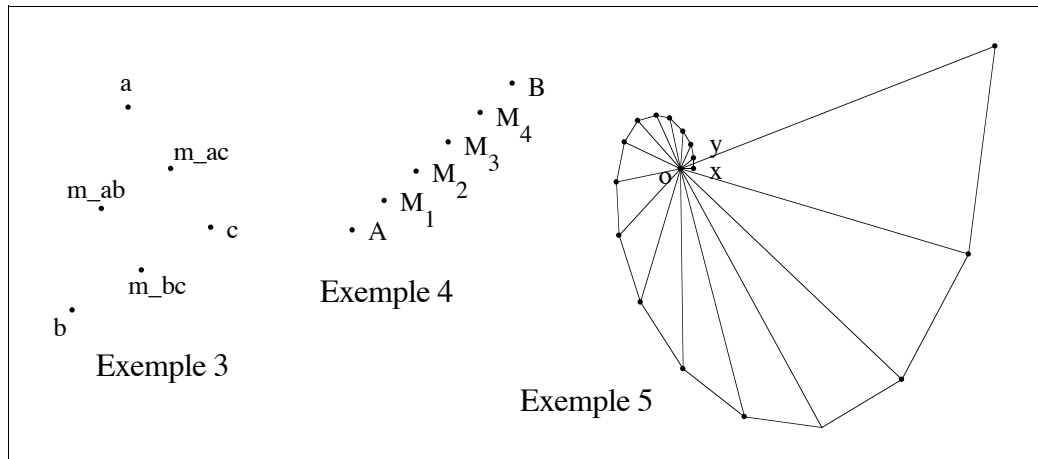


figure II.4-a : Problèmes linéaires.

Problèmes linéaires.

La complétude de notre résolveur pour les systèmes d'équations linéaires trouve une application dans le cas de figures ne comportant que des relations "milieu" puisque cette relation est linéaire. Parmi les contraintes géométriques considérées à la section précédente II.4-a, la notion de milieu est effectivement la seule à être linéaire (avec le parallélisme, mais sans l'incidence, le parallélisme ne sert à rien). Il n'y a pas de contrainte du second degré en une seule variable. Dans quelques cas particuliers, cette complétude donne des résultats intéressants.

Exemple 3 : La spécification suivante d'un triangle avec les milieux des cotés est linéaire :

- spécification : point(a), point(b), point(c), point(m_ab), point(m_ac), point(m_bc),
 m_{ab} milieu de [a,b], m_{ac} milieu de [a,c], m_{bc} milieu de [b,c].

Cette spécification permet d'obtenir la construction du triangle à partir de tous les ensembles de points de base comportant 3 points non liés. Par exemple :

- ensemble 1 de points de base : point(a), point(b), point(c).
- ensemble 2 de points de base : point(a), point(b), point(m_ac).
- ensemble 3 de points de base : point(a), point(m_ab), point(m_ac).
- ensemble 4 de points de base : point(m_ab), point(m_ac), point(m_bc).

Exemple 4: La spécification suivante ne comporte que des contraintes de milieu, elle est donc linéaire:

- spécification : point(A), point(B), point(M₁), point(M₂),...,point(M_n), M₁ milieu de [A,M₂], M₂ milieu de [M₁,M₃], ..., M_n milieu de [M_{n-1},B].

Cette spécification possède (n+2) points et n contraintes de milieu. D'un point de vue algébrique, le système de contraintes correspondant comprend $2*(n+2)$ variables et $2*n$ équations. Il suffit donc d'instancier deux points quelconques pour construire toute la figure.

<p>Environnement :</p> <p>coupe(A,B,B,0) -> ;</p> <p>coupe(A,B,C,n) -></p> <p style="padding-left: 40px;">B milieu de [A,D]</p> <p style="padding-left: 40px;">coupe(B,D,C,n-1);</p> <p>Requête :</p> <p>coupe(A,B,n);</p>
--

Figure II.4-b : Programme géométrique

D'un point de vue géométrique, cette spécification coupe un segment $[A,B]$ en $n+1$ segments $[M_i, M_{i+1}]$ d'égale longueur.

La forme de la spécification nous invite à penser à la définition d'un langage de programmation géométrique évolué. Par exemple, nous pourrions avoir le programme figure II.4-b constitué d'un ensemble de règles (environnement), et d'une requête.

Exemple 5 : La spécification suivante à base de contraintes de milieu donne une spirale dont 3 points quelconques non liés permettent de définir l'ensemble. Construction de M_{n+1} à partir de M_n , M_{n-1} et o :

- Spécification : point(M_n), point(M_{n-1}), point(o), point(M_{n+1}), point(m), point(n), point(p), point(q), point(r), m milieu de $[M_n, M_{n-1}]$, n milieu de $[M_n, m]$, p milieu de $[M_n, n]$, p milieu de $[q, M_{n-1}]$, q milieu de $[r, p]$, M_{n+1} milieu de $[q, r]$.

Spécifications constructives.

La section II.3-e démontre la complétude géométrique de l'algorithme de résolution pour toute figure dont la spécification est constructive vis à vis d'un ensemble de points de base choisi.

Ainsi, l'ensemble des 512 configurations géométriques recueillies par Chou dans sa thèse sont obtenues dans notre environnement avec une spécification et un ensemble de points de base adaptés de ceux donnés dans son livre.

L'ensemble des constructions faites avec Cabri I (première version de Cabri) ou Cabri II (sans les ellipses, lieux de points et points à l'infini), ou Sketchpad (dont je connais moins les capacités), ou d'une manière générale les logiciels de construction géométrique disponibles pour l'enseignement de la géométrie, est obtenu par notre approche.

D'une manière générale, toute figure, aussi complexe soit elle, correspondant à un procédé de construction à la règle et au compas peut être obtenue dans notre système.

Exemple 6 : Nous reprenons la figure I.1-e: construction du pentagone régulier à partir de son centre et d'un sommet. La figure est donnée par :

- spécification : point(M), point(M'), point(N), point(N'), point(a), point(a'), point(b), point(c), point(d), point(f), point(j), point(o), cercle C''(M), cercle C'(b), cercle C(o), o milieu de [a,a'], (o,j) \perp (a,o), b milieu de [o,a'], a \in cercle C(o), j \in cercle C(o), j \in cercle C'(b), c \in (a,o), c et a du même côté de (o,j), c \in cercle C'(b), d milieu de [c,o], (d,M) \perp (a,o), M \in cercle C(o), a \in cercle C''(M), M et N de chaque côté de (o,j), N \in cercle C(o), N \in cercle C''(M), d milieu de [M,M'], (N,N') \perp (a,o), f \in (a,o), f \in (N,N'), f milieu de [N,N'],
- points de base : point(a), point(o).

Exemple 7 : La construction d'un triangle équilatéral à partir d'un côté donné est possible. La spécification est beaucoup plus simple :

- spécification : point(a), point(b), point(c), cercle Ca(a), cercle Cb(b), b \in cercle Ca(a), a \in cercle Cb(b), c \in cercle Ca(a), c \in cercle Cb(b),
- points de base : point(a), point(b).

Cette spécification ne donne pas un rôle symétrique pour a,b et c. Et de fait, la construction n'est possible que si le côté [a,b] est donné. Pour obtenir la construction du triangle équilatéral (abc) à partir du côté [b,c], il y a une solution simple : compléter la spécification pour qu'elle soit symétrique en a, b et c. Les contraintes supplémentaires à ajouter, "cercle Cc(c), a \in cercle Cc(c), b \in cercle Cc(c)," sont des contraintes redondantes pour la figure initiale qui signifient que le triangle est équilatéral. Ces contraintes auraient tout aussi bien pu avoir été introduites initialement pour vérifier que la construction donne bien un triangle équilatéral.

La spécification enrichie permet la construction du triangle équilatéral à partir de n'importe quel côté du triangle.

Commentaire : D'une manière générale, les constructions obtenues grâce au résultat de complétude utilisé ici n'ont rien d'étonnant. C'est le minimum attendu. Cependant plusieurs idées intéressantes apparaissent avec les 7 exemples proposés.

Avec les exemples 1 et 2, nous avons une spécification constructive pour un ensemble de points de base donné, appliquée à un autre ensemble de points de base. La spécification 1 est constructive pour la figure 1, cela paraît évident. Appliquée à la figure 2, l'effort humain pour produire une spécification constructive est donc nul, et pourtant il est suffisant pour résoudre le problème 2 grâce à l'aide offerte par la construction automatique.

Les exemples 3-5 et 7 donnent des figures dont l'animation est possible à partir de n'importe quel élément. La classe d'animation de ces figures est maximale. Elle n'est pas restreinte aux éléments de base de la "construction" initiale.

Ces deux points, l'obtention de figure "sans effort" et l'obtention de classes d'animation maximale, sont développés dans la suite du chapitre. Ils nous semblent intéressants en vue d'applications pour l'enseignement de la géométrie, ou plus exactement pour la construction d'EIAH de la géométrie.

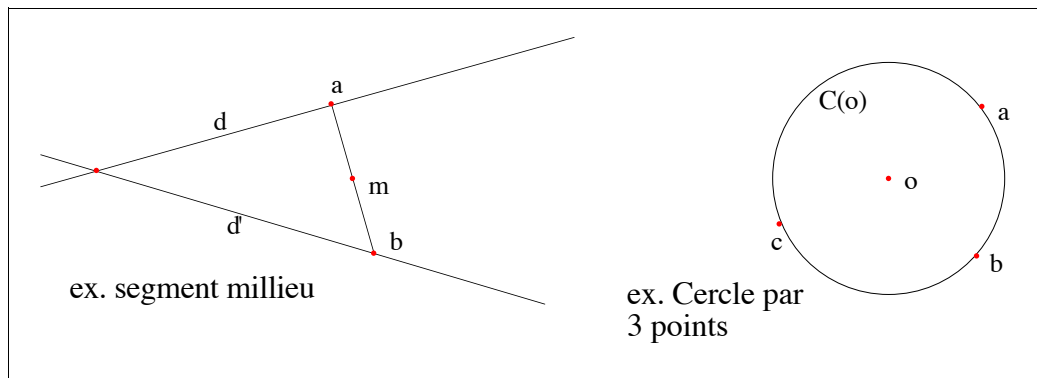


figure II.4-c : Résultats positifs.

c) Résultats positifs.

Parmi les problèmes classiques de construction quelques solutions sont obtenues par notre algorithme. En ce qui concerne les constructions classiques d'un triangle à partir des médianes, hauteurs, médiatrices, la section II.4-e récapitule les succès et échecs.

Segment porté par deux droites connaissant son milieu.

L'énoncé du problème complet est le suivant : Construire un segment $[a,b]$, tel que a et b soient sur deux droites données d et d' , en connaissant le milieu m du segment.

La spécification du problème est la suivante :

- spécification : point(a), point(b), point(m), droite(d), droite(d'), $a \in (d)$, $b \in (d')$, m milieu de $[a,b]$;
- points de base : point(m), droite(d), droite(d').

C'est le cas typique où il est facile de faire la construction à l'envers du problème pour obtenir une figure. La spécification de la situation à l'envers est :

- spécification : point(a), point(b), point(m), droite(d), droite(d'), $a \in (d)$, $b \in (d')$, m milieu de $[a,b]$;
- points de base : point(a), point(b), droite(d), droite(d').

Le problème de construction correspond à cette même spécification, seul l'ensemble de points de base change. Notre algorithme arrive à obtenir une figure dans les deux cas. Nous verrons en III.4-a comment cette construction algébrique permet de trouver une construction géométrique dans le cadre d'un EIAH de géométrie.

Cercle passant par 3 points donnés.

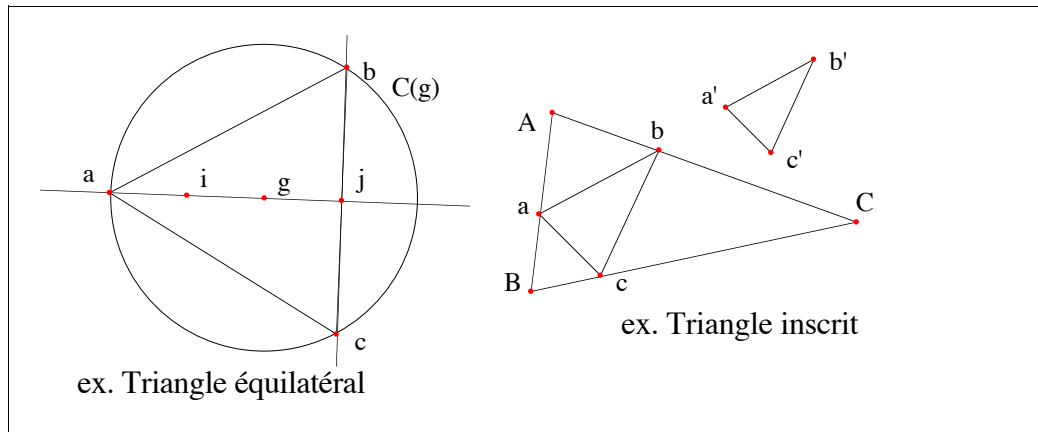


figure II.4-d : Résultats positifs (suite).

C'est un problème classique, dont la solution passe par la construction des médiatrices du triangle. La spécification du problème est la suivante :

- spécification : point(a), point(b), point(c), point(o), cercle C(o), $a \in \text{cercle } C(o)$, $b \in \text{cercle } C(o)$, $c \in \text{cercle } C(o)$;
- points de base : point(a), point(b), point(c).

La construction est possible grâce aux cordes communes ajoutées dans l'algorithme de résolution. En effet, ces cordes communes définissent ici les médiatrices de $[a,b]$, $[a,c]$, et $[b,c]$. Et le point o est à l'intersection de ces médiatrices.

Notons au passage qu'il est nécessaire de vérifier un grand nombre de contraintes redondantes : d'une part que les trois médiatrices se coupent en un seul point, d'autre part que le cercle de centre o et passant par le point a passe aussi par les points b et c.

Triangle équilatéral à partir d'un sommet et de son centre de gravité g.

Le problème complet est le suivant : Étant donné deux points a et g, construire un triangle équilatéral abc, tel que g soit le centre de gravité de ce triangle.

Si aucune construction directe n'est connue, alors une construction à l'envers peut-être envisagée. Éventuellement, cette construction peut aider à la résolution du problème. C'est ce que nous faisons à la fin de cette sous-section. Cas 2.

Mais auparavant supposons qu'une construction est connue. Alors, on peut se demander si cette spécification permet d'obtenir des figures avec pour points de base les sommets b et c, ou s'il faut la symétriser pour obtenir les constructions de abc à partir d'un sommet b ou c et de g. Cas 1.

Cas 1.

Supposons que la construction de (abc) triangle équilatéral en fonction de a et g, corresponde à la figure suivante :

- spécification : point(a), point(b), point(c), point(i), point(j), point(g), cercle C(g), i milieu de $[g,a]$, g milieu de $[i,j]$, $(j,b) \perp (g,a)$, $a \in \text{cercle } C(g)$, $b \in \text{cercle } C(g)$, j milieu de $[b,c]$;

- points de base : Point(a), Point(g).

Cette spécification permet-elle d'obtenir une figure pour l'ensemble de points de base b, g ? La réponse est non. Cependant, si l'on ajoute la spécification d'un triangle équilatéral, comme nous l'avons écrite pour l'exemple 7, alors la construction est possible. En l'occurrence, il suffit de rajouter seulement "a ∈ cercle Cb(b), et c ∈ cercle Cb(b)," ou même seulement la contrainte "(g,b) ⊥ (a,c)".

La spécification obtenue permet d'obtenir toutes les animations possibles de la figure comprenant un triangle équilatéral, et son centre. En effet, pour chaque ensemble de points de base imaginable, c'est à dire pour les 6 ensembles de deux points parmi les 4 points suivants a, b, c, et g, la spécification est suffisante.

Cas 2.

Supposons maintenant qu'aucune construction d'un triangle équilatéral à partir d'un sommet a et du centre de gravité g ne soit connue. Une spécification à l'envers du problème correspond à la construction d'un triangle équilatéral et de son centre de gravité. C'est la figure suivante :

- spécification : point(a), point(b), point(c), point(m), point(n), point(l), point(g), cercle Ca(a), cercle Cb(b), cercle Cc(c), b et c ∈ cercle Ca(a), a et c ∈ cercle Cb(b), a et b ∈ cercle Cc(c), m milieu de [a,b], n milieu de [b,c], l milieu de [a,c], g ∈ (m,c), g ∈ (n,a), g ∈ (l,b);
- points de base : Point(a), Point(g).

Cette spécification donne la figure pour les ensembles de points de base suivants :

- (a et b), (a et c), (b et c) : nous l'avons comme pour l'exemple 7,
- (a et m), (a et l), (b et m), (b et n), (c et n), (c et l) : c'est trivial car les milieux donnent aussitôt un côté du triangle, ce qui nous ramène au cas précédent,
- (m et n), (l et n), (l et m) : c'est moins trivial.

Mais cette spécification ne donne pas de figure pour les ensembles de points de base (a et g), (b et g), et (c et g). Pour qu'elle le soit, il suffit d'ajouter que g est au tiers de la médiane, et que a, b et c sont sur une même cercle de centre g. Alors, les ensembles de base suivants permettent de construire la figure : (a et g), (a et n), (n et g), (b et g), (b et l), (l et g), (c et g), (c et m) et (m et g) !

La spécification obtenue permet d'obtenir toutes les animations possibles de la figure comprenant un triangle équilatéral, son centre et les milieux des cotés. En effet pour chaque ensemble de points de base imaginable, c'est à dire pour l'un des 21 ensembles de deux points parmi les 7 points suivants a, b, c, g, l, m et n, la spécification permet d'obtenir une figure.

Triangle inscrit.

L'énoncé du problème est le suivant : Comment construire un triangle (abc) inscrit dans un triangle (ABC) connaissant les directions de (abc).

La spécification du problème est la suivante, les directions sont celles d'un triangle auxiliaire (a'b'c') :

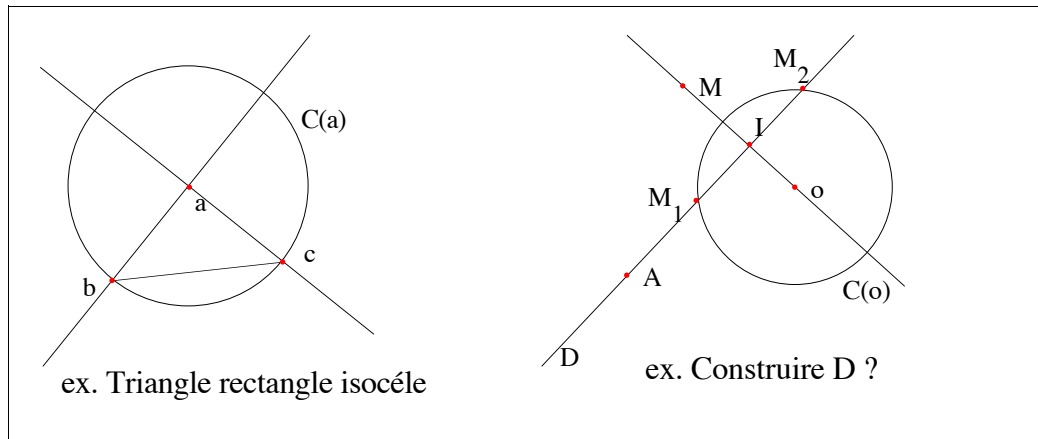


figure II.4-e : Résultats négatifs.

- spécification : point(A), point(B), point(C), point(a), point(b), point(c), point(a'), point(b'), point(c'), $a \in (A,B)$, $b \in (B,C)$, $c \in (A,C)$, $(a',b') \parallel (a,b)$, $(b',c') \parallel (b,c)$, $(a',c') \parallel (a,c)$;
- points de bases : point(A), point(B), point(C), point(a'), point(b'), point(c').

La figure est obtenue directement avec cette spécification. Le problème revient en fait à la recherche d'un rapport d'homothétie pour construire le bon triangle (abc) semblable qui s'inscrit dans (ABC), le parallélisme imposant des contraintes linéaires. Tout le problème se restreint donc à la recherche du rapport d'homothétie, ce qui est résolu avec l'algorithme de construction proposé. Souvent dans des cas similaires où le problème revient à la recherche d'une homothétie, l'algorithme de résolution donne de bon résultats.

d) Résultats négatifs.

L'ensemble des spécifications insuffisantes que nous avons rencontrées peut être décomposé en deux groupes. Le premier groupe comporte des problèmes difficiles à résoudre. Le second groupe concerne des problèmes relativement simples que nous aurions aimé voir résolus.

Problèmes ardu.

Parmi les problèmes ardu, nous en citerons seulement deux :

- pour son élégance : la spécification minimale d'un pentagone régulier. A savoir : construire 5 points a,b,c,d,e sur un cercle, tels que $labl=|bcl|=|cdl|=|del|=|eal$,
- le problème de Steiner : construire un triangle (abc) avec a, b et c sur trois droites données concourantes et les cotés ab, ac, et bc passant par trois points donnés. Ce problème est intéressant car il ne comporte que des équations de droites.

Problèmes "faciles"

Parmi les problèmes relativement faciles que nous aurions aimé résoudre, nous pouvons en citer deux. Ces deux problèmes sont instructifs car ils nous ouvrent des perspectives.

Triangle rectangle isocèle

Le premier problème est la recherche d'un triangle (abc) rectangle et isocèle en a connaissant l'hypoténuse bc. La figure est simple :

- spécification : point(a), point(b), point(c), cercle C(a), $b \in \text{cercle } C(a)$, $c \in \text{cercle } C(a)$, $(a,b) \perp (a,c)$;
- points de base : point(b), point(c).

Cette spécification ne permet pas l'obtention de figures pour l'ensemble des points de base (b et c). En fait, la contrainte " $(a,b) \perp (a,c)$ " signifie que le point a est sur le cercle de diamètre [b,c]. Il suffit d'ajouter "i milieu de [b,c], $b \in \text{cercle } C(i)$, $a \in \text{cercle } C(i)$ " pour que la spécification soit constructive.

Cet ajout peut se faire automatiquement comme l'ajout des cordes communes. Cependant, le cas idéal où la contrainte " $(a,b) \perp (a,c)$ " est dans la spécification ne représente pas tous les cas possibles. Si la recherche des " $(a,b) \perp (a,c)$ " peut se faire en temps quasi linéaire, il faut aussi songer à la recherche des contraintes associées qui correspondent à la même situation, par exemple les cas où il y a " $a \in (d)$, $b \in (d)$, $a \in (d')$, $c \in (d')$, $(d) \perp (d')$ ". La recherche de tels cas de figure peut prendre un temps de complexité polynomiale de l'ordre de $O(n^5)$.

La juste mesure pour l'ajout automatique d'éléments est un travail délicat. Nous n'avons pas entrepris pour l'instant de plus amples travaux dans ce sens, mais nous voyons bien l'intérêt et la difficulté de telles démarches.

Un second problème relativement simple résiste à notre résolveur : étant donné un cercle C(O) ainsi que A et M deux points du plan, on cherche à construire une droite D passant par A, coupant le cercle en M_1 et M_2 , tels que la droite passant par M et I le milieu de $[M_1, M_2]$, soit perpendiculaire à D. La figure correspond à la spécification suivante :

- spécification : point(A), point(I), point(M), point(M_1), point(M_2), point(O), droite(D), cercle C(O), $A \in (D)$, $M_1 \in (D)$, $M_2 \in (D)$, $M_1 \in \text{cercle } C(O)$, $M_2 \in \text{cercle } C(O)$, I milieu de $[M_1, M_2]$, $(D) \perp (M,I)$;
- éléments de base : point(A), point(M), cercle C(O).

La solution est obtenue dès que la droite (M,I) est connue. Or, (M,I) est la médiatrice de $[M_1, M_2]$, comme la figure II.4-e le montre. Elle passe donc par O. Mais la contrainte "O E (M,I)" n'est pas dans la spécification initiale. Avec cette spécification supplémentaire, le problème est résolu. Cette spécification peut être ajoutée avec la technique évoquée au paragraphe précédent, mais le coût algorithmique est d'ordre $O(n^7)$.

Une autre solution a été imaginée : essayer de trouver d'autres propriétés dans la figure. Cela peut se faire en construisant la figure à partir d'un autre ensemble de points de base. Ici par exemple, en prenant M, M_1 , M_2 comme ensemble de points de base, le cercle C(O) est construit, et on peut observer sur la figure (incomplète) que O est sur (M,I). Cette exploration permet d'obtenir la contrainte supplémentaire qui résout le problème (c'est la méthode envisagée en II.3-e).

e) Récapitulatif pour le triangle et ses éléments caractéristiques.

Cette section comporte les tableaux récapitulatifs des figures comportant un triangle et ses éléments caractéristiques.

Triangle et orthocentre.

Commençons avec la figure comprenant le triangle (abc) et l'orthocentre h. Nous avons vu depuis le chapitre II-1 que tous les ensembles de triplets de points (a, b, c), (a, b, h), (b, c, h) et (a, c, h) forment un ensemble de points de base pour lequel la spécification suivante est constructible :

- spécification : point(a), point(b), point(c), point(h), (h,a)⊥(b,c), (h,b)⊥(c,a), (h,c)⊥(a,b).

Considérons maintenant le triangle avec les pieds des hauteurs et l'orthocentre. La spécification considérée est la suivante. :

- spécification : point(a), point(b), point(c), point(h), point(h_a), point(h_b), point(h_c), (h,a)⊥(b,c), (h,b)⊥(c,a), (h,c)⊥(a,b), h_a ∈ (b,c), h_a ∈ (h,a), h_b ∈ (a,c), h_b ∈ (h,b), h_c ∈ (a,b), h_c ∈ (h,c).

Le tableau II.4-a suivant récapitule les cas où la spécification permet d'obtenir la figure en fonction de l'ensemble des points de base. Les indications "oui" signifient que le point en tête de colonne est construit pour l'ensemble de base sur l'entête de ligne. Les "-" signifient que le problème est sous-contraint. Les cas symétriques ont été éliminés ((a, b, h_b), n'est pas considérés car il est équivalent par symétrie au cas (a, b, h_a), etc...).

	a	b	c	h	h _a	h _b	h _c
a b c	oui	oui	oui	oui	oui	oui	oui
a b h	oui	oui	oui	oui	oui	oui	oui
a b h _a	-	-	-	-	-	-	-
a b h _c	-	-	-	-	-	-	-
a h h _a	-	-	-	-	-	-	-
a h h _b	-	-	-	-	-	-	-
a h _a h _b	oui	oui	oui	oui	oui	oui	oui
a h _b h _c	oui	oui	oui	oui	oui	oui	oui
h h _a h _b	oui	oui	oui	oui	oui	oui	oui
h _a h _b h _c	non	non	non	non	oui	oui	oui

Tableau II.4-a : Triangle, hauteurs et orthocentre.

Le seul échec correspond à la recherche de la construction du triangle (abc) à partir des pieds des hauteurs. La solution de ce problème difficile fait intervenir deux extensions quadratiques : l'orthocentre h recherché est le centre du cercle inscrit dans le triangle (h_a h_b h_c). En fait, il y a 4 solutions, car il y a aussi 3 centres de cercle ex-inscrits à ce même triangle.

Rappel du II.3-a : le triangle donné par trois points m_a, m_b et m_c correspondant au milieu de l'orthocentre h avec un sommet a, b et c est constructible.

Triangle et centre de gravité.

Considérons le triangle avec les pieds des médianes et le centre de gravité. La spécification considérée est la suivante :

- spécification : point(a), point(b), point(c), point(g), point(m_a), point(m_b), point(m_c), m_a milieu de [b,c], m_b milieu de [a,c], m_c milieu de [a,b], $g \in (m_a, a)$, $g \in (m_b, b)$, $g \in (m_c, c)$.

Le tableau II.4-b récapitule les cas où la spécification permet d'obtenir la figure en fonction de l'ensemble des points de base. A part les cas où cela n'a pas de sens car le problème est sous-contraint ((a, b, m_c) et (a, g, m_a)), la figure est toujours construite. Ce résultat est lié à la complétude pour les problèmes linéaires. C'est un résultat prévisible.

	a	b	c	g	m _a	m _b	m _c
a b c	oui	oui	oui	oui	oui	oui	oui
a b g	oui	oui	oui	oui	oui	oui	oui
a b m _a	oui	oui	oui	oui	oui	oui	oui
a b m _c	-	-	-	-	-	-	-
a g m _a	-	-	-	-	-	-	-
a g m _b	oui	oui	oui	oui	oui	oui	oui
a m _a m _b	oui	oui	oui	oui	oui	oui	oui
a m _b m _c	oui	oui	oui	oui	oui	oui	oui
g m _a m _b	oui	oui	oui	oui	oui	oui	oui
m _a m _b m _c	oui	oui	oui	oui	oui	oui	oui

Tableau II.4-b : Triangle, médianes et centre de gravité.

Triangle et cercle circonscrit.

Considérons le triangle avec les pieds des médiatrices et le centre du cercle circonscrit. La spécification considérée est la suivante :

- spécification : point(a), point(b), point(c), point(C), point(M_a), point(M_b), point(M_c), M_a milieu de [b,c], M_b milieu de [a,c], M_c milieu de [a,b], $(C, M_a) \perp (b, c)$, $(C, M_b) \perp (c, a)$, $(C, M_c) \perp (a, b)$.

Le tableau II.4-c suivant récapitule les cas où la spécification permet d'obtenir la figure en fonction de l'ensemble des points de base.

	a	b	c	g	m _a	m _b	m _c
a b c	oui	oui	oui	oui	oui	oui	oui
a b C	-	-	-	-	-	-	-
a b M _a	-	-	-	-	-	-	-
a b M _c	-	-	-	-	-	-	-
a C M _a	-	-	-	-	-	-	-
a C M _b	-	-	-	-	-	-	-
a M _a M _b	oui	oui	oui	oui	oui	oui	oui
a M _b M _c	oui	oui	oui	oui	oui	oui	oui
C M _a M _b	oui	oui	oui	oui	oui	oui	oui
M _a M _b M _c	oui	oui	oui	oui	oui	oui	oui

Tableau II.4-c : Triangle, médiatrice et centre du cercle circonscrit.

Une étude plus large.

Une étude plus large des constructions possibles dans un triangle a été entreprise. Elle concerne des figures plus complètes, comportant des points et des droites. Il est alors souvent nécessaire d'autoriser que ces objets soient instanciés à moitié :

- pour un point, cela correspond à fixer le point sur une droite quelconque,
- pour une droite, cela correspond à la faire passer par un point quelconque.

La complexité du problème est alors beaucoup plus grande. Pour en donner une idée, l'annexe 5 comporte une liste de 77 problèmes de construction possibles avec la situation classique comportant hauteurs d'un triangle. Cette liste, établie "à la main", évite des situations symétriques, ou sous-contraintes. Automatiquement il est beaucoup plus difficile d'éviter les cas redondants. Et l'analyse des résultats est délicate. Par exemple, pour une situation comportant un triangle, les médianes et le centre de gravité, un algorithme de recherche d'ensemble de points de base pour lequel l'algorithme de construction réussit, trouve jusqu'à 279 solutions.

Cependant nous pouvons tirer les conclusions suivantes :

- pour la figure sur les médianes, sans la contrainte redondante "g est au tiers des médianes", l'algorithme de recherche d'ensemble de points de base donne 132 solutions. Avec cette contrainte redondante, l'algorithme trouve 279 solutions.

- pour la figure sur les médiatrices, sans la contrainte redondante "a, b, c sont cocycliques sur le cercle circonscrit de centre C", l'algorithme trouve 114 solutions. Avec cette contrainte redondante, l'algorithme trouve 215 solutions.

Dans tous les cas, l'ajout de contraintes redondantes améliore significativement les performances de l'algorithme de construction. Ces contraintes redondantes sont donc très importantes. Ceci renforce notre choix pour une arithmétique exacte qui autorise l'emploi de ces contraintes.

conclusion de ce chapitre :

Le présent chapitre porte sur un problème difficile, celui de la construction automatique de figures géométriques exactes. Nous avons vu que l'utilisation d'un algorithme naïf permet l'obtention de classes de problèmes importantes pour lequel cet algorithme est complet et correct. Dans tous les cas cet algorithme est correct, mais dans le cas général il reste incomplet. Les conclusions du II-3 donnent des perspectives d'améliorations.

Notons que la nature "exacte" des figures obtenues est un point important. Elle autorise l'utilisation de spécifications redondantes particulièrement utiles comme la fin de ce chapitre l'a montré.

Conclusion de la seconde partie :

La seconde partie a porté sur la définition d'un environnement de programmation de type CLP adapté au domaine de la géométrie. L'objectif a été l'obtention de figures géométriques exactes à partir d'une spécification éventuellement redondante. Cet objectif a été atteint avec un succès total pour les spécifications constructives et dans une moindre mesure dans le cas général.

Notre intention est maintenant d'utiliser cet environnement CLP(Géométrie) pour la définition d'applications où il ne sera plus l'élément central de nos préoccupations, mais le moteur de nos travaux.

III. Applications.

Introduction de la troisième partie.

Les résultats présentés dans la seconde partie portent sur deux aspects. Le premier concerne une représentation normale des nombres constructibles. Le second aspect définit l'environnement CLP(Géométrie) associé aux contraintes quadratiques. Cette troisième partie intitulée "Applications" porte sur l'utilisation des résultats obtenus dans ces deux domaines pour la définition d'applications appartenant à trois domaines distincts : celui du calcul formel, celui de la géométrie et plus particulièrement des démonstrateurs automatiques, enfin celui des environnements informatisés d'apprentissage humain de la géométrie (EIAH de géométrie).

Ce sont des applications au sens où les deux aspects cités précédemment sont employés largement mais où ils ne sont jamais détaillés. Ils apparaissent comme des "boîtes noires" sur lesquelles se fondent ces applications. Ces dernières sont donc indépendantes de la mise en œuvre des concepts précédents. Il est possible de lire cette partie indépendamment des deux autres. Tout comme il est possible de lire chacune des deux autres parties indépendamment.

Tout d'abord, au chapitre III-1, une application immédiate, et indépendante de la géométrie, est donnée. Elle porte sur l'expression et la simplification des expressions arithmétiques comportant des racines carrées. Le milieu visé est l'enseignement des sciences dans le secondaire où l'emploi de racines carrées est courant. Des comparaisons sont effectuées avec les logiciels de calcul formel, ainsi qu'avec les travaux de [Borodin, Fagin, Hopcroft, Tompa-85].

Ensuite, au chapitre III-2 et III-3, des applications géométriques sont présentées. Il s'agit d'une part d'expliquer les méthodes utilisées en partie I pour obtenir des figures rationnelles. D'autre part, la représentation normale des nombres constructibles est utilisée pour la vérification des contraintes redondantes. En III-3, nous appliquons ces travaux à la démonstration automatique de théorèmes de géométrie. Des comparaisons sont effectuées avec les travaux de [Wu-94], [Chou-88], [Hong-86] et [Deng, Zhang, Yang-90].

Enfin, au chapitre III-4, les aspects initiaux de mon travail sont abordés, c'est à dire la réalisation d'un environnement informatisé d'apprentissage humain de la géométrie. Trois réalisations sont données. L'une est tournée vers l'enseignement de la géométrie de manière déclarative, cf. [Bouhineau-95a]. La seconde réalisation est orientée vers l'aide aux professeurs pour la vérification de figures et la recherche de contre-exemples, cf. [Bouhineau, Channac-96]. Enfin, des travaux que j'ai initiés sur la réalisation d'un environnement de dessin géométrique à main levée sont également cités rapidement.

III-1. Simplification d'expressions arithmétiques avec racines carrées pour le calcul formel.

Introduction :

L'intérêt présenté par les expressions arithmétiques avec racines carrées est principalement concentré dans le domaine de l'enseignement scientifique du secondaire. Que ce soit en mathématiques, en sciences physiques ou en chimie, l'emploi de racines carrées est courant. Une preuve supplémentaire est donnée indirectement par les logiciels de calcul formel. En effet, parmi Derive, Maple et Mathematica, le premier est le plus porté sur l'enseignement secondaire et c'est le seul qui comporte des connaissances implicites sur la simplification des expressions avec racines carrées. Mathematica et Maple n'ont aucune connaissance sur la manipulation des racines carrées, excepté bien sûr l'élévation au carré.

Au delà de ce qui concerne l'enseignement scientifique, c'est un domaine qui a été étudié spécifiquement par [Borodin, Fagin, Hopcroft, Tompa-85] et [Zippel-85]. Sur le plan plus général, les travaux de [Landau-92] ont porté sur la simplification d'expressions avec racines de degré quelconque. Habituellement, les logiciels de calcul formel sont avides du genre de résultats obtenus là, mais excepté Derive, et encore pour une faible part, ce n'est pas le cas pour ces résultats ci. Notre approche est légèrement différente de celle de [Borodin, Fagin, Hopcroft, Tompa-85]. Elle porte sur l'utilisation d'une représentation normale des expressions avec racines carrées. Cette représentation constitue une forme de simplification efficace, comme le montre la suite.

Quelques exemples permettent de présenter et de critiquer des méthodes de représentation des expressions avec racines carrées utilisées en calcul formel. L'utilisation de notre représentation leur est comparée. Un algorithme de simplification est succinctement présenté. Il est comparé aux travaux de [Borodin, Fagin, Hopcroft, Tompa-85].

a) Représentation d'expressions avec racines carrées en calcul formel

Prenons quelques exemples simples pour comparer différentes représentations possibles d'une expression arithmétique comportant des racines carrées.

Modes de représentation d'une suite.

Soit la suite définie par la formule de récurrence suivante, qui simule une équation de droite :


```

suite(e,u,n) ->
  suite_tmp(e,n,u,v,w);

suite_tmp(e,0,u,d,t) ->
  ng_un(e,u)
  ng_deux(e,d)
  ng_sqrt(e,d,t);
suite_tmp(e,n,z,u,d) ->
  suite_tmp(e,n-1,u,d,t)
  ng_mult(e,u,d,x)
  ng_plus(e,x,t,z)
{n>0};

```

Figure III.1-a : Construction d'une suite.

$$\begin{cases} U_0=1, U_1=2, U_2=\sqrt{2} \\ U_n=U_{n-1}*U_{n-2}+U_{n-3} \end{cases}$$

Sur le plan mathématique, cette suite est définie dans l'extension algébrique $Q[\sqrt{2}]$, et le terme général vaut $U_n=a_n+b_n*\sqrt{2}$ où a_n et b_n des suites de nombres rationnels définies par :

$$\begin{cases} a_0=1, a_1=2, a_2=0 \\ b_0=0, b_1=0, b_2=1 \\ a_n=a_{n-1}*a_{n-2}+2*b_{n-1}*b_{n-2}+a_{n-3} \\ b_n=b_{n-1}*a_{n-2}+a_{n-1}*b_{n-2}+b_{n-3} \end{cases}$$

Avec Mathematica, cette suite est définie classiquement par le programme suivant :

```

U[0]=2;
U[1]=2;
U[2]=SQRT[2];
U[n]=U[n-1]*U[n-2]+U[n-3];

```

Avec Prolog III, en utilisant l'arithmétique CLP(Nombres Constructibles), cette suite est définie par le programme figure III.1-a.

La suite produite par Mathematica est la suivante :

```

U3 = 2√2+1
U4 = √2(2√2+1)+2
U5 = (√2(2√2+1)+2)(2√2+1)+√2
U6 = ((√2(2√2+1)+2)(2√2+1)+√2)(√2(2√2+1)+2)+2√2+1
U7 = (((√2(2√2+1)+2)(2√2+1)+√2)(√2(2√2+1)+2)+2√2+1)
      * ((√2(2√2+1)+2)(2√2+1)+√2) + √2(2√2+1)+2
U8 = (nécessite 4 lignes)
...

```

U_{20} = échec (après plusieurs minutes d'attente, les limites en mémoire de la machine sont atteintes).

Avec la représentation CLP(géométrie), nous obtenons avec Prolog :

$$U_3 = (1, 2)$$

$$U_4 = (6, 1)$$

$$U_5 = (10, 14)$$

$$U_6 = (89, 96)$$

$$U_7 = (3584, 2207)$$

$$U_8 = (742730, 540501)$$

$$U_9 = (5047715823, 3576360790)$$

$$U_{10} = (7615143139931954, 5384565899606230)$$

$$U_{11} = \dots$$

...

U_{20} = résultat après une quinzaine de secondes.

Commentaire succinct : les formules proposées par Mathematica sont rapidement trop longues et difficiles à lire. Par exemple, l'écriture de U_{12} demande plus d'une page. Il ne s'agit pas ici de dire que Mathematica est moins puissant que Prolog III pour effectuer du calcul formel. Il s'agit de mettre en avant que, sans un effort mathématique, les nombres constructibles sont hors d'atteinte avec une représentation naïve. Et pourtant ici, les nombres constructibles en présence sont simples du point de vue algébrique. Il n'y a qu'une seule extension algébrique quadratique.

Notons que dans le cas où le programme Mathematica simplifie les formules à chaque étape, les résultats obtenus sont semblables à ceux du programme Prolog. Cependant, les temps de calculs sont nettement plus importants

Maple se comporte de manière semblable à Mathematica.

Le logiciel Derive sur la calculatrice TI-92 se débrouille très bien. En effet, en des temps comparables à ceux d'un Macintosh, les formules simplifiées sont obtenues directement. Cependant, Derive échoue dans le calcul du terme u_{19} de la suite, comme Prolog III échoue après les calculs de u_{23} ; dans les deux cas, nous supposons que les entiers qui obtenus sont trop grands. Dans le cas de Prolog III, les entiers sont trop grands pour être affichés ! La complexité relevant de la taille des entiers reste un problème délicat (cf. conclusion du II.2).

Évaluation d'une formule booléenne.

Prenons les deux formules booléennes A et B ci-dessous pour les évaluer. Le lecteur pourra vérifier aisément, en élevant au carré les deux membres de chaque égalité, que A et B sont "vrai".

$$A : (\sqrt{2} + \sqrt{3} = \sqrt{5 + 2\sqrt{6}})$$

$$B : (1 + \sqrt{2} + \sqrt{3} = \sqrt{5 + 2\sqrt{2} + 2\sqrt{3} + 2\sqrt{6}})$$

Avec la représentation réduite utilisée pour CLP(géométrie), l'évaluation est correcte, elle donne pour les deux formules le booléen "vrai".

Avec les logiciels Mathematica et Maple les deux évaluations échouent : par précaution, en l'absence de preuves positives, elles évaluent à "faux" ces deux booléens. Cependant, une évaluation "numérique" donne la bonne réponse. Selon toutes les possibilités d'évaluation numérique proposées, les booléens sont "vrai". Mais quel sens ces évaluations numériques ont-elles ? Sont-elles garanties comme dans le projet Exact Real, cf. [Yap, Dubé-94]. Notons que, étape après étape, conduite par l'homme, Mathematica arrive à prouver l'égalité formelle; il en est de même pour Maple.

Avec Derive, qui connaît certaines simplifications particulières, la première formule A est évaluée correctement. Mais la seconde formule B est mal évaluée, et la calculatrice répond "false".

b) Algorithme de simplification, exemples.

Pour la simplification de formule notre objectif principal est le suivant : avoir le moins de racines différentes distinctes. Un objectif secondaire consiste à favoriser une expression simplifiée la plus proche possible de l'expression initiale. Trois simplifications interviennent :

- 1^{ère} simplification : Le codage condensé (normal) permet d'avoir un ensemble minimum de racines différentes. Exemple : $A = 2\sqrt{6} - \sqrt{2}\sqrt{3} \rightarrow 2\sqrt{2}\sqrt{3} - \sqrt{2}\sqrt{3}$.
- 2^{nde} simplification : Les opérations arithmétiques après réécriture dans une base minimale simplifient l'expression. Exemple : $A \rightarrow \sqrt{2}\sqrt{3}$.
- 3^{ième} simplification : Le regroupement des produits de racines dans le cas où la base est trop riche pour l'expression donnée simplifie encore l'expression. Exemple : $A \rightarrow \sqrt{6}$.

Ces trois simplifications sont prises en compte dans l'algorithme suivant :

Algorithme informel de simplification d'expressions numériques avec racines carrées :

Soit E une expression constructible

1) Écriture sous forme standard comme somme de produit de racines carrées : $E = \sum e_k \prod \sqrt{a_i}$.

2) Construction de la liste L des racines carrées apparaissant (syntaxiquement) dans E. $L = \{\sqrt{a_i}\}$.

3) Tri de la liste L des racines selon un ordre qui respecte les dépendances syntaxiques (si $\sqrt{a_i}$ contient syntaxiquement $\sqrt{a_j}$, alors il doit être placé après), la liste triée est appelée L'.

4) Réduction de L' pour avoir une sous-liste normale L''.

5) Minimisation de L'' :

- les racines n'apparaissant plus dans E sont supprimées;
- les couples de racines apparaissant seulement sous forme de produit sont remplacés par leur produit.

L'algorithme garantit qu'un nombre minimal de racines distinctes est utilisé.

Exemple 1.

Reprenant l'exemple de la section précédente ($1+\sqrt{2}+\sqrt{3} = \sqrt{5+2\sqrt{2}+2\sqrt{3}+2\sqrt{6}}$), nous retrouvons l'égalité suivante, construite récursivement à partir de la précédente :

$$\sqrt{3+2\sqrt{6}+2\sqrt{5+2\sqrt{2}+2\sqrt{3}+2\sqrt{6}}} = 1+\sqrt{2}+\sqrt{3}$$

En théorie, les égalités suivantes sont également retrouvées :

$$\sqrt{3+2\sqrt{6}+2\sqrt{3+2\sqrt{6}+2\sqrt{3+2\sqrt{6}+2\sqrt{3+2\sqrt{6}+2\sqrt{3+2\sqrt{6}+2\sqrt{\dots}}}}}} = 1+\sqrt{2}+\sqrt{3}.$$

Dans la pratique, la manipulation d'expressions comportant plus de quelques (< 4) racines carrées différentes devient vite trop coûteuse en temps.

Exemple 2.

Certains algorithmes de simplification donnent des résultats un peu étranges. Avec Derive sur la calculatrice TI-92, nous avons cherché à simplifier l'expression

$$A = \frac{(22-4\sqrt{29})}{(110-20\sqrt{29})}$$

Avec notre algorithme, nous avons trouvé $A=1/5$, mais la réponse de Derive est :

$$A = \frac{(2\sqrt{29}-11)*(2\sqrt{29}+11)}{25}$$

La notion de simplification n'est pas si évidente.

c) Comparaison avec l'algorithme de simplification de Borodin, Fagin, Hopcroft et Tompa.

L'algorithme présenté au III.1-b assure que le nombre de racines employé pour l'expression d'une formule est minimal. Par rapport à cet algorithme, celui proposé dans [Borodin, Fagin, Hopcroft, Tompa-85] assure de plus que les racines carrées employées sont de poids minimum, c'est à dire que les imbrications de racines carrées sont aussi peu nombreuses que possibles. Mais ce dernier point est critiquable. Il favorise l'emploi d'autres racines, de poids plus faibles que celles qui apparaissent initialement dans la formule, pour l'expression du même réel.

Dans les deux cas, le nombre de racines carrées différentes nécessaires est le même. Ce nombre dépend du degré algébrique de la formule. Il est indépendant de son expression.

Comme il n'y a pas de représentation canonique des nombres constructibles, ce sont des facteurs "humains" qui décident du choix de chaque racine utilisée.

Notre algorithme favorise la conservation dans la formule initiale du maximum de racines carrées apparaissant dans la formule finale. Cependant, il n'est pas garanti que cette conservation soit maximale.

Le critère pour le choix des racines carrées employées n'a de raison d'être que dans une utilisation humaine de ces algorithmes de simplification. Il favorise la lecture et la reconnaissance, par l'homme, de la formule initiale dans la formule finale. Un exemple distinguant les deux algorithmes conclut cette section.

Exemple 3 : Comment simplifier l'expression $E = \sqrt{16 - 2\sqrt{29} + 2\sqrt{55 - 10\sqrt{29}}}$?

L'expression E contient trois racines distinctes. Si l'expression de E est minimale, E est de degré algébrique huit. Et effectivement E est de degré algébrique huit, comme le montre notre algorithme. Les trois racines carrées qui apparaissent sont suffisantes et nécessaires à l'expression de E.

L'algorithme de Borodin, et al. propose l'expression de E dans l'extension $Q[\sqrt{5}, \sqrt{29}, \sqrt{11 - 2\sqrt{29}}]$: $E = \sqrt{5 + \sqrt{11 - 2\sqrt{29}}}$. Cette dernière expression comporte des racines carrées qui n'apparaissent pas dans la formule initiale. Le degré algébrique de E est toujours huit, puisque trois racines sont nécessaires. Cependant, le lecteur peut-il reconnaître l'expression initiale ?

Conclusion de ce chapitre :

Ce chapitre est volontairement court. Il concerne un domaine, celui du calcul formel, que nous connaissons peu du point de vue scientifique. Mais c'est un domaine que nous connaissons bien en tant qu'utilisateur. C'est dans ce cadre qu'il faut placer ce chapitre. Il contient une part des observations que nous avons pu faire au sujet des logiciels de calcul formel. Il montre d'une part les difficultés inhérentes au domaine des simplifications d'expressions arithmétiques et d'autre part les limites d'un usage naïf des logiciels de calculs formels. Le manque d'intérêt que semble témoigner Mathematica ou Maple aux travaux de [Borodin, Fagin, Hopcroft, Tompa-85], [Zippel-85] et [Landau-92], n'est peut-être qu'apparent. Dans un contexte où les nombres réels côtoient les nombres complexes, les notions de simplification et de forme normale ne sont pas clairement définies. Enfin, la notion de "simplification" est délicate. Elle comporte une part humaine non négligeable, comme le montre l'exemple de Derive.

III-2. Vérification de contraintes redondantes en géométrie.

Introduction :

Les chapitres sur la construction automatique (cf. II-3 et II-4), sur la démonstration automatique (cf. I-1 et III-3), sur la manipulation de figures (cf. III-4), donnent une grande importance aux contraintes redondantes. Ceci impose la résolution du problème suivant : "*Comment vérifier numériquement la cohérence d'une spécification géométrique contenant des contraintes redondantes ?*". Une arithmétique exacte sur les nombres constructibles assure une vérification aisée. Cet apport, que l'on doit aux travaux purement algébriques du chapitre II-2, est décisif.

Cependant, nous développons dans ce chapitre l'idée qu'il est nécessaire et souvent suffisant de réduire la complexité des figures géométrique. En effet, l'utilisation d'une arithmétique exacte sur les nombres constructibles reste sous le coût d'une complexité exponentielle. Ceci limite son utilisation dans la pratique. Il est donc nécessaire de limiter la complexité des figures géométriques considérées. Comme le montrent les résultats en I-4 b sur les 512 théorèmes parus dans [Chou-88], la complexité géométrique peut s'effondrer avec des modifications judicieuses des constructions employées. Ces modifications permettent alors l'utilisation d'une arithmétique rationnelle. Ce qui simplifie beaucoup le problème initial.

Ce chapitre reprend à la section III.2-a l'apport du travail effectué en II-2 sur la représentation normale des nombres constructibles. Cette section présente également un panorama des arithmétiques possibles. Ensuite au III.2-b, sont présentées les modifications des constructions géométriques qui ont donné les résultats du I.4-b. Cette section tire sa matière du travail d'analyse des 512 situations géométriques de [Chou-88].

a) Aspect algébrique.

Arithmétique exacte.

L'utilisation d'une arithmétique exacte assure la vérification numérique des contraintes redondantes. Comme nous l'avons montré en II.2 et II.3, la représentation normale des nombres constructibles permet d'obtenir ce résultat. De plus, cette représentation normale permet l'utilisation des algorithmes d'élimination de Gauss et de résolution de systèmes non-linéaires du II.3. Notre objectif est donc réalisé.

Arithmétique usuelle.

Il est nécessaire de rappeler que l'arithmétique usuelle des ordinateurs, dite "à virgule flottante" n'est pas exacte et ne permet pas de vérifier précisément l'égalité de deux expressions. La littérature est remplie d'exemples de comportements aberrants de cette arithmétique approchée. Nous en citerons deux :

- un premier exemple est donné par [Yap, Dubé-94]: soit $a = 77617$ et $b = 33096$. Prenons l'expression $f = 333.75 b^6 + a^2 (11 a^2 b^2 - b^6 - 121 b^4 - 2) + 5.5 b^8 + a/2b$. La valeur de f est évaluée à

1.172603 pour un IBM système 370,

-1.18059e+21 pour la norme IEEE double précision,

-.99999...999998827e+17 pour Maple avec 20 chiffres significatifs,

.1000000011726039400531786318588 pour Maple avec 30 chiffres,

0.e+7 pour Mathematica avec 0 chiffres précis, quand les calculs sont effectués avec 20 chiffres,

-0.83 pour Mathematica avec 2 chiffres précis, quand les calculs sont effectués avec 40 chiffres,

alors que f vaut $-0.827396059946821368141165095479816291999$ (pour les 20 premiers chiffres significatifs).

- un deuxième exemple est donné par Jean-Michel Muller : soit la suite définie par la formule de récurrence $X_0 = 1,510005072136258$ et

$$X_{n+1} = (3X_n^4 - 20X_n^3 + 35X_n^2 - 24)/(4X_n^3 - 30X_n^2 + 70X_n - 50).$$

La limite de cette suite est :

4,000000033 si les calculs sont effectués avec 10 chiffres,

1,00000000000 si les calculs sont effectués avec 12 chiffres,

3,0000000000000 si les calculs sont effectués avec 14 chiffres,

4,000000000000033 si les calculs sont effectués avec 16 chiffres et

1,00000000000000000 si les calculs sont effectués avec 18 chiffres.

Le résultat correct est 1. Un utilisateur qui aurait fait le calcul sur 10 puis sur 16 chiffres aurait été conduit à penser que le bon résultat est très proche de 4.

Comme ces exemples le montrent même si les calculs sont limités aux rationnels, l'arithmétique usuelle des ordinateurs est incorrecte.

Arithmétiques évoluées.

D'autres arithmétiques pour les nombres constructibles, ou plus généralement pour les expressions arithmétiques avec racines d'ordre quelconque, ont été proposées. En particulier, le projet "Exact Computation Paradigm" de [Yap, Dubé-94] propose une évaluation exacte de type paresseuse. Cette évaluation porte sur les valeurs avec un double calcul de la précision nécessaire, et de la précision courante. De la même manière, une arithmétique "on line" peut être utilisée avec vérification de la précision nécessaire, ou calcul de l'erreur maximale. Dans [Hong-86], une démarche analogue est employée pour la démonstration automatique en

géométrie à partir d'exemples numériques approchés. Ces formes de représentation autorisent la vérification de contraintes redondantes au prix d'un calcul supplémentaire qui peut être onéreux. En tous cas, cette vérification n'est pas aussi immédiate qu'avec l'emploi d'une représentation normale.

Pour les calculs rationnels, nous pouvons nous intéresser aux travaux effectués en géométrie computationnelle. Dans ce domaine, la représentation canonique des nombres rationnels par un doublet d'entiers en précision infinie n'est pas la plus efficace, car le nombre de calculs est important. Une arithmétique paresseuse est proposée dans [Michelucci, Moreau-94], qui conservent l'expression formelle et une valeur approchée. Lorsque l'expression approchée demande à être plus précise, l'expression formelle est utilisée. Dans [Fortune, Van Wyk-93], la représentation des entiers est étudiée, une évaluation statique des expressions est effectuée pour éviter les surcoûts à l'exécution et obtenir une estimation de la précision nécessaire des calculs. Mais il se peut que l'estimation statique soit trop pessimiste, comme pour [Hong-86]. L'adaptation de ces travaux aux expressions algébriques quelconques, ou à l'arithmétique normale du II-2, est un problème ouvert.

Les projets d'arithmétiques évoluées sont distincts, selon que ce sont des réponses au coût des calculs [Fortune, Van Wyk-93][Michelucci, Moreau-94], ou à la précision absolue [Hong-86][Yap, Dubé-94]. Dans certains cas, le nombre de racines carrées n'est pas réellement une limite. Pour notre part, le nombre de racines carrées est limitant pour la complexité algorithmique. Notre arithmétique réalise une troisième voie, celle du calcul exact en présence de peu de racines carrées.

Dans le domaine de l'enseignement de la géométrie, rappelons que le nombre de calculs à produire est faible, mais que la complexité algébrique a priori est grande. Aussi, il se peut que l'estimation des erreurs apparaissant dans les propositions d'arithmétiques citées plus haut soit particulièrement défavorable. De plus, d'après les résultats du I-4, il s'avère que la complexité effective des calculs est faible. La plupart des figures sont rationnelles. Par suite, les avantages d'une représentation exacte laissant une grande place aux rationnels sont clairs. Notons de plus que parmi les 20 figures non rationnelles tirées de [Chou-88], la moitié s'effectue dans $Q[\sqrt{3}]$ à cause de triangles équilatéraux. Avec cette information, il est possible de rationaliser ces calculs.

De plus, l'ensemble de ces travaux pose le problème de l'intégration dans un langage de programmation type CLP. La communauté CLP s'est orientée plutôt vers l'arithmétique à base d'intervalles, qui n'autorise pas la vérification exacte et immédiate des égalités.

b) Aspect géométrique : Recherche d'une construction rationnelle.

En I.3, il est montré que les constructions géométriques ne sont pas toutes de complexité équivalente, et que le choix judicieux d'une construction permet de diminuer la complexité d'une figure. Rappelons que sur 512 théorèmes donnés dans [Chou-88], il ne subsiste grâce à

des choix judicieux que 20 théorèmes nécessitant des figures non-rationnelles, cf I.4-b. Nous explicitons ici la méthode qui permet de diminuer cette complexité algébrique.

Essentiellement, quatre situations particulières apparaissent de façon récurrente parmi les 512 théorèmes. Ce qui explique les résultats du I.4-b. Ces quatre constructions suffisent à donner la méthode. Les deux premières concernent des modifications locales. Pour reprendre la terminologie de I-4, il s'agit de :

- 1- prendre un point quelconque sur un cercle,
- 2- effectuer la construction de l'inverse d'un point par rapport à un d'un cercle.

Le premier cas concerne plus d'une centaine d'occurrences parmi les 512 théorèmes. Le second seulement 6 cas¹⁹. La troisième et la dernière situation concernent des modifications globales de la figure. Elles font intervenir :

- 3- la construction de bissectrices d'un angle,
- 4- des points construits à l'intersection d'un cercle et d'une droite ou de deux cercles.

Ici, les modifications à apporter sont globales au sens où il faut transformer l'ensemble de la figure, ou l'ordre des constructions. Rapidement, pour la construction de la bissectrice d'un angle, la modification consiste à changer l'ensemble des points de base. Ainsi, la bissectrice peut être donnée, et le reste de la figure peut s'obtenir en reconstituant l'angle par dédoublement. Parmi les 512 situations de Chou, 38 occurrences de bissectrices sont modifiées par cette méthode. La dernière manipulation est la plus délicate. Elle apparaît dans quelques cas seulement parmi les 512 situations, en particulier dans la construction de la moitié d'un arc (ce qui s'apparente à la construction d'une bissectrice).

Ces quatre situations sont explicitées ci après. Les modifications locales sont présentées hors contexte. Les modifications globales sont données sur deux exemples (ce sont des modifications contextuelles).

Point sur cercle.

La première situation concerne la construction de points quelconques à prendre sur des cercles existants. Soit un cercle de centre $C(x_c, y_c)$, passant par $R(x_r, y_r)$, C et R sont des points rationnels pour simplifier. Le carré du rayon est $K=(x_c-x_r)^2+(y_c-y_r)^2$. Soit à prendre un point quelconque $M(x_m, y_m)$ sur ce cercle.

La méthode usuelle consiste à prendre x_m arbitraire rationnel. En fixant l'abscisse x_m , il vient $y_m = y_c + \sqrt{K-(x_c-x_m)^2}$. C'est la méthode employée dans Chou. Elle est illustrée à la figure III.2-a. A priori, la racine carrée n'est pas rationnelle. Par exemple, pour $K=9$, $x_c=0$, $x_m=2$, $y_c=0$, nous avons $u=\sqrt{5}$, qui n'est pas rationnel. En reprenant les notations du II.3-b, le système d'équations est le suivant :

$$\begin{aligned} Q: & \quad (x_c-x_m)^2 + (y_c-y_m)^2 = (x_c-x_r)^2 + (y_c-y_r)^2 \\ L: & \quad \emptyset \end{aligned}$$

¹⁹Situations non-rationnelles avec construction d'un inverse dans [Chou-88] : 367, 368, 369, 370, 371, 372.

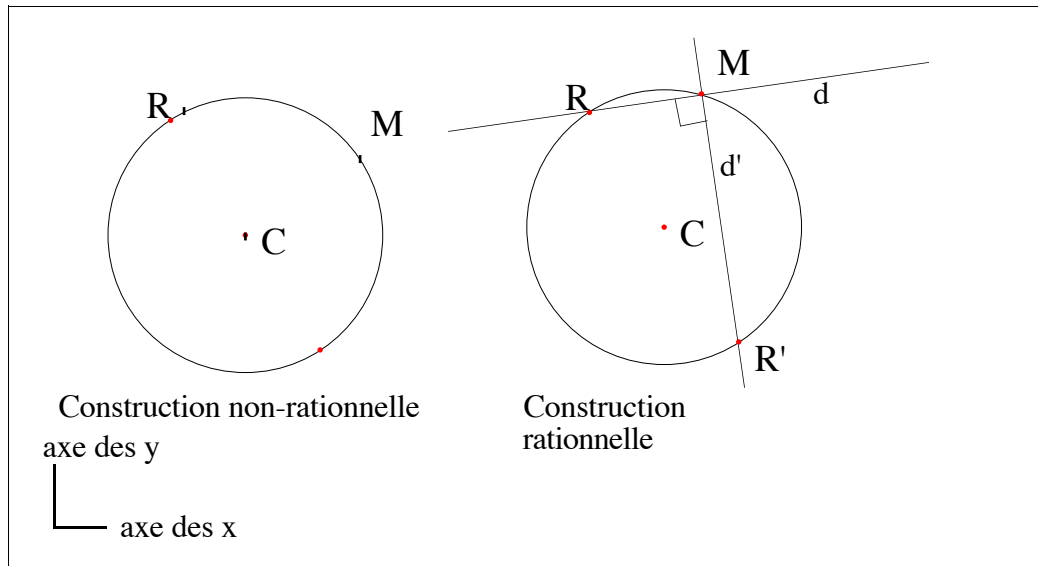


Figure III.2-a : Construction d'un point quelconque sur un cercle.

I : $x_c = \dots, y_c = \dots, x_r = \dots, y_r = \dots, x_m = \dots$ où les points de suspension sont des constantes rationnelles.

Nous proposons la méthode suivante pour rationaliser et linéariser cette construction : prendre le point M à l'intersection du cercle et d'une droite quelconque D rationnelle passant par le point R. Nous obtenons alors un point rationnel. C'est le projeté orthogonal du symétrique R' du point R par rapport au point C sur la droite quelconque. La figure III.2-a illustre cette construction, qui correspond au système d'équations suivant :

$$Q' : \emptyset$$

$$L' : x_r + x_{r'} = 2 \cdot x_c \quad (R' \text{ symétrique de } r \text{ par rapport à } C)$$

$$y_r + y_{r'} = 2 \cdot y_c \quad (R' \text{ symétrique de } r \text{ par rapport à } C)$$

$$y_r = x_r \cdot m_d + p_d \quad (R \text{ est sur la droite } d)$$

$$m_d \cdot m_{d'} = -1 \quad (d' \text{ perpendiculaire à } d)$$

$$y_{r'} = x_{r'} \cdot m_{d'} + p_{d'} \quad (R' \text{ est sur } d')$$

$$y = x \cdot m_d + p_d \quad (M \text{ est sur la droite } d)$$

$$y = x \cdot m_{d'} + p_{d'} \quad (M \text{ est sur la droite } d')$$

I' : $x_c = \dots, y_c = \dots, x_r = \dots, y_r = \dots, P_d = \dots$ où les points de suspension sont des constantes rationnelles.

Inverse par rapport au cercle.

La seconde situation concerne la construction de l'inverse M d'un point N par rapport à un cercle de centre C dont le carré du rayon est K.

Les méthodes usuelles opèrent comme à la figure III.2-b :

Si le point N est à l'intérieur du cercle, construire la droite D passant par le point N et le centre du cercle C. La perpendiculaire à D passant par le point N coupe le cercle au point P. Le

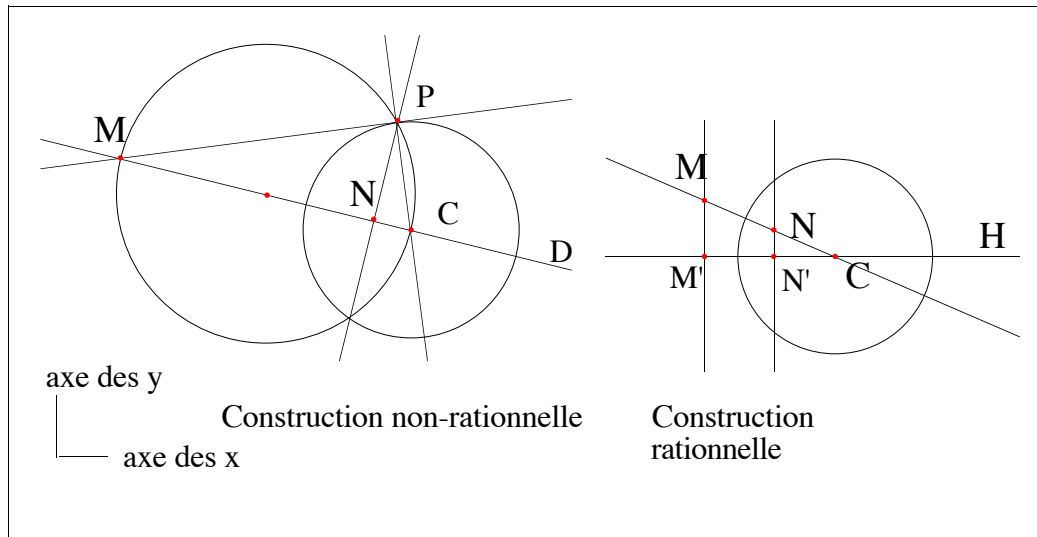


Figure III.2-b : Construction classique de l'inverse d'un point par rapport à un cercle.

point P n'est pas rationnel a priori. La tangente à P au cercle coupe D au point recherché M. Cette construction introduit avec le point P une extension algébrique supplémentaire.

Nous proposons la construction rationnelle suivante :

Considérons une droite horizontale H passant par le centre du cercle C. Le point N est projeté sur H en N'. Le point M' est obtenu sur H tels que $Y_M = Y_C$ et

$$X_{M'} = X_C + K / (X_N - X_C) |CN|^2.$$

Il est important de voir que $X_{M'}$ est rationnel. Ensuite, le point M recherché est obtenu à l'intersection de la verticale passant par M' et de D.

Bissectrice.

La troisième situation se trouve par exemple dans le théorème suivant :

Théorème 111 : Soient D l'intersection de l'une des bissectrices en A du triangle ABC avec le coté BC, et E une intersection de cette même bissectrice avec le cercle circonscrit à ABC, alors $AB.AC = AD.AE$.

Dans ce théorème, la construction de la bissectrice de l'angle \hat{A} dans le triangle ABC correspond à la situation qui nous intéresse. Nous avons vu que la construction d'une bissectrice introduit habituellement une nouvelle extension algébrique, cf. I-3. C'est le cas ici avec la construction proposée dans [Chou-88], qui reprend la forme du théorème :

- Soient A, B, C trois points quelconques. Construire la bissectrice en A de l'angle \hat{BAC} . Construire le point D à l'intersection de cette bissectrice avec le coté BC. Construire le cercle circonscrit au triangle ABC. Construire le point E comme intersection de cette bissectrice avec ce cercle.

La construction proposée n'est pas rationnelle : au début, la construction de la bissectrice introduit une extension algébrique.

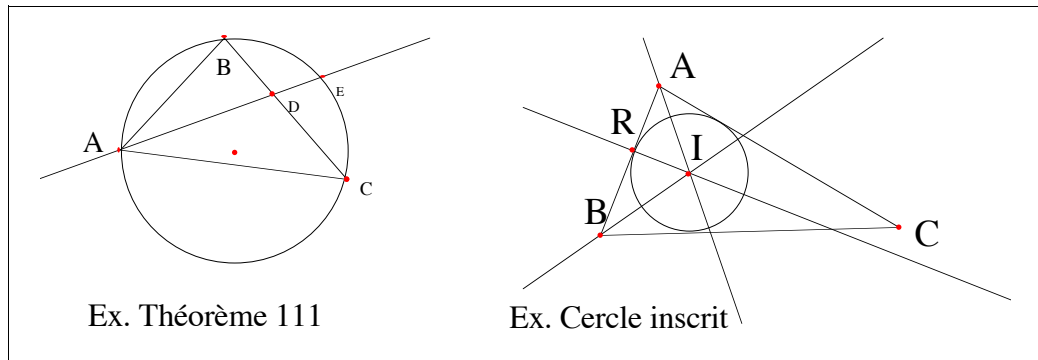


figure III.2-c : Situations avec bissectrices.

Une version rationnelle de cette situation est proposée en changeant l'ordre des constructions. La bissectrice de l'angle \hat{A} est donnée et l'angle \hat{A} est tracé par dédoublement :

- Soient A, B, E trois points quelconques. Construire la droite symétrique de (AB) respectivement à (AE). Construire le cercle circonscrit au triangle ABE. Prendre le point C à l'intersection de cette droite et du cercle. Construire le point D à l'intersection de (BC) et (AE).

Excepté la construction du point C, cette construction est linéaire. Ceci assure que la figure est rationnelle. Pour le point C, c'est un cas de construction rationnelle; comme le donne la section I.3-a car le point C est le symétrique de A par rapport au projeté du centre du cercle sur la droite symétrique. Par suite, l'ensemble de la figure est rationnel.

Notons l'existence d'un grand nombre de situations parmi les 512 où la modification que nous proposons sur les bissectrices est déjà effectuée. Ce sont les 12 situations²⁰ parmi les 13 où intervient le cercle inscrit dans un triangle. Ce cercle a pour centre le point de concours des bissectrices du triangle. La construction usuelle de la situation est la suivante :

- Étant donné un triangle ABC. Construire les bissectrices en A et B des angles \hat{BAC} et \hat{ABC} du triangle ABC. Prendre le point I à l'intersection de ces droites. Prendre le point R comme projeté de I sur AB. Le cercle de centre I et passant par le point R est le cercle recherché.

Cette construction du cercle inscrit dans ABC introduit 2 extensions, une pour chaque bissectrice. Pour des raisons qui ne sont pas données dans [Chou-88], cette construction usuelle est remplacée selon les principes que nous préconisons par la construction rationnelle suivante :

- Étant donnés A, B et I trois points arbitraires. Prendre le point C à l'intersection des droites symétriques de la droite AB par rapport aux droites AI et BI. Prendre le point R comme projeté de I sur AB. Construire le cercle de centre I et passant par le point R.

Cette construction est rationnelle.

Intersection avec un cercle.

²⁰Situation non-rationnelle de construction d'une bissectrice dans [Chou-88] avec construction par ABC : 172. Situations rationnelles avec construction par ABI : 38, 64, 78, 128, 154, 202, 252, 254, 319, 347, 380, 383.

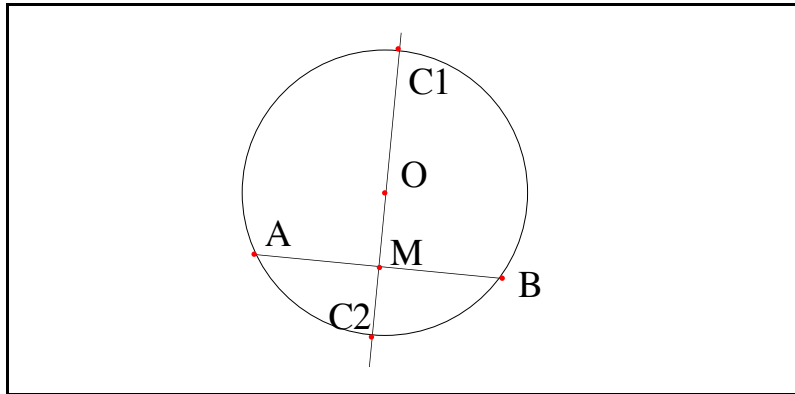


Figure III.2-d : Construction du milieu d'un arc.

La dernière situation qui nous intéresse apparaît dans le théorème suivant :

Théorème 108 : Soient C le milieu de l'arc AB d'un cercle de centre O, D un point sur ce cercle. (AB) coupe (CD) au point E. Montrer que $CA^2 = CE \cdot CD$.

Le début de la situation du théorème 108 nous intéresse: "Soit C le milieu de l'arc AB d'un cercle de centre O". La construction proposée par Chou est la suivante :

- Soient A et B deux points arbitraires. Construire M le milieu de [A,B]. Prendre un point O arbitraire sur la médiatrice de [A,B]. Construire C comme l'intersection de la droite (OM) et du cercle de centre O passant par A.

Cette construction représentée figure III.2-c. introduit un point C non-rationnel (par ex : si $A(-1,-1)$, $B(-1,1)$, $O(0,0)$, C vaut $(\sqrt{2},0)$ ou $(-\sqrt{2},0)$). Notons le fait que A, B, O étant fixés, il y a deux cas de figures pour C (C et son symétrique par rapport au point O). C'est un symptôme du caractère non-rationnel de la figure.

La solution rationnelle que nous proposons est la suivante :

- Soient A et B deux points arbitraires. Construire M le milieu de [A,B]. Prendre un point C rationnel arbitraire sur la médiatrice de [A,B]. Construire le point O comme centre du cercle circonscrit au triangle ABC.

La figure est ici rationnelle. D'ailleurs, il n'y a qu'un seul cas de figure (A, B, C étant fixés). La construction de O est rationnelle comme nous l'avons montré en I-2.

L'exemple présenté, recherche du milieu d'un arc, est relativement simple, et courant. Parfois, transformer une intersection introduisant des extensions algébriques en figure rationnelle est beaucoup plus difficile : la complexité de la recherche d'un ensemble de points de base qui rationalise la figure est combinatoire.

Résultats.

L'intérêt de cette recherche de constructions rationnelles est évident pour la résolution du problème de la vérification de contraintes redondantes. Puisque 98 % des problèmes proposés dans [Chou-88] deviennent rationnels, il suffit d'avoir une arithmétique rationnelle exacte pour

l'essentiel des cas. Cependant, il convient de rappeler que les 4 transformations proposées sont essentiellement des heuristiques qui sont apparues dans l'établissement des données de la section I.4-b. Parmi elles, les modifications locales 1 et 2 sont facilement automatisables. Les modifications globales 3 et 4 sont beaucoup plus difficiles à automatiser, car elles dépendent du contexte.

Plus modestement, nous observons que ces heuristiques aident la démonstration automatique : l'obtention d'une construction rationnelle s'apparente le plus souvent à l'obtention d'une construction pseudo-linéaire. C'est à dire où chaque propriété est représentée par un polynôme de degré partiel au maximum 1 en chacun de ces variables. Cette linéarisation des équations est d'une grande importance dans les méthodes de démonstration automatique. C'est probablement l'une des raisons qui expliquent le choix des constructions rationnelles pour les situations avec cercle inscrit dans [Chou-88].

Conclusion de ce chapitre :

L'apport d'une arithmétique exacte pour la vérification des contraintes redondantes est l'évidence qui sous-tend ce chapitre. Nous ajoutons qu'avec les travaux supplémentaires effectués sur la rationalisation des constructions géométriques, ces vérifications deviennent peu coûteuses.

Il apparaît de plus dans ce chapitre l'importance d'une double démarche pour résoudre les problèmes géométriques : d'une part algébrique, d'autre part géométrique. C'est une évidence qui transparaît dans de nombreux travaux. Mais la pluridisciplinarité que cela nécessite des efforts importants.

III-3. Démonstration numérique exacte de théorèmes de géométrie.

Introduction :

La démonstration automatique en géométrie a fait des progrès immenses depuis les travaux formels et algébriques de Wu Wentsun (84). En particulier à sa suite, les travaux de Schang-Ching Chou (88) ont mené à la réalisation pratique des premiers démonstrateurs. Ces démonstrateurs se sont appuyés sur des méthodes de démonstration algébriques. Les 512 théorèmes démontrés par la méthode de Schang-Ching Chou en sont la marque la plus convaincante.

A la même époque, parallèlement aux travaux de Schang-Ching Chou, et toujours en prolongeant le travail de Wu Wentsun, Jiwaie Hong étudie la complexité algorithmique des démonstrations automatiques. La méthode de Schang-Ching Chou est au moins exponentielle dans les pires cas. En 1986, Jiwaie Hong démontre la possibilité de démonstrations automatiques en géométrie dont la complexité en temps et en espace reste polynomiale. Il s'inspire d'une méthode de démonstration automatique de théorèmes de géométrie s'appuyant sur un algorithme numérique et une arithmétique approchée à précision contrôlée. En 1990, à la suite de Jiwaie Hong, une méthode numérique parallèle de démonstration voit le jour, utilisant toujours une arithmétique approchée grâce à Mike Deng, Jingzhong Zhang, et Lu Yang.

Cette série de méthodes de démonstration automatique en géométrie est décrite en section III.3-a.

Il est proposé dans ce chapitre, en section III.3-b, d'appliquer la méthode de démonstration automatique de Jiwaie Hong dans le cadre de nos travaux. Une nouvelle méthode de démonstration automatique est ainsi proposée. Elle s'appuie sur la vérification numérique avec une arithmétique exacte d'un système de contraintes redondantes. L'apport d'une arithmétique exacte comme nous la proposons, légitime à elle seule l'introduction de cette nouvelle méthode de démonstration automatique. Nous décrivons et employons de surcroît les résultats de nos recherches de constructions rationnelles du chapitre précédent III-2. Enfin, cette nouvelle méthode de démonstration automatique numérique s'inspire d'une nouvelle méthode de majoration qui est proposée. Tout aussi bien, cette méthode de démonstration peut être adaptée à la méthode de démonstration numérique parallèle proposée par Mike Deng, Jingzhong Zhang, et Lu Yang. Mais cette adaptation ne sera pas envisagée.

Finalement, quelques théorèmes sont démontrés pour permettre une comparaison entre les algorithmes de démonstrations automatiques issus de la méthode de Hong, section III.3-c.

L'ensemble de ce chapitre est placé sous la marque d'une double réserve, en ce qui concerne la pertinence et l'exhaustivité.

D'une part, la démonstration automatique ne fait pas partie des compétences que nous revendiquons. Nous effectuons ici une modeste incursion qui nous semble naturelle dans la suite des travaux entrepris au cours de la seconde partie. Nous reprenons quelques résultats en éludant volontairement quelques éléments secondaires. Nos contributions qui semblent apporter un peu de nouveauté sont mentionnées, en espérant l'indulgence des spécialistes en la matière.

D'autre part, la démonstration automatique ne constitue pas notre objectif premier. Il se peut que ce chapitre laisse le lecteur avec une envie d'en savoir plus. Le domaine est en effet très vaste, et demande un exposé beaucoup plus long pour atteindre une certaine exhaustivité. Nous avons tenté de proposer un exposé minimum pour produire un contenu compréhensible de lui-même. L'approfondissement des questions sous-jacentes que relève ce chapitre aurait produit une autre thèse, si l'orientation que nous avons voulu donner à ce travail avait été différente.

a) Méthodes de démonstrations automatiques.

Démonstrations automatiques algébriques : Wentsun Wu.

Les travaux de Wentsun Wu, poursuivis par ceux de Schang-Ching Chou, ont été déterminants dans les années 1980 pour la réalisation effective des premiers démonstrateurs automatiques en géométrie. Auparavant, les travaux théoriques de Tarsky qui montrèrent la possibilité d'un démonstrateur à partir d'une démarche algébrique et formelle, et les essais de démonstrations naturelles selon des méthodes combinatoires, n'ont pas abouti à un succès.

Rappelons, cf. I.1-a et I.1-b, qu'en 1988 Schang-Ching Chou donne dans sa thèse 512 exemples de théorèmes démontrés automatiquement en des temps convenables. La quasi totalité est constituée de théorèmes démontrés en moins d'une minute sur les ordinateurs de l'époque. Ces 512 théorèmes n'ont rien de trivial. En particuliers, ils comportent quelques théorèmes fameux qui sont restés longtemps des conjectures, et peut-être (selon Schang-Ching Chou) quelques nouveaux théorèmes. En effet, le démonstrateur de Chou est un puissant découvreur de théorèmes.

La principale critique que l'on peut apporter au démonstrateur initial de Chou vient de ce qu'il ne concerne qu'une géométrie non ordonnée. Mais les géométries euclidiennes et non euclidiennes, ordonnées ou non, sont aujourd'hui atteintes par certains démonstrateurs. Pour avoir un panorama plus large des travaux entrepris en démonstration automatique, nous renvoyons le lecteur à l'exposé [Wang-96]. En particulier, Dongming Wang cite les travaux de [Chou, Gao et Zhang-94] qui nous semble d'intérêt. Ils tendent vers la rédaction de preuves lisibles par l'homme.

La méthode de démonstration automatique en géométrie de Schang-Ching Chou est essentiellement algébrique. En résumé, elle peut être présentée ainsi :

Méthode de démonstration automatique de Schang-Ching Chou :

Étant donné un théorème à démontrer et une figure représentative, un système de coordonnées est choisi et les points M de la figure y sont représentés par des variables algébriques (X_M, Y_M) .

Par convention, les points constituant un ensemble de points de base pour la figure sont représentés par des variables dites "indépendantes", notées U_j . Ce sont des paramètres arbitraires de l'ensemble des figures possibles. Leur nombre correspond au nombre de degrés de liberté de la figure. Ils peuvent prendre toutes les valeurs possibles. Les autres points sont dit "dépendants", ils sont déterminés par les points indépendants. Ils sont représentés par des variables algébriques notées X_i .

Les hypothèses du théorème sont exprimées par des équations algébriques liant les points dépendants aux points indépendants :

$$H_k : f_k(u_1, \dots, u_i, x_1, \dots, x_j) = 0, (k=1, \dots, n)$$

La conclusion du théorème est également exprimée sous forme d'une équation algébrique :

$$C : T(u_1, \dots, u_i, x_1, \dots, x_j) = 0$$

La première étape de la démonstration est constituée par la triangularisation (selon les x_j) des équations H_k . L'emploi de l'algorithme de Wu-Ritt assure l'obtention d'un système d'hypothèses équivalentes exprimées sous forme triangulaire lorsque celui-ci existe, ou la terminaison de l'algorithme sinon. Essentiellement, cet algorithme agit par divisions algébriques, mais aussi parfois par factorisations. Dans la plupart des cas, les factorisations ne sont pas indispensables, mais elles peuvent éventuellement accélérer la démonstration. Après triangularisation, les hypothèses équivalentes s'expriment selon :

$$H'_k : F_k(u_1, \dots, u_i, x_1, \dots, x_k) = 0, (k=1, \dots, n)$$

La mise sous forme triangulaire fait apparaître un ensemble de conditions de non-dégénérescence (les facteurs du diviseur) que nous n'étudierons pas ici. Pour simplifier, nous supposons que les divisions sont licites, c'est à dire que les conditions de non-dégénérescence sont vérifiées. Elles correspondent à des cas de figures particuliers, qui ne sont pas étudiés. Le théorème sera donc vrai, dans le cas général.

La seconde étape de la démonstration consiste à effectuer les divisions successives de T par F_n, \dots, F_1 dans cet ordre. Autrement dit, la seconde étape effectue l'élimination successive de x_n, \dots, x_1 dans T , en utilisant les relations $F_n=0, \dots, F_1=0$. À la fin de la seconde étape, nous avons :

$$I(\dots).T(u_1, \dots, u_i, x_1, \dots, x_j) = R(u_1, \dots, u_i) + \sum F_k(u_1, \dots, u_i, x_1, \dots, x_k).Q_k(\dots)$$

où $I(\dots)$ représente les conditions de non-dégénérescence (les facteurs coefficients intervenant dans les divisions), $R(\dots)$ est le reste des divisions successives et les $Q_k(\dots)$ sont les quotients successifs. Comme les $F_k(\dots)$ sont nul dans le cas du théorème et que, dans le cas général, par convention $I(\dots)$ est non nul, il vient :

$$C: T(u_1, \dots, u_i, x_1, \dots, x_j) = 0 \iff R(u_1, \dots, u_i) = 0$$

Comme $R(\dots)$ est un polynôme en u_i , paramètres indépendants de la figure, il vient que le théorème est prouvé si et seulement si $R(\dots)$ est le polynôme identiquement nul. Sinon, pour certaines valeurs des paramètres u_i , on aurait $R(\dots) \neq 0$, et dans les cas de figure qui respectent les conditions de non-dégénérescence, ce serait un contre-exemple au théorème.

Par conséquent, les divisions successives de T par les F_i doivent donner un reste nul pour que le théorème soit vrai.

[[fin de l'exposé de la méthode]]

Notons au passage que dans le cas où le théorème est donné sous une forme constructive, la première étape (triangularisation) est immédiate. La seconde étape reste cependant coûteuse, la complexité est élevée.

Des méthodes similaires dans l'esprit s'appuient sur l'utilisation de bases de Gröbner pour déterminer la nullité du reste $R(\dots)$ sans effectuer les divisions algébriques explicitement. La complexité est exponentielle, les résultats pratiques sont comparables en temps.

Démonstrations automatiques numériques : Hong.

En même temps que Schang-Ching Chou aboutissait dans ses travaux pour trouver la juste formulation algébrique des travaux de Wu Wentsun; Jiawei Hong propose une adaptation numérique de ces mêmes travaux en vue d'obtenir un résultat sur la complexité d'un démonstrateur automatique. Il s'agit de vérifier numériquement, sur un exemple particulier choisi en fonction d'une majoration du reste $R(\dots)$, la nullité de ce polynôme. De ces deux approches, deux styles de démonstrateurs sont nés, algébriques et numériques. De ces deux démarches, celle qui a le plus de succès est la démarche algébrique de Chou : le nombre de travaux poursuivant ceux de Hong est relativement faible. Pourtant comme nous le verrons, le champ ouvert par Hong est très large.

L'inconvénient de la méthode de Hong vient peut-être de ce qu'il tient dans le rapprochement de deux résultats. Le premier résultat utilisé par Hong consiste en une méthode qui permet, à partir d'une évaluation numérique exacte, de décider de la nullité d'une expression algébrique quelconque. Le second résultat nécessaire à Hong est l'utilisation d'une arithmétique approchée pour effectuer des calculs numériques exacts. L'absence d'une arithmétique exacte pour les nombres constructibles apparaît ici. Ce sera, au paragraphe suivant, un point d'amélioration de notre méthode. En ce qui concerne l'utilisation d'arithmétiques approchées pour effectuer des calculs exacts, nous renvoyons à la section III.2-a.

Méthode de démonstration automatique de Jiawei Hong :

La méthode de démonstration automatique en géométrie de Jiawei Hong s'effectue en trois étapes.

La première étape de la démonstration coïncide avec celle de Shang-Ching Chou. Seulement, pour réduire le problème, Hong suppose que le théorème à démontrer, est proposé sous forme constructive. Par conséquent, comme nous l'avons signalé précédemment, la première étape de la démonstration, la triangularisation est quasi immédiate.

À la fin de la première étape, les deux méthodes se trouvent donc dans la même situation. Étant donnée une modélisation de la figure $M_i(U_{2i}, U_{2i+1})$ pour les points de base, et $M_j(X_{2j}, X_{2j+1})$ pour les points construits, les hypothèses et le théorème sont représentés sous forme d'un ensemble d'équations algébriques triangulaire :

$$H_k : F_k(u_1, \dots, u_i, x_1, \dots, x_k) = 0, \quad (k=1, \dots, n)$$

$$C : T(u_1, \dots, u_i, x_1, \dots, x_j) = 0$$

Il reste donc, comme pour la méthode de démonstration de Chou, à vérifier que le reste $R(\dots)$ des divisions successives de $T(\dots)$ par F_n, \dots, F_1 est identiquement nul. À la différence de la méthode précédente, ces divisions ne sont pas effectivement réalisées. La nullité de $R(\dots)$ est décidée par une méthode numérique selon un procédé en trois étapes.

La première étape (c'est à dire, la seconde de la méthode globale) consiste à majorer la forme du polynôme $R(\dots)$. Plusieurs formes de majorations sont possibles. Celle employée par Hong revient à déterminer pour chaque variable U_i le degré maximal que peut prendre un monôme en u_i dans $R(\dots)$. Elle est explicitée après l'exposé de la méthode. De plus, le coefficient maximal de $R(\dots)$ est estimé. Cette double majoration se fait par transposition des divisions successives dans cette structure de majorant.

La seconde étape consiste à déterminer des valeurs numériques particulières pour les u_i qui permettent, à partir d'une majoration de $R(\dots)$ en u_i , de prouver la nullité de $R(\dots)$. Elle s'effectue en utilisant la majoration de $R(\dots)$ à l'étape précédente. Un exemple est donné après l'exposé de la méthode.

Enfin, l'évaluation numérique constitue la troisième étape et conclut la preuve globale. Les valeurs numériques particulières des u_i sont utilisées pour calculer l'ensemble de la figure, c'est à dire les x_i . Si la propriété est vérifiée sur la figure, le théorème est vrai. Sinon, les u_i forment un contre exemple au théorème.

[[fin de l'exposé de la méthode]]

Donnons un exemple effectif de la première étape de la preuve numérique. Prenons une méthode simple d'estimation des degrés maximum. Pour effectuer la division de $P(\dots)$ par $Q(\dots)$ selon X_k :

- supposons que $P(\dots)$ possède une estimation : X_i de degré d_i au maximum,
- supposons que $Q(\dots)$ soit parfaitement connu : par exemple Q est linéaire en X_k avec pour monômes des cofacteurs maximum $X_1^{a_1}, \dots, X_j^{a_j}$ dans les monômes où apparaît X_k , et $X_1^{b_1}, \dots, X_j^{b_j}$ dans les autres monômes,
- alors le reste de la division de $P(\dots)$ par $Q(\dots)$ selon X_k donne un reste avec l'estimation : le degré maximum de X_i est $d_i' = d_i + \max(a_i, b_i)$.

Donnons un exemple de la seconde étape de la preuve numérique. Si $R(\dots)$ est un polynôme en une seule variable u_1 de degré au plus d_1 , soit il est identiquement nul, soit il possède au plus d_1 racines. L'évaluation de $R(\dots)$ en (d_1+1) points distincts permet d'affirmer : s'il est (d_1+1) fois nul, alors il est identiquement nul; sinon les évaluations non nulles donnent des contre-exemples au théorème.

Dans le cas où $R(\dots)$ est un polynôme en deux indéterminées u_1, u_2 , dont l'estimation est u_i , de degré d_i au maximum ($i=1,2$), le polynôme $R(\dots)$ peut être vu comme un polynôme en u_2 , avec pour coefficients des polynômes R_1, \dots, R_{d_1} en u_1 . Pour une valeur particulière de u_1 , le polynôme $R(\dots)$ est un polynôme R' en u_2 . D'après ce qui précède, d_2+1 évaluations permettent de dire si R' est identiquement nul ou non. Dans le cas où R' est identiquement nul, cela signifie que les polynômes R_1, \dots, R_{d_1} sont nuls au point particulier u_1 choisi. Ainsi, si pour d_1+1 valeurs particulières de u_1 , l'évaluation de d_2+1 valeurs particulières de u_2 montre la nullité des R_1, \dots, R_{d_1} aux points particuliers de u_1 , alors par ce qui précède, les R_1, \dots, R_{d_1} sont identiquement nuls. Par suite, $R(\dots)$ est identiquement nul. Pour résumer, il faut donc évaluer numériquement $R(\dots)$ sur un ensemble de $(d_1+1).(d_2+1)$ points pour avoir la preuve de la nullité formelle de $R(\dots)$.

Dans le cas général où $R(\dots)$ est un polynôme en n variables, un ensemble E de $\prod (d_i+1)$ points distincts est nécessaire pour prouver la nullité de $R(\dots)$. Cet ensemble est pris en prenant d_1+1 valeurs particulières pour u_1 , et pour ces valeurs d_2+1 valeurs particulières de u_2 , et pour ces valeurs d_3+1 valeurs particulières de u_3 , et ainsi de suite, et pour ces valeurs d_n+1 valeurs particulières de u_n . Par suite, si l'ensemble des évaluations numériques de $R(\dots)$ sur les points de E donne 0, alors $R(\dots)$ est nul, sinon un contre exemple au théorème est trouvé.

Il apparaît clairement que la méthode proposée par Hong est très générale. En particulier, nous avons pris la liberté de la présenter dans un cas particulier plus proche de celui présenté dans [Deng, Yang et Zhang-90].

En particulier, trois points peuvent être adaptés,

- 1- la méthode de majoration,
- 2- la transposition de la preuve de la nullité de $R(\dots)$ à une recherche de cas positifs numériques particuliers et enfin
- 3- les méthodes numériques d'évaluation de chaque cas positif numérique. Nous apporterons au paragraphe II.3-b suivant notre propre solution sur ces trois points.

Pour reprendre ces trois éléments, nous citons au passage que la méthode de démonstration choisie par Hong repose

- 1) sur une majoration à partir des degrés et coefficients maximum,
- 2) sur une transposition qui s'effectue en recherchant un point unique particulier qui constitue un exemple positif et enfin
- 3) sur une évaluation numérique qui s'effectue à partir d'une arithmétique à précision contrôlée.

La méthode de [Deng, Yang et Zhang-90] présentée ici repose

- 1) sur une simple majoration des degrés partiels maximum,
- 2) sur l'élaboration d'un ensemble de cas positifs.

Quant à la méthode d'évaluation numérique, négligée dans l'article, elle suppose que l'arithmétique usuelle des ordinateurs est suffisante. Cette méthode, avec son grand nombre d'évaluations identiques sur des points différents, est orientée vers l'utilisation d'ordinateurs parallèles.

En conclusion de ce paragraphe : autant de méthodes de majoration, autant de méthodes de transposition, et autant de méthodes d'évaluation numérique, donnent autant de méthodes de démonstration différentes.

b) Méthodologie de démonstration automatique par évaluation numérique exacte.

Notre méthodologie générale est simple :

Méthode de démonstration automatique :

- 1- choix des variables indépendantes et de variables dépendantes,
 - 2- rationalisation heuristique de la proposition à démontrer,
 - 3- majoration de la forme du reste $R(\dots)$,
 - 4- choix d'un cas particulier qui effectue la transposition preuve numérique/preuve formelle,
 - 5- évaluation numérique avec une arithmétique exacte.
- [[fin de l'exposé]]

Les points essentiels que nous développons dans cette section sont les trois derniers de la méthode de démonstration. Nous citons au passage un troisième exemple de méthode de démonstration défini lors d'un travail effectué en compagnie de Patrick Lixi. La méthode de majoration prend en compte 4 paramètres : le coefficient maximum et le degré maximum pour l'ensemble des variables, et selon la variable à éliminer. L'évaluation numérique se déroule sur un exemple unique avec une arithmétique exacte, cf. [Lixi-96].

Majoration de la forme du reste $R(\dots)$.

Notre objectif est de donner une méthode de majoration paramétrique avec un paramètre représentant le degré de précision du majorant.

Nous avons pensé prendre pour paramètre k , la profondeur ou la largeur de l'arbre représentant le polynôme $P(X_1, \dots, X_n)$, ou la complexité $O(n^k)$ de l'algorithme d'évaluation. Ce choix n'a pas donné de bons résultats. Dans l'idéal, nous aimerions que le cas où la valeur du paramètre tend vers l'infini représente une estimation exacte, et que le cas où ce paramètre vaut 0 s'apparente à la méthode de Hong. Éventuellement, nous aimerions de plus faire coïncider le cas où le paramètre vaut 1 et une méthode semblable à celle de [Zhang/Yang/Deng-90]. Malheureusement, nous n'avons pas atteint cet objectif.

Nous proposons la classification informelle des majorations suivante. Supposons que le polynôme $P(X_1, \dots, X_n)$ est représenté par l'arbre Prolog \mathbf{P} suivant : $P_{X_1}(a_0.X_1^0.P_{0X_2}(\dots), \dots, a_n.X_1^n.P_{nX_2}(\dots))$. Une méthode de majoration sera définie par un paramètre \mathbf{K} représenté par un arbre semblable en partie avec l'arbre \mathbf{P} . Les noeuds de \mathbf{K} donnent le nombre d'informations conservées sur le sous-arbre de \mathbf{P} correspondant à ce noeud. Selon cette classification, nous retrouvons les procédés de majoration utilisés par Hong et [Deng, Yang et Zhang-90] :

- Selon la méthode que nous nommons $\mathbf{K}(1)$, le majorant du polynôme P est représenté par un seul facteur : le degré maximum en une indéterminée quelconque. C'est la méthode de Hong. La complexité de l'estimation est en $O(n)$.

- Selon la méthode que nous nommons $\mathbf{K}(1(1(1\dots)))$, le majorant du polynôme P est donné par n facteurs : le degré maximum de chaque indéterminée X_i . C'est la méthode de Zhang/Yang/Deng. La complexité de l'estimation est en $O(n^2)$.

Nous ajoutons deux majorations possibles :

- Pour la méthode que nous nommons $\mathbf{K}(1(1(1(1\dots))), 1(1(1(1(1\dots))), 1(1(1(1(1(1\dots))), \dots))$, le majorant du polynôme P est donné par n^2 facteurs. Pour chaque indéterminée X_i , les polynômes cofacteurs sont représentés par la méthode $\mathbf{K}(1(1(1(1\dots))))$.

- Enfin, pour la méthode que nous nommons $\mathbf{K}(d_1(d_2(\dots), d_3(\dots), \dots), d_2(d_3(\dots), \dots), \dots))$, l'ensemble de P est représenté.

Exemple : Quel est le majorant du polynôme $X*Y^2+Z^3$?

- Pour $\mathbf{K}(1)$, le majorant du polynôme $X*Y^2+Z^3$ est représenté par le coefficient 3. Les polynômes majorés comme ce polynôme par le coefficient 3 sont ceux de la forme (les k sont des constantes arbitraires distinctes) :

$$\begin{aligned} X^3 & \left(Y^3(k.Z^3+k.Z^2+k.Z+k) + Y^2(k.Z^3+k.Z^2+k.Z+k) + Y(k.Z^3+k.Z^2+k.Z+k) + \right. \\ & k.Z^3+k.Z^2+k.Z+k \left. \right) + X^2 \left(Y^3(k.Z^3+k.Z^2+k.Z+k) + Y^2(k.Z^3+k.Z^2+k.Z+k) + \right. \\ & Y(k.Z^3+k.Z^2+k.Z+k) + k.Z^3+k.Z^2+k.Z+k \left. \right) + X \left(Y^3(k.Z^3+k.Z^2+k.Z+k) + \right. \\ & Y^2(k.Z^3+k.Z^2+k.Z+k) + Y(k.Z^3+k.Z^2+k.Z+k) + k.Z^3+k.Z^2+k.Z+k \left. \right) \\ & + Y^3(k.Z^3+k.Z^2+k.Z+k) + Y^2(k.Z^3+k.Z^2+k.Z+k) + Y(k.Z^3+k.Z^2+k.Z+k) \\ & + k.Z^3+k.Z^2+k.Z+k. \end{aligned}$$

- Pour $\mathbf{K}(1(1(1(1\dots))))$, le majorant du polynôme $X*Y^2+Z^3$ est représenté par le triplet (X^1, Y^2, Z^3) . Les polynômes majorés comme ce polynôme par le triplet (X^1, Y^2, Z^3) sont ceux de la forme (les k sont des constantes arbitraires distinctes) :

$$X \left(Y^2(k.Z^3+k.Z^2+k.Z+k) + Y(k.Z^3+k.Z^2+k.Z+k) + k.Z^3+k.Z^2+k.Z+k \right)$$

$$+ Y^2(k.Z^3+k.Z^2+k.Z+k) + Y(k.Z^3+k.Z^2+k.Z+k) + k.Z^3+k.Z^2+k.Z+k.$$

- Pour $\mathbf{K}(1(1(1(\dots))),1(1(1(\dots))),1(1(1(\dots))),\dots)$, le majorant du polynôme : $X*Y^2+Z^3$, est représenté par 9 facteurs que nous résumons à $(X(Y^2),Z^3)$. Les polynômes majorés comme ce polynôme sont ceux de la forme (les k sont des constantes arbitraires distinctes) :

$$X(k.Y^2+k.Y+k)+k.Z^3+k.Z^2+k.Z+k$$

Choix d'un cas particulier unique pour effectuer la transposition de la preuve algébrique à une évaluation numérique.

Dans la section III.3-a, une méthode de preuve simple par évaluation numérique multiple est présentée. Nous définissons dans cette sous-section une méthode de preuve par évaluation numérique unique plus adaptée à nos préoccupations. D'autres méthodes de preuves numériques peuvent être trouvées dans [Davis-77].

L'idée de la méthode est donnée pour les polynômes en une indéterminée :

- Les racines entières non nulles d'un polynôme P en une variable X avec des coefficients entiers inférieurs à m sont inférieures à m+1.

[[En effet si r est une racine entière non nulle de $P=\sum p_i.X^i$.

Alors $P(r) = 0$ entraîne $\sum p_i r^i = 0$. Donc $r(\sum p_i r^{i-1}) = -p_0$.

Par suite, r divise p_0 , donc r est inférieur à m+1.]]

- Par suite, l'évaluation de P en $X=r+1$ donne la nullité formelle de P si l'évaluation numérique est nulle.

[[Le point précédent donne $((P \text{ non nul}) \Leftrightarrow (P(r+1) \neq 0))$, ce qui donne le point présent.]]

La méthode dans le cas général est la suivante :

- soit $P(X_1,\dots,X_n)$ un polynôme en X_1,\dots,X_n de degré maximum d_i en X_i , et de coefficient maximum m. C'est à dire, dans une décomposition $P(X_1,\dots,X_n)=\sum P_i(X_2,\dots,X_n).X_1^i$, avec $P_i(X_2,\dots,X_n)=\sum P_{i,j}(X_3,\dots,X_n).X_{n-1}^j$, et ainsi de suite, d_i est le maximum des degrés des polynômes Pe_{1,\dots,e_i} .

- L'évaluation en $K_n=m+1$ donne : si $Pe_{1,\dots,e_{n-1}}(K_n) = 0$, alors $Pe_{1,\dots,e_{n-1}}(X_n)$ est formellement nul, c'est à dire les Pe_{1,\dots,e_n} sont tous nuls.

- Pour une évaluation au point $K_n=m+1$, les coefficients $Pe_{1,\dots,e_{n-1}}(K_n)$ du polynôme $Pe_{1,\dots,e_{n-2}}$ en X_{n-1} ont pour valeur maximale $(1-K_n^{d_n})/(1-K_n)$. Les racines de $Pe_{1,\dots,e_{n-2}}$ sont donc inférieures à cette valeur maximale + 1.

- Par suite, si l'évaluation numérique en $K_{n-1}=1+(1-K_n^{d_n})/(1-K_n)$, $K_n=m+1$, donne $Pe_{1,\dots,e_{n-2}}(K_{n-1},K_n) = 0$, alors le polynôme $Pe_{1,\dots,e_{n-2}}(X_{n-1},K_n)$ est nul; donc les coefficients $Pe_{1,\dots,e_{n-1}}(K_n)$ sont nuls et par conséquent, les Pe_{1,\dots,e_n} sont nuls aussi.

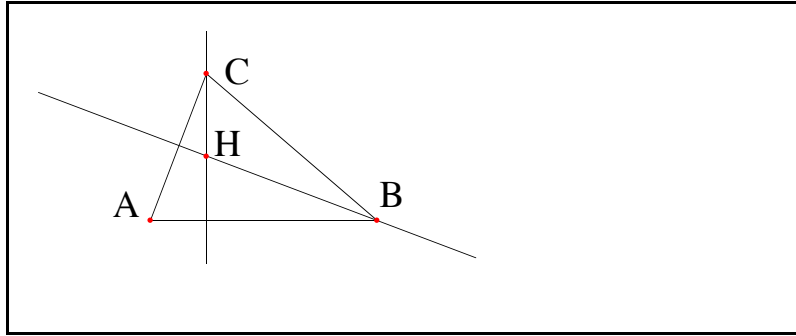


Figure III.3-a : Théorème de l'orthocentre.

- Ainsi de suite, pour une évaluation en $K_1=1+(1-K_2^{d_2})/(1-K_2), \dots, K_{n-1}=1+(1-K_n^{d_n})/(1-K_n), K_n=m+1$, l'évaluation numérique $P(K_1, \dots, K_n)=0$ donne la nullité des coefficients $Pe_1(K_2, \dots, K_n)$ et ainsi de suite jusqu'à ce que la preuve que les Pe_1, \dots, e_n sont tous nuls, c'est à dire P est identiquement nul.

c) Vérification de contraintes numériques redondantes.

La méthode de démonstration numérique se termine par la vérification d'un système de contraintes redondantes. Cela se fait naturellement en utilisant l'arithmétique exacte de la partie II. C'est l'objet du chapitre précédent III-2. Nous ajoutons ici brièvement quelques points intéressants qui apparaissent en appliquant les résultats du III-2.

- La section III.2-b montre que les transformations heuristiques permettent d'avoir le plus souvent des constructions rationnelles et linéaires : la linéarité des polynômes représentant les propriétés de la figure permet une majoration meilleure du polynôme $R(\dots)$.

- L'existence de l'arithmétique exacte permet d'éviter l'emploi d'une arithmétique approchée qui obscurcit les travaux de Hong.

d) Démonstration de théorèmes significatifs.

Nous commençons par deux théorèmes relativement simples et usités dans l'enseignement de la géométrie en classe de secondaire. Nous poursuivons avec deux autres théorèmes généraux. Chaque preuve comporte l'étude comparative des diverses formes de majoration de $R(\dots)$. Les démonstrations complètes de trois premiers théorèmes sont en annexe 6. Elles sont malheureusement assez illisibles par l'homme, comme la plupart des démonstrations algébriques.

Théorème de l'orthocentre

Théorème de l'orthocentre : Soient un triangle ABC et H le point de concours des hauteurs issues de C et B, alors H est sur la hauteur issue de A.

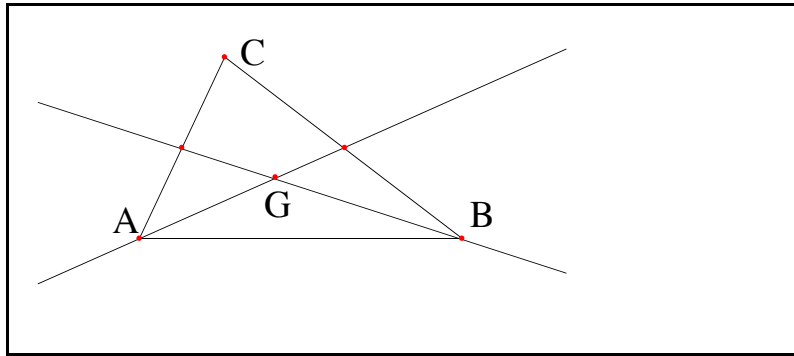


Figure III.3-b : Théorème du centroïde.

[[preuve]] Modélisation : soient A (0,0), B(U₁,0), C(U₂,U₃); le point H(X₁,X₂) est construit selon :

$$(CH \perp AB) : X_1 - U_2 = 0, \quad (F_1)$$

$$(BH \perp AC) : X_1 \cdot U_2 - U_1 \cdot U_2 + X_2 \cdot U_3 = 0. \quad (F_2)$$

À démontrer :

$$(AH \perp BC) : X_1 \cdot U_1 - X_1 \cdot U_2 + X_2 \cdot U_3 = 0. \quad (T)$$

Le système F₁, F₂ est triangulaire en X₁, X₂. L'élimination des variables se fait selon X₂ avec F₂, puis selon X₁ avec F₁.

Selon la majoration **K**(1), nous avons le point suivant à évaluer :

$$U_1=7, U_2=(U_1^5-1)/(U_1-1)+1, U_3=(U_2^5-1)/(U_2-1)+1.$$

Selon la majoration **K**(1(1(1(...))))), nous avons le point suivant à évaluer :

$$U_1=7, U_3=(U_1^3-1)/(U_1-1)+1, U_2=(U_3^3-1)/(U_3-1)+1.$$

Selon la majoration **K**(1(1(1(...))),1(1(1(...))),1(1(1(...))),...), nous avons le point suivant à évaluer :

$$U_1=5, U_3=(U_1^2-1)/(U_1-1)+1, U_2=(U_3^2-1)/(U_3-1)+1.$$

La dernière estimation donne la preuve du théorème dans le cas général si et seulement si la figure U₁=5, U₃=7, U₂=9 possède la propriété.

Comme la figure A(0,0), B(5,0), C(9,7) donne H(9,-48/7) et que H est sur la hauteur issue de A dans cette figure, le théorème est vrai.

[[fin de la preuve]]

Théorème du centroïde.

Théorème du centroïde : Soient un triangle ABC et G le point de concours des deux médianes issues de A et B; alors G est sur la médiane issue de C.

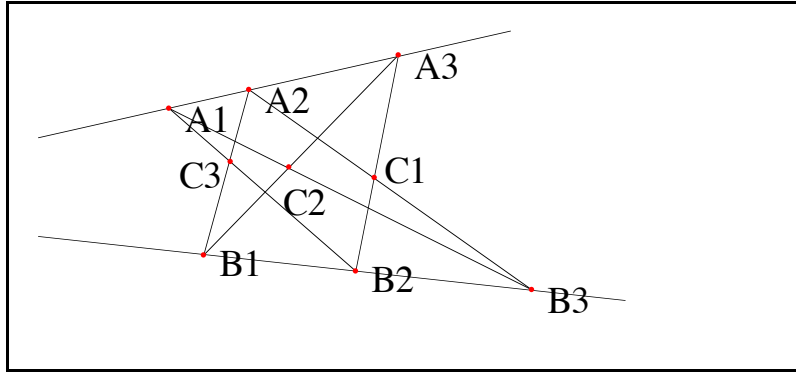


Figure III.3.c : Théorème de Pappus.

[[preuve]] Modélisation : Soient $A(0,0)$, $B(0,U_1)$, $C(U_2,U_3)$; le point $G(X_1,X_2)$ est construit selon :

$$H \text{ sur la médiane issue de } B : X_2 \cdot U_2 - X_1 \cdot U_3 + 2 \cdot X_1 \cdot U_1 - U_1 \cdot U_2 = 0, \quad (F_1)$$

$$H \text{ sur la médiane issue de } A : X_2 \cdot U_2 - X_1 \cdot U_1 + X_1 \cdot U_3 = 0. \quad (F_2)$$

Par triangularisation locale (élimination de X_2 dans F_1 à partir de F_2),

F_1 est remplacé par : $-3 \cdot X_1 + U_2 = 0$ (F_1'). Les conditions de non-dégénérescence imposent $U_1 \neq 0$

À démontrer :

$$H \text{ sur la médiane issue de } C : 2 \cdot X_2 \cdot U_2 - 2 \cdot X_1 \cdot U_3 - X_1 \cdot U_1 - U_2 \cdot U_1 = 0. \quad (T)$$

Le système F_1', F_2 est triangulaire en X_1, X_2 . L'élimination des variables se fait selon X_2 avec F_2 , puis selon X_1 avec F_1' .

Selon la majoration $\mathbf{K}(1)$, nous avons le point suivant à évaluer :

$$U_1=109, U_2=(U_1^5-1)/(U_1-1)+1, U_3=(U_2^5-1)/(U_2-1)+1.$$

Selon la majoration $\mathbf{K}(1(1(1\dots)))$, nous avons le point suivant à évaluer :

$$U_1=61, U_3=(U_1^3-1)/(U_1-1)+1, U_2=(U_3^3-1)/(U_2-1)+1.$$

Selon la majoration $\mathbf{K}(1(1(1(\dots))),1(1(1(\dots))),1(1(1(\dots))),\dots)$, nous avons le point suivant à évaluer :

$$U_1=17, U_3=(U_1^2-1)/(U_1-1)+1, U_2=(U_3^2-1)/(U_3-1)+1.$$

La dernière estimation donne la preuve du théorème dans le cas général si et seulement si la figure $U_1=17, U_2=21, U_3=19$ possède la propriété.

Comme la figure $A(0,0), B(0,17), C(21,19)$ donne $G(21/3,36)$ et que G est sur la médiane issue de C dans cette figure, le théorème est vrai.

[[fin de la preuve]]

Théorème de Pappus.

Le théorème de Pappus est bien connu pour des raisons historiques et géométriques. C'est un théorème central en géométrie projective.

Par rapport aux 512 théorèmes de géométrie récoltés par Chou, il est tout à fait particulier, car il fait intervenir 6 points. La plupart des 512 théorèmes de cette collection concernent un triangle, donc trois points.

Du point de vue de la démonstration automatique, c'est un théorème qui fait donc intervenir deux fois plus de variables algébriques dites "paramètres" que la majorité des théorèmes courants. Par suite, la complexité de la démonstration est bien plus grande, toutes choses étant égales par ailleurs.

Dans le cas d'une démonstration par multi-instanciations, l'ajout d'une variable "paramètre" multiplie le nombre de cas à étudier par le degré de cette variable dans $R(\dots)$. Par suite, quand il y a doublement du nombre de variables "paramètre", le nombre de points à évaluer est élevé au carré, toutes choses étant égales par ailleurs.

Dans le cas d'une démonstration par une seule instanciation, la dernière constante à considérer sera de l'ordre de la puissance $n^{\text{ième}}$ de la précédente, si n est le degré de cette variable dans $R(\dots)$. Quand il y a doublement du nombre de variables "paramètre", la complexité augmente donc de plusieurs ordres.

Théorème de Pappus : Soient trois points A_1, A_2, A_3 alignés, trois autres points B_1, B_2, B_3 alignés, alors les points $C_1 = (A_2B_3) (A_3B_2)$, $C_2 = (A_1B_3) (A_3B_1)$ et $C_3 = (A_1B_2) (A_2B_1)$ sont alignés.

[[preuve]] Modélisation : Soient $A_i(0, U_i)$, $B_i(V_i, 0)$; les points $C_i(X_i, Y_i)$ sont construits selon :

$$C_3 \text{ sur } (A_1B_2) : V_2 \cdot X_3 + U_1 \cdot Y_3 - U_1 \cdot V_2 = 0, \quad (F_1)$$

$$C_3 \text{ sur } (A_2B_1) : V_1 \cdot X_3 + U_2 \cdot Y_3 - U_2 \cdot V_1 = 0. \quad (F_2)$$

Par triangularisation locale (élimination de X_3 dans F_2 à partir de F_1),

$$F_2 \text{ est remplacé par : } -V_1 \cdot U_1 \cdot Y_3 + V_1 \cdot U_1 \cdot V_2 + V_2 \cdot U_2 \cdot Y_3 - V_2 \cdot U_2 \cdot V_1 = 0. \quad (F_2')$$

$$C_2 \text{ sur } (A_1B_3) : V_3 \cdot X_2 + U_1 \cdot Y_2 - U_1 \cdot V_3 = 0, \quad (F_3)$$

$$C_2 \text{ sur } (A_3B_1) : V_1 \cdot X_2 + U_3 \cdot Y_2 - U_3 \cdot V_1 = 0. \quad (F_4)$$

Par triangularisation locale (élimination de X_2 dans F_4 à partir de F_3),

$$F_4 \text{ est remplacé par : } -V_1 \cdot U_1 \cdot Y_2 + V_1 \cdot U_1 \cdot V_3 + V_3 \cdot U_3 \cdot Y_2 - V_3 \cdot U_3 \cdot V_1 = 0. \quad (F_4')$$

$$C_1 \text{ sur } (A_2B_3) : V_3 \cdot X_1 + U_2 \cdot Y_1 - U_2 \cdot V_3 = 0, \quad (F_5)$$

$$C_1 \text{ sur } (A_3B_2) : V_2 \cdot X_1 + U_3 \cdot Y_1 - U_3 \cdot V_2 = 0. \quad (F_6)$$

Par triangularisation locale (élimination de X_1 dans F_6 à partir de F_5),

$$F_6 \text{ est remplacé par : } -V_2 \cdot U_2 \cdot Y_1 + V_2 \cdot U_2 \cdot V_3 + V_3 \cdot U_3 \cdot Y_1 - V_3 \cdot U_3 \cdot V_2 = 0. \quad (F_6')$$

À démontrer :

$$C_1, C_2, C_3 \text{ alignés : } X_1 \cdot Y_2 + Y_1 \cdot X_3 + X_2 \cdot Y_3 - X_3 \cdot Y_2 - Y_3 \cdot X_1 - X_2 \cdot Y_1 = 0. \quad (T)$$

Le système $F_1, F_2', F_3, F_4', F_5, F_6'$ est triangulaire en $X_3, Y_3, X_2, Y_2, X_1, Y_1$. L'élimination des variables se fait selon X_3 avec F_1 , puis selon Y_3 avec F_2' , puis selon X_2 avec F_3 , puis selon Y_2 avec F_4' , puis enfin selon X_1 avec F_5 , et selon Y_1 avec F_6' .

Selon la majoration $\mathbf{K}(1)$, nous avons le point suivant à évaluer :

$$U_1 = 3^{21} 2^{44} + 1, \quad U_2 = (U_1^{65} - 1) / (U_1 - 1) + 1, \quad U_3 = (U_2^{65} - 1) / (U_2 - 1) + 1,$$

$$V_1 = (U_3^{65} - 1) / (U_3 - 1) + 1, \quad V_2 = (V_1^{65} - 1) / (V_1 - 1) + 1, \quad V_3 = (V_2^{65} - 1) / (V_2 - 1) + 1.$$

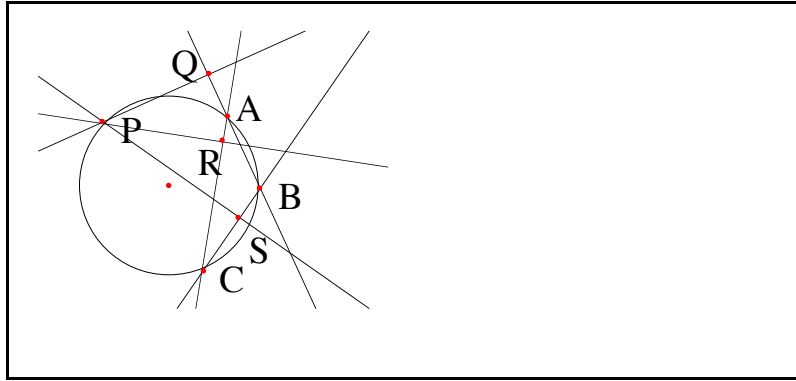


Figure III-2.d : Théorème de Simson.

Selon la majoration $\mathbf{K}(1(1(1\dots)))$, nous avons le point suivant à évaluer :

$$U_3=1025, V_1=(U_3^5-1)/(U_3-1)+1, U_2=(V_1^5-1)/(V_1-1)+1, \\ V_2=(U_2^6-1)/(U_2-1)+1, U_1=(V_2^6-1)/(V_2-1)+1, V_3=(U_1^7-1)/(U_1-1)+1.$$

Selon la majoration $\mathbf{K}(1(1(1(\dots))),1(1(1(\dots))),1(1(1(\dots))),\dots)$, nous avons le point suivant à évaluer :

$$U_3=65, V_1=(U_3^3-1)/(U_3-1)+1, U_2=(V_1^3-1)/(V_1-1)+1, \\ V_2=(U_2^4-1)/(U_2-1)+1, U_1=(V_2^4-1)/(V_2-1)+1, V_3=(U_1^4-1)/(U_1-1)+1.$$

Notes à propos des méthodes de majoration :

- La dernière majoration correspond aux polynômes en $U_3^2V_1^2U_2^3V_2^3U_1^3V_3^4$ avec pour coefficient maximum : 64.

- La meilleure majoration possible est obtenue d'après les divisions effectives : c'est un polynôme en $U_3^2V_1^2U_2^2V_2^3U_1^2V_3^4$ avec pour coefficient maximum 1.

- Dans le cas de presque-divisions, c'est à dire où les soustractions sont transformées en additions de telle sorte que les associations entre monômes soient toujours additives et jamais destructives, on obtient un polynôme en $U_3^2V_1^2U_2^3V_2^3U_1^3V_3^4$. avec pour coefficient maximum 8.

La majoration obtenue est donc très bonne du point de vue des degrés partiels et moins bonne pour les coefficients maximum.

La dernière estimation donne la preuve du théorème dans le cas général si et seulement si la figure $U_3=65, V_1=(U_3^3-1)/(U_3-1)+1, U_2=(V_1^3-1)/(V_1-1)+1$ possède la propriété. L'évaluation de la figure définit par ces paramètres montre que les points C_1, C_2, C_3 sont alignés sur cette figure. Par suite le théorème est vrai. Cette évaluation s'est effectuée en moins de 4 seconde sur un Macintosh Quadra 700.

[[fin de la preuve]]

Théorème de Simson.

Le théorème de Simson est intéressant car il comporte nécessairement des points sur un cercle, et permet donc de profiter de nos rationalisations.

Théorème de Simson : Soit P un point sur le cercle circonscrit au triangle ABC. Soient R, S, et T les projections orthogonales de P sur BA, CA, et AB; alors les points R, S et T sont alignés.

[[preuve]]

Modélisation (conformément aux heuristiques du III-2 pour les points sur cercles) : Le cercle est celui passant par O(0,0), et P(1,0), A(X₁,Y₁), B(X₂,Y₂), C(X₃,Y₃) sont définis à l'aide de trois paramètres U₁, U₂, U₃ :

$$(U_1^2+1).X_1-U_1^2=0, \quad (F_1)$$

$$X_1-U_1.Y_1=0, \quad (F_2)$$

$$(U_2^2+1).X_2-U_2^2=0, \quad (F_3)$$

$$X_2-U_2.Y_2=0, \quad (F_4)$$

$$(U_3^2+1).X_3-U_3^2=0, \quad (F_5)$$

$$X_3-U_3.Y_3=0. \quad (F_6)$$

Les points R, S, T ainsi que la suite sont définis comme dans [Deng, Yang et Zhang-90] (exceptée bien sûr la modélisation des points sur le cercle qui est comme il vient d'être écrit).

$$(1-Z_1.X_2-(1-Z_1).X_3)*(X_2-X_3)-(Z_1.Y_2+(1-Z_1).Y_3)*(Y_2-Y_3)=0 \quad (F_7)$$

$$(1-Z_2.X_3-(1-Z_2).X_1)*(X_3-X_1)-(Z_2.Y_3+(1-Z_2).Y_1)*(Y_3-Y_1)=0 \quad (F_8)$$

$$(1-Z_3.X_1-(1-Z_3).X_2)*(X_1-X_2)-(Z_3.Y_1+(1-Z_3).Y_2)*(Y_1-Y_2)=0 \quad (F_9)$$

La propriété à vérifier : R, S, T sont alignés :

$$(Z_1.X_2+(1-Z_1).X_3)*(Z_2.Y_3+(1-Z_2).Y_1-Z_3.Y_1-(1-Z_3).Y_2)+ \\ (Z_2.X_3+(1-Z_2).X_1)*(Z_3.Y_1+(1-Z_3).Y_2-Z_1.Y_2-(1-Z_1).Y_3)+ \\ (Z_3.X_1+(1-Z_3).X_2)*(Z_1.Y_2+(1-Z_1).Y_3-Z_2.Y_3-(1-Z_2).Y_1) = 0 \quad (T)$$

Selon la majoration K(1), nous avons le point suivant à évaluer :

$$U_1=2^{384}+1, \quad U_2=(U_1^{1729}-1)/(U_1-1)+1, \quad U_3=(U_2^{1729}-1)/(U_2-1)+1.$$

Selon la majoration K(1(1(1...))), nous avons le point suivant à évaluer :

$$U_1=2^{39}+1, \quad U_2=(U_1^{26}-1)/(U_1-1)+1, \quad U_3=(U_2^{26}-1)/(U_2-1)+1.$$

Selon la majoration K(1(1(1(...))),1(1(1(...))),1(1(1(...))),...), nous avons le point suivant à évaluer :

$$U_1=2^{29}+1, \quad U_2=(U_1^{16}-1)/(U_1-1)+1, \quad U_3=(U_2^{16}-1)/(U_2-1)+1.$$

Notes à propos des méthodes de majoration :

- La dernière majoration correspond aux polynômes en $U_3^{15}U_2^{15}U_1^{15}$ avec pour coefficient maximum 2^{29} .

- La meilleure majoration possible est obtenue d'après les divisions effectives : c'est un polynôme en $U_3^{11}U_2^{11}U_1^{11}$ avec pour coefficient maximum 56.

- Dans le cas de presque-divisions, on obtient un polynôme en $U_3^{15}U_2^{15}U_1^{15}$. avec pour coefficient maximum 4392.

La majoration obtenue est donc bonne du point de vue des degrés partiels et moins bonne pour les coefficients maximum.

[[fin de la preuve]]

Conclusion de ce chapitre :

L'apport d'une arithmétique exacte redondante aux méthodes de démonstrations automatiques en géométrie est bien mise en évidence dans ce chapitre. Les travaux supplémentaires que nous avons effectués sur la rationalisation des constructions géométriques, et sur la majoration des formes polynomiales apportent leur contribution respective à la vérification de contraintes et à la démonstration de propriétés géométriques.

Au-delà du domaine de la démonstration automatique formelle abordé ici, nous envisageons l'emploi de CLP(Géométrie) pour des démonstrations automatiques approchées selon deux pistes. La première piste porte sur une démonstration probabiliste des propriétés géométriques. Un exemple numérique quelconque démontre un théorème à 100% en moyenne, c'est à dire avec un taux d'erreur nul en moyenne. Une seconde piste constitue une approche plus délicate et controversée. Nous envisageons une démonstration des propriétés géométriques à probabilité fixée non nulle. Ce que nous perdrons en exactitude, nous le gagnerions en complexité algorithmique. Pour cela, nous adapterions une arithmétique en prenant comme corps de base de grands corps finis (cf conclusion du II.2).

III-4. Environnements informatisés d'apprentissage humain de la géométrie.

Introduction :

Ce dernier chapitre porte sur l'application essentielle de ce travail, : la réalisation d'environnements informatisés d'apprentissage humain de la géométrie.

En effet, si nous prenons le temps de nous arrêter sur l'historique du développement des idées contenues dans cet ouvrage, nous trouvons au départ la recherche d'un environnement déclaratif de dessins géométriques. En deux mots, il s'agit dans ce projet idéal de pouvoir créer et manipuler des figures géométriques obtenues par simple spécification géométrique. Le problème difficile de la construction automatique apparu dans ce projet donne lieu aux développements de la seconde partie.

Comme nous l'avons vu en II-3 et II-4, ce problème peut recevoir des solutions complètes ou incomplètes dans le cadre de la programmation logique avec contraintes. Le parti pris "linéaire" que nous avons choisi pour la résolution de ce problème, ainsi que notre volonté d'accepter des spécifications redondantes mène à un autre problème : l'obtention d'une représentation et d'une arithmétique normale pour les nombres constructibles. L'obtention de cette représentation normale constitue le principal résultat formel de ce manuscrit (cf II-2).

La recherche de la complexité effective de cette représentation nous a conduit au principal résultat expérimental de ce travail donné en partie I: il est possible dans beaucoup de circonstances de travailler sur des figures et des constructions géométriques linéaires et/ou rationnelles.

Ces deux résultats obtenus, il reste donc à revenir au projet idéal pour accéder aux résultats pratiques que nous cherchons : l'obtention d'environnements informatisés d'apprentissage humain de la géométrie. Notre objectif au cours de ce chapitre consiste à proposer des outils informatiques pour l'enseignement de la géométrie. Nous laissons bien sur le soin d'en apprécier la valeur didactique aux enseignants eux-mêmes.

Une proposition d'environnement idéal d'apprentissage humain de la géométrie est décrit largement dans la première section de ce chapitre. Écrit en 1995 pour les journées EIAO de l'ENS-Cachan, l'article [Bouhineau-95a] dont cette section est issue, a été remanié pour y insérer les résultats quant à la recherche d'une arithmétique normale qui y étaient pressentis. Il comporte de nouveaux exemples que l'exiguïté des actes n'avait permis d'exhiber.

La seconde section de ce chapitre porte sur une réflexion quant à la place des enseignants dans les environnements informatisés d'apprentissage humain. Elle mène à quelques résultats

intéressants du point de vue informatique quant à l'utilisation conjointe de différents paradigmes de la programmation logique avec contraintes : les paradigmes CLP(Géométrie) introduit ici, et CLP(Intervalle) présenté rapidement en II-1. Cette section s'inspire en grande partie de la publication [Bouhineau, Channac-96] à ITS'96 (Intelligent Tutoring System).

Enfin, la dernière section de ce chapitre porte sur la réalisation d'un environnement de dessin géométrique à main levée, comme sur un cahier de brouillon informatique... En ce qui concerne une partie de ce travail, effectué en collaboration avec Florence Nebon, nous nous inspirons librement du mémoire d'ingénieur CNAM [Nebon-96] rédigé à la fin de ce travail.

a) Environnements informatisés impératifs ou déclaratifs d'apprentissage humain de la géométrie.

Depuis quelques années, avec l'apparition des logiciels Cabri [Laborde-86], [Baulac-90], [Bellemain-92] et [Sketchpad-90] destinés à la construction et la manipulation directe en géométrie plane, l'enseignement de la géométrie évolue. Il semble même qu'une nouvelle branche de la géométrie apparaisse : la géométrie des animations ("dynamic geometry" en anglais). Elle a pour origine l'animation des figures rigides que l'on peut dessiner sur le papier. Elle se différencie des géométries usuelles par la manière de considérer les propriétés et d'une figure, et l'importance accordée aux lieux de points.

Plus précisément, la géométrie des animations s'intéresse à l'observation des invariants et des changements d'une figure géométrique quand est modifiée progressivement la position d'un élément de celle-ci. Ces animations sont effectuées tout en conservant les propriétés de définition de la figure. Elles sont appelées "manipulations directes". Par exemple, Cabri propose à chaque instant, la manipulation directe des objets de base de cette figure, et réaffiche en temps réel un dessin renouvelant la construction de l'utilisateur ayant défini la figure. L'utilisateur observe ainsi comment la figure se déforme tout en respectant les propriétés qui l'ont définie.

Les notions de propriété et de lieu de points sont redéfinies en fonction des animations. Une figure possède telle propriété, si cette propriété, dans le cadre de la géométrie statique, reste vérifiée à chaque instant lors de toutes déformations de la figure. Par exemple dans le micro monde Cabri, les constructions telles qu'elles ont été introduites plus haut, définissent des propriétés d'une figure. En effet, c'est le cas puisque la reconstruction de la figure en temps réel s'effectue en respectant ces constructions. Nous retrouvons la définition suivante: une propriété est un invariant d'une classe de figures, où le mot "invariant" prend une connotation expérimentale, et où le terme "classe de figures" désigne l'ensemble des figures obtenues par animation. Le lieu d'un point d'une figure, est l'ensemble des positions prises par ce point au cours d'une déformation de la figure. Par exemple, le lieu du centre de gravité G, d'un triangle ABC, lorsque A décrit une droite d, B et C restant fixe, est une droite d' parallèle à d. C'est une des conséquences du théorème de Thalès qui s'applique, car G est au $\frac{2}{3}$ de la médiane passant par A.

La géométrie des animations est tributaire des logiciels qui ont participé à sa naissance. Deux approches sont possibles quant à l'acquisition de la figure et son animation.

La première approche est constructive c'est à dire impérative. Cabri et Geometer'sketchpad en sont des exemples. Elle demande à l'utilisateur une construction de la figure à partir d'objets donnés nommés objets de base. Elle s'apparente aux langages de programmation procéduraux ou impératifs. La classe des animations possibles correspond aux animations des points de base de la figure. L'ensemble des invariants et des lieux de points observables dépend donc d'une part de la construction effectuée, d'autre part de l'ensemble des points de base choisis.

La seconde approche est déclarative, cf. [Allen, Idt, Trilling-93] et [Allen, Trilling-95]. Elle demande à l'utilisateur une spécification logique de la figure et indépendamment, un ensemble d'objets de base. Elle s'appuie sur des algorithmes de construction automatique pour transformer ce couple spécification/ensemble de point de base en une figure animable. La classe des animations envisagée est beaucoup plus large, puisqu'elle s'appuie sur un ensemble d'objets de base qui peut être changé sans modifier la spécification logique de la figure. Notre effort va dans la définition d'un environnement qui réalise cette approche.

Cette section contient deux parties :

- La première partie constituée d'exemples porte sur la géométrie dynamique. Des exemples d'activités de géométrie sont donnés. Ils redécouvrent un problème de construction, la méthode des lieux, et la géométrie des transformations du plan. Les deux approches, constructive (ou impératives) et déclaratives, sont comparées.

- La seconde partie porte sur la réalisation d'un environnement qui réalise l'approche déclarative. Elle s'appuie essentiellement sur le travail effectué dans la partie II de cette thèse.

Exemples.

Trois exemples d'activité de géométrie sont décrits ici. Le premier exemple concerne un problème de construction où la méthode des lieux est employée pour atteindre une solution. Le second exemple porte sur la recherche d'une démonstration pour un théorème donné. Le troisième exemple porte sur un autre problème de construction dont la résolution fait apparaître la géométrie des transformations du plan et la notion de "boite noire" en géométrie (cf [Cabri-93]).

L'apport de la géométrie dynamique, i.e. les animations, est clairement exprimé dans les recherches de construction ou de preuves. Les deux approches, constructive (ou impérative) et déclarative, sont mises en concurrence au cours de ces exemples.

Résolution d'un problème de construction en utilisant la méthode des lieux.

Commençons par observer une activité de résolution pour un problème de construction. Considérons l'énoncé suivant tiré de [Capponi-94] et proposé comme activité de construction de figures contraintes géométriquement sur Cabri, à des élèves d'une classe de troisième :

Étant donnés deux droites d_1 , d_2 et un point m , construire les points a et b , tels que $a \in d_1$, $b \in d_2$ et tel que m soit le milieu de $[a,b]$

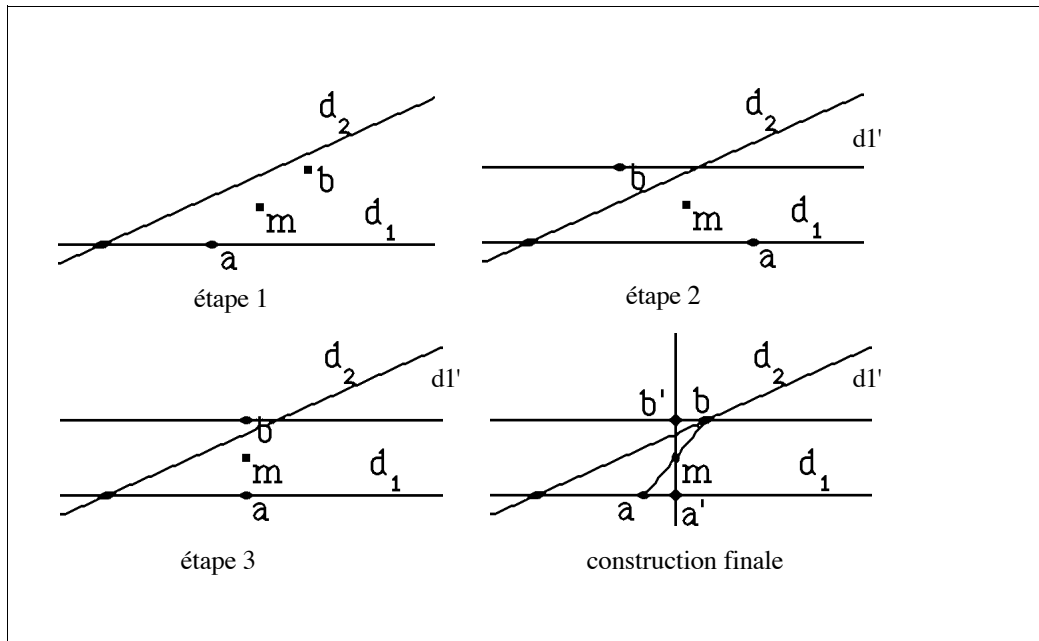


Figure III.4-a: Dessin d'un segment s'appuyant sur deux droites.

Une méthode classique pour résoudre les problèmes de construction consiste à utiliser la méthode des lieux cf. [Buthion-75], [Dufourd-90] i.e. chercher le lieu des inconnues du problème pour des versions sous contraintes du problème initial. Ici, il s'agit de chercher le lieu de **b** quand **a** "bouge" dans l'énoncé moins contraint suivant :

Étant donnés deux droites d_1, d_2 et un point m , construire a et b , tels que $a \in d_1$ et m milieu de $[a,b]$

Première étape : On résout ce problème facilement (cf. figure III.4-a), en prenant pour **a** un point quelconque de (d_1) et en construisant **b** le symétrique de **a** par rapport au point **m**.

Deuxième étape : En prenant d'autres positions pour le point **a** sur (d_1) , c'est à dire en animant **a**, on observe que le lieu du point **b** est une droite. Comme on peut le constater la géométrie des animations s'est introduite naturellement dans le problème. En dessinant le lieu de **b**, on observe de plus que la droite décrite par **b** est parallèle à (d_1) et qu'elle coupe (d_2) en un point solution. Pour résoudre notre problème de construction, il faut donc trouver comment construire cette droite, que l'on va nommer (d_1') .

Troisième étape : Il s'agit de déplacer **b** sur (d_1') pour trouver des éléments caractéristiques de (d_1') . Comment faire cela ? Dans une optique constructive, cette animation nécessite une nouvelle construction car **b** est "construit", il ne fait pas parti des objets de base. Dans une optique déclarative, il suffit de libérer **a**, et animer **b** (**a** n'appartient plus aux objets de base de la figure). Amenons **b** proche de **m**, nous observons **a** se déplacer sur (d_1) et venir à l'endroit de (d_1) le plus proche de **m**. On en conclut que (d_1) et (d_1') sont à égale distance de **m**, ce qui permet de construire définitivement la figure demandée. Au passage, nous

redécouvrons le résultat suivant de la géométrie des transformations du plan : "L'image d'une droite d par une symétrie de centre m est une droite parallèle à d et à égale distance de m ".

Construction finale : En partant du point m , construire le point a' projeté de m sur (d_1) . Puis le point b' symétrique de a' par rapport à m . Puis alors construire (d_1') , parallèle à (d_1) passant par m . Enfin prendre b intersection de (d_1') et de (d_2) et construire a symétrique de b par rapport à m .

La comparaison des deux approches dans le cas d'activités de construction, ou d'énoncés d'exercices non constructifs, est en faveur de l'approche déclarative. Les animations sont plus faciles à obtenir et leur nombre est plus grand pour un même effort de spécification. Par ailleurs, l'utilisation d'un algorithme de construction automatique dans cette approche permet d'obtenir plus rapidement un dessin solution du problème que l'utilisateur peut explorer. Ici par exemple, la spécification initiale permet d'obtenir un dessin où l'on observe que b est fixe pour la classe de tous les points m décrivant une parallèle à (d_1) (ce qui donne une autre idée de construction).

Autour de la démonstration d'un théorème, illustration d'une notion de hiérarchie de classes de figures

Considérons l'exercice portant sur la démonstration de la propriété suivante associée à une construction simple :

Étant donné un triangle (ABC) isocèle en B, et D l'intersection des parallèles à (BC) et (AB) passant par A et C respectivement, montrer que le triangle (ACD) est isocèle en D.

Le schéma de résolution de ce type d'exercices a été proposé par [Allen, Nicolas, Trilling-90] pour le système Mentoniez (voir aussi [Py-90]), il peut comporter quatre phases :

- Acquisition de la figure : l'élève construit une figure conforme aux hypothèses du théorème à démontrer.
- Appropriation de la figure : l'élève fait évoluer la figure pour découvrir des invariants ou des lieux de points intéressants.
- Exploration des propriétés : l'élève examine d'éventuelles propriétés suggérées par le système.
- Organisation de la démonstration : enfin l'élève construit une démonstration visant à prouver le théorème recherché.

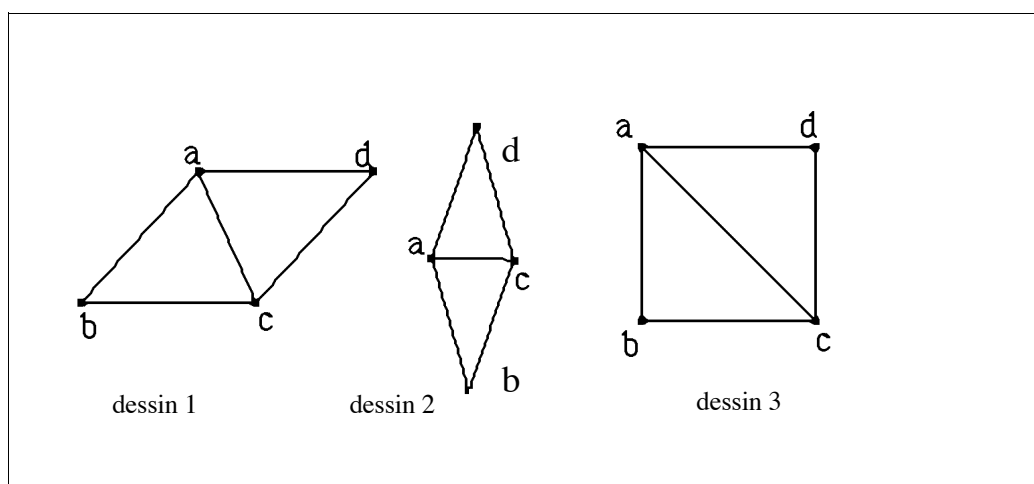


Figure III.4-b: animation d'un parallélogramme

L'animation d'une figure représente donc la deuxième phase du processus imaginé pour Mentoniezsh. Imaginons que l'élève confronté au problème décrit au début de ce paragraphe obtienne une figure correspondant aux hypothèses de son énoncé. L'élève se trouve donc en face d'un premier dessin qui peut être selon le triangle de départ (**abc**), le dessin 1, 2 ou 3 de la figure III.4-b. Ces dessins sont riches d'implicites. En l'absence d'animations, les figures du III.4-b peuvent sembler différentes alors qu'elles sont définies par la même construction. Le premier dessin se présente comme la plupart des parallélogrammes qui se trouvent dans les livres de cours, ou comme l'habitude nous guide pour les dessiner au tableau. La direction horizontale pour base reprend l'une des directions pour lesquelles il y a une symétrie de translation. Le second dessin est représentatif d'un losange avec ses deux axes de symétrie respectivement horizontal et vertical. Enfin, le dernier exemple représente à s'y méprendre à un carré, ou au moins un rectangle.

Déformons la figure. Dans une approche constructive, la classe des animations est contrainte par la construction du triangle (**abc**). A priori, l'élève peut déplacer certains points parmi **a**, **b** et **c** selon des modalités qui dépendent de la construction et du logiciel de dessin. Dans une approche déclarative, l'élève peut manipuler les quatre points plus librement. La différence entre les deux approches paraît minime mais elle est significative car **d** appartient à la partie de la figure sur laquelle porte la conclusion du théorème.

La manipulation conduit au second dessin, le plus général, qui représente un losange. Plus précisément, en manipulant la figure sans forcément atteindre le deuxième dessin, l'utilisateur se rend compte d'une part que la figure n'est pas seulement un parallélogramme mais aussi un losange, et seulement cela et d'autre part qu'elle n'est pas un carré. Les conclusions erronées tirées un peu trop rapidement de l'étude des dessins statiques sont corrigées avec l'animation.

Nous aboutissons au résultat suivant : l'animation de figures géométriques permet la compréhension et l'appropriation de la notion de classe de figures géométriques (comme la classe des rectangles ou celle des losanges), et des liens d'inclusion entre les différentes classes de figures (comme l'inclusion de la classe des losanges dans celle des parallélogrammes). Par

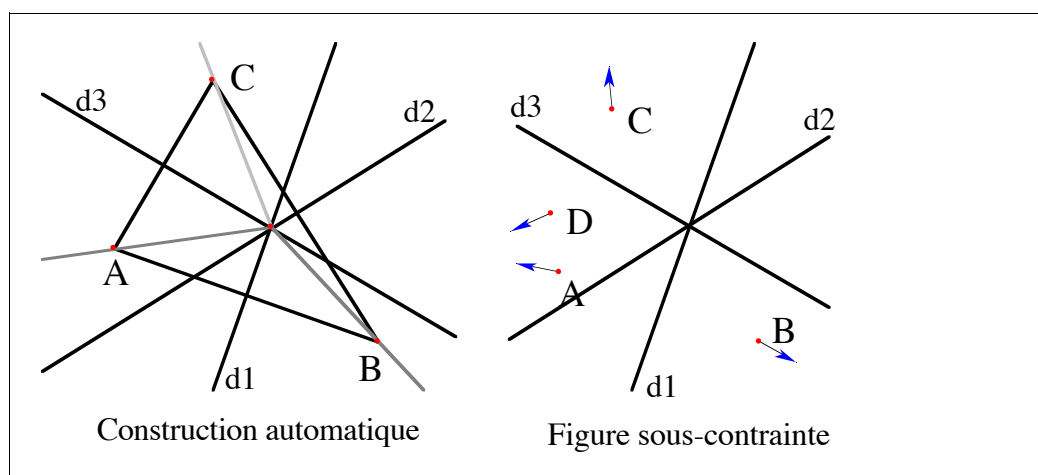


Figure III.4-c : Construction d'un triangle à partir des médiatrices.

ailleurs, l'animation d'une figure, obtenue pendant la résolution d'un problème, permet de découvrir plus justement son appartenance à la bonne classe de figures pointées par l'énoncé de l'exercice.

La démonstration du théorème est presque finie. Il s'agit maintenant de montrer que la figure est effectivement un losange, conformément à notre intuition guidée par l'animation. Enfin, il suffit d'utiliser les propriétés du losange pour conclure.

Transformations du plan, invariants et boîtes noires.

La géométrie offre un visage radicalement différent lorsque les transformations du plan sont prises en considération prioritairement, au lieu des configurations de points. La manipulation directe peut être un outil de découverte étonnant à cette occasion.

Par exemple, il saute aux yeux quand un point A est manipulé si un point B est ou non son image par une translation. La translation est démasquée dans une approche dynamique, d'autant plus quand la possibilité de manipuler B est offerte. De même avec un peu d'expérience, il est aisé de reconnaître, grâce à quelques manipulations, les points liés par des rotations, des symétries, ou des isométries.

L'exemple suivant correspond à un problème de construction issu du livre de classe terminale scientifique aux Éditions Hachette (1991).

Étant données trois droites d_1 , d_2 , d_3 construire un triangle (ABC) tels que les droites d_1 , d_2 , d_3 soient les médiatrices des trois cotés du triangle.

Plaçons-nous dans l'hypothèse où l'élève obtient à partir de l'énoncé, une figure qu'il peut manipuler. Figure "construction automatique" du III.4-c. Cette hypothèse interdit l'approche constructive, car l'énoncé n'est pas constructif.

En manipulant le triangle, l'élève se rend compte que les points A, B, et C sont astreints à évoluer sur trois droites dessinées en pointillés d_A , d_B , d_C . Ces trois droites sont de plus

concourantes et passent par le point d'intersection des trois droites d_1 , d_2 , d_3 . Il reste à découvrir quelles sont ces droites.

Relâchons une contrainte du problème comme dans la méthode des lieux. Ici, les lieux seront plutôt des invariants en termes de transformation du plan comme nous allons le voir. Observons la figure "sous contrainte" du III.4-c où trois segments $[A,B]$, $[B,C]$, $[C,D]$ sont construits tels que les droites d_1 , d_2 , d_3 soient leurs médiatrices respectives. Pour rappel, la solution du problème initial consiste à ajouter la contrainte $A=D$. En manipulant le point A, une réflexion selon d_A est découverte, en effet il semble que D est l'image de A par cette réflexion.

Le problème est proche de sa conclusion. Il reste se rappeler le fait que la composée de trois réflexions par rapport à des droites concourantes est une réflexion.

Un autre exemple d'activité s'apparente à la découverte et à l'investigation des transformations du plan. Il s'agit des activités où l'élève est confronté à une "boîte noire". Il s'agit d'une activité où l'élève observe, au travers des animations qu'il produit, une figure dont une partie est cachée (la construction) et dont il ne voit que les comportements dynamiques. Bien sûr, l'approche fonctionnelle biaise un peu le jeu au sens où les seuls éléments de base peuvent être manipulés par l'élève. L'approche déclarative est exempte de cette critique.

A chaque couple de points de la figure correspond implicitement une transformation du plan, éventuellement complexe. Et effectivement, souvent les activités de "boîtes noires" portent sur la redécouverte des transformations du plan. Par exemple, dans [Caponi-97] sont étudiées l'homothétie de rapport -2 et la projection orthogonale. L'exercice pour l'élève peut consister à réaliser une figure géométrique qui a le même comportement que la boîte noire.

Dans le cadre d'activités de découvertes, la démarche va dans le sens d'une association fonctionnelle entre une transformation du plan encore inconnue et une construction géométrique maîtrisée par l'élève. Dans le cadre d'un exercice de réinvestissement, le jeu consiste à reconnaître la transformation du plan qui apparaît dans l'animation.

Vers une approche déclarative.

L'écriture d'un logiciel prototype, GéoSpécif, qui insiste sur la manipulation directe des figures géométriques nous a permis de mieux comprendre les exigences imposées par un tel projet, en particulier l'importance de l'algorithme de construction automatique. Nous présentons ici le résultat de notre travail. Nous montrons d'abord que la géométrie des animations nécessite une approche déclarative. Nous décrivons ensuite l'emploi de l'algorithme de construction automatique décrit lors de la partie II de cette thèse.

Géométrie des animations et géométrie déclarative.

Les logiciels comme Cabri ou Geometer Sketchpad ont permis de découvrir toute la puissance des animations. Les activités présentées dans la première partie ne sont que partiellement réalisables avec ces logiciels. En effet, ces environnements fournissent la classe des animations possibles avec les points de base de la figure seulement. Cela vient du mode opératoire choisi pour obtenir une figure. Ces logiciels construisent une figure progressivement à partir des objets de base, et d'opérations géométriques introduisant de nouveaux objets dit "construits". Lors de la manipulation des objets de base, la figure se reconstruit en temps réel en suivant le processus de construction donné initialement. Les objets "construits" ne peuvent

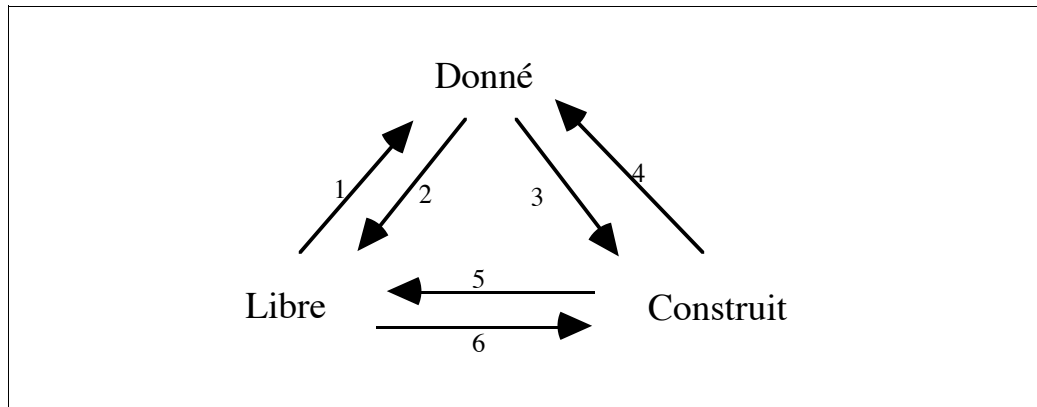


Figure III.4-d : Statut d'un objet pour une approche déclarative.

pas être manipulés de la même manière, car ils sont les résultats d'opérations de construction qui ne sont pas inversibles.

Pour rappel, cette difficulté est connue depuis le XIX^e siècle avec les travaux de Galois puis de Wantzel sur la recherche de moyens pour reconnaître si un problème de géométrie peut se résoudre par la règle et le compas. Elle est reprise dans [Lebesgue-89] qui montre que certains problèmes ne peuvent être résolus à la règle et au compas, quand bien même il suffirait d'inverser un modèle comme dans le cas de la trisection de l'angle. Cette limitation théorique s'applique particulièrement au domaine de la construction automatique : il n'y a pas de méthode générale simple pour résoudre tous les problèmes de géométrie à la règle et au compas. Et nous rappelons que les algorithmes les plus performants dans le domaine sont de complexité exponentielle, ce qui nous a amené aux résultats de la partie II.

Statut d'un objet dans une figure dynamique

Avant de considérer l'architecture du système s'appuyant sur l'algorithme de construction automatique, attardons-nous sur un choix essentiel. Pour favoriser la classe des animations possibles, il faut reconsidérer les priorités attribuées à la construction et à la spécification. C'est le lieu du passage entre la programmation géométrique procédurale et la programmation géométrique déclarative.

Dans l'approche constructive, la construction constitue un élément indispensable : une spécification ne peut être ajoutée à une figure, si elle ne correspond à une nouvelle construction. De fait, l'importance accordée à la construction divise les objets en deux classes figées, la classe des objets de base et la classe des objets construits, et les animations sont déterminées.

Dans l'approche déclarative, la spécification est privilégiée par rapport à la construction. Par exemple, dans GéoSpécif les constructions se font automatiquement, lorsque cela est possible, à partir des objets donnés par l'utilisateur et des spécifications qui lient ces objets selon l'algorithme de la partie II. À la différence de l'approche constructive, le statut des objets géométriques n'est pas figé. Un point n'a plus un statut figé de point de base ou de point construit, mais un statut que l'utilisateur peut faire évoluer au cours de la session selon la figure III.4-d. C'est l'un des points novateurs, importants et délicats du système.

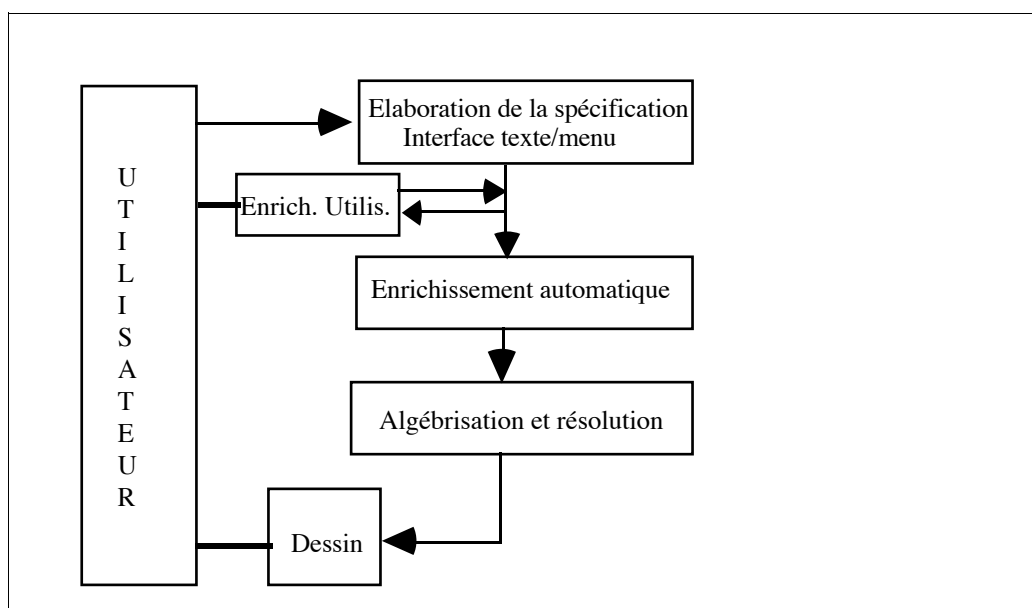


Figure III-4.e : Architecture de GéoSpécif.

Décrivons plus précisément la figure III.4-d. Les points donnés et construits sont ceux qui apparaissent à l'écran à l'emplacement désigné par l'utilisateur ou calculé par le logiciel pour respecter la spécification. Les points libres apparaissent dans la spécification de la figure mais ne sont pas affichés parce qu'ils sont insuffisamment contraints ou qu'ils restent inconnus. Les flèches 1 et 4 de la figure III.4-d correspondent à des instanciations, 2 et 3 à des libérations, ces quatre flèches correspondent à des actions volontaires de l'utilisateur relatives à l'animation des objets et non à la spécification géométrique de la figure. Les flèches 5 et 6 correspondent aux constructions automatiques ou aux défauts de construction. Elles ne sont pas directement commandées par l'utilisateur, elles le sont soit par un changement dans la spécification de la figure, soit par l'utilisation de l'algorithme de construction automatique.

Il est envisageable de compléter le diagramme de la figure III.4-d avec des points dont le degré de liberté est de 1, au lieu de 2 naturellement pour les points, cf. II.4-e. De même, il est envisageable de considérer des algorithmes d'instanciation automatique qui infèrent automatiquement le statut des points d'une figure.

Description du système.

Pour valider notre approche déclarative, nous nous appuyons sur l'algorithme de construction automatique cité en II-3. Il prend en compte la spécification logique de la figure de manière non déterministe (au sens de Prolog) et incrémentale. Pour rappel, l'algorithme utilisé opère en deux temps. Tout d'abord il complète automatiquement la spécification avec des contraintes géométriques vérifiées par la figure. Ensuite, il effectue une traduction algébrique du problème et tente de résoudre le système d'équations algébriques obtenu (cf figure III-4.e).

La combinaison de ces méthodes de construction automatique permet d'avoir déjà des résultats acceptables, comme nous l'avons montré en II-4.

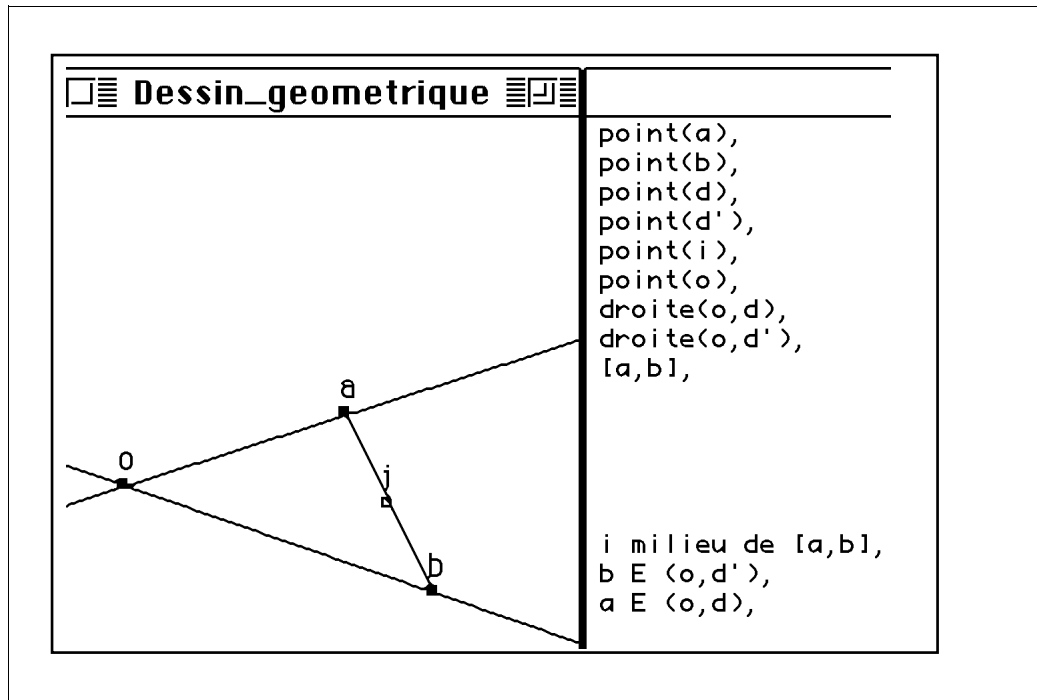


Figure III-4.f : Copie d 'écran de GéoSpécif sur le premier exemple.

Exemple : Les figures III.4-a,b,c sont obtenues directement à partir de leur spécification, en respectivement 0,1 seconde et moins de 0,1 seconde. Ces performances permettent une animation en temps réel. Pour la figure III.4-a, par exemple, la spécification introduite est: $A \in d1$, $B \in d2$, P milieu de $[A,B]$.

Conclusion : Notre démarche se situe au croisement de deux mouvements appartenant à deux domaines relativement éloignés. Chronologiquement, le premier mouvement s'est développé avec Prolog, il s'agit de la programmation déclarative et non déterministe, il se prolonge aujourd'hui avec l'apport de la programmation par contraintes. Le second mouvement, plus récent, a été engendré par Cabri, il s'agit de la géométrie des animations et il se développe dans le domaine de l'éducation avec un grand succès. La réalisation de GéoSpécif, notre logiciel de géométrie déclarative tourné vers l'animation, et son utilisation montrent le pouvoir d'expression et la facilité d'utilisation des paradigmes mis en jeu. En quelque sorte, il participe à la validation de ces deux mouvements.

b) Génération d'exemples et de contre-exemples géométriques dans un "préceptoriel".

Notre démarche consiste à apporter une aide à un enseignant au cours du déroulement d'une séquence d'enseignement.

Pour nommer notre approche nous introduisons un nouveau terme "préceptoriel" qui s'ajoute à la typologie des Environnements Informatisés d'Apprentissages Humains. Par rapport aux tutoriels, didacticiels, imagiciels, et micro-mondes, un préceptoriel appartient à la classe des EIAH centrés sur l'enseignant. Il fait référence à notre volonté de distinguer

l'apprenant de l'enseignant dans un EIAH, et développer une partie de l'intelligence du système à l'intention du professeur.

L'aide qu'un préceptoriel de géométrie peut proposer au professeur est double. D'une part, l'aide peut porter sur le diagnostique de la production de l'apprenant, d'autre part, sur la recherche de contre-exemples les plus expressifs pour montrer les erreurs de l'apprenant.

Sur le premier point, nous renvoyons à la lecture de [Desmoulins-95]. La méthode de démonstration automatique présentée en III-3 pourrait être utilisée suivant un schéma identique à celui proposé par Cyrille Desmoulins.

En ce qui concerne le second point, cette section lui est consacrée. Elle comporte deux parties. La première porte sur la définition et la production de "contre-exemples" et la seconde sur des exemples de telles productions. Pour la production de contre-exemples, l'algorithme de construction automatique de CLP(Géométrie) a été remplacé par un algorithme utilisant une approche CLP(Intervalle). Cette approche a fourni des résultats satisfaisant sur les exemples présentés.

Production de contre-exemples.

Le cadre de cette étude est la construction de figures géométriques dans un environnement informatisé d'apprentissage humain. Cette activité de construction peut être une activité en soi afin de vérifier que l'élève a bien compris les hypothèses et les conclusions du problème [Desmoulins-94]. Elle peut aussi être le support d'une activité de découverte et de démonstration de propriétés géométriques.

Attendu que, comme l'énonce Colette Laborde [Laborde-93] :

"Le tracé à l'écran d'un dessin attaché à un objet géométrique doit garder au cours du déplacement ses propriétés spatiales rendant compte des propriétés géométriques de cet objet,"

et comme Bernard Capponi ajoute :

"Cet aspect [...] change la nature du travail proposé aux élèves. Ce n'est plus un simple dessin qui doit être réalisé mais ils [les élèves] doivent fournir une description de la construction".

Il s'ensuit que la production d'un élève est double. Elle est d'une part visuelle avec le dessin rapporté à l'écran, et d'autre part géométrique avec le procédé de construction suivi.

Ce qui précède inspire la classification des erreurs suivante. Nous classons ces doubles productions d'un élève dans l'une des trois catégories suivantes :

- **les dessins visuellement justes et construits correctement.**
- **les dessins visuellement justes et construits incorrectement.**
- **les dessins visuellement non justes et donc construits incorrectement.**

Comme énoncé précédemment par Colette Laborde, la figure de l'élève doit respecter la classe d'animation du problème posé par le professeur. Elle doit naturellement contenir toutes

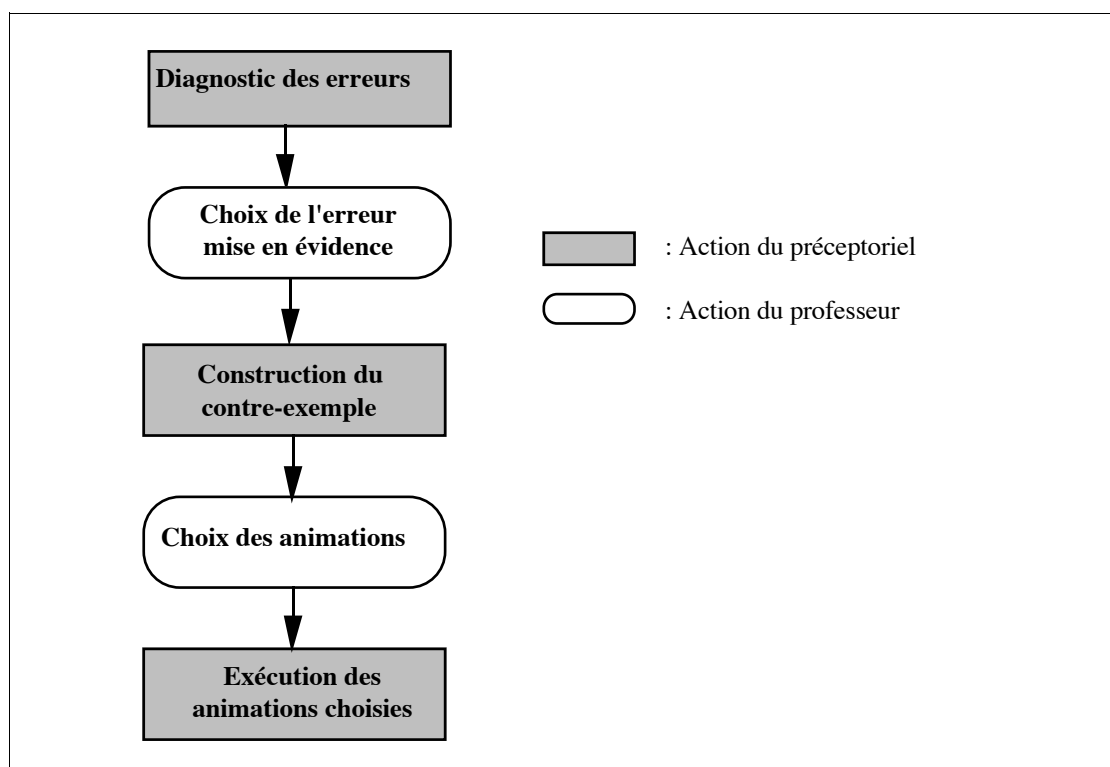


Figure III-4.g : Schéma d'un processus d'aide en géométrie dynamique.

les propriétés demandées, mais seulement les propriétés demandées. En effet, les enseignants ne se satisfont pas de figures particulières. Ce rejet est motivé par la volonté d'éviter les fausses conjectures des élèves induites par la particularité de telle ou telle figure. Ainsi, les deux types d'erreurs retrouvées sont les :

- **sous-contraintes** : lorsqu'une propriété demandée n'est pas satisfaite,
- **sur-contraintes** : lorsqu'une propriété est satisfaite alors qu'elle n'est pas demandée.

Les moyens d'investigation mis à dispositions du professeur pour vérifier les productions des apprenants sont au nombre de trois. Jean-François Bonnet les mentionne dans [Bonnet-93] :

- **L'utilisation de l'"historique"**. Par cet outil, le professeur accède à la séquence d'opérations caractérisant la construction. Ainsi, il peut décider de la correction ou non de la figure.

- **La manipulation de tous les objets de base de la figure**. Par ces déformations, le professeur peut comprendre les propriétés qui lient les objets liés à l'objet manipulé. Mais ce n'est qu'à la suite de toutes les déformations de tous les objets de la figure que le professeur peut valider ou non celle-ci.

- **L'affichage de tous les éléments de la figure**. Par cet affichage, le professeur ne peut avoir qu'une idée du procédé de construction suivi.

Comme moyen de test pour la validité de la production de l'élève, Colette Laborde [Laborde-93] propose :

Propriété manquante	Propriété antagoniste ajoutée
Un point n'appartient pas à une droite	Le point est placé à une distance supérieure à un seuil fixé de la droite
Un point n'appartient pas à un cercle	Le point est placé à une distance supérieure au diamètre du cercle, du centre de ce dernier
Un objet n'est pas parallèle à un autre objet	Les objets sont spécifiés perpendiculaires
Un objet n'est pas perpendiculaire à un autre objet	Les objets sont spécifiés parallèles
Un point n'est pas le milieu d'un segment	Le point est placé à une distance d'un dixième de la longueur du segment, d'une extrémité de celui-ci

Table III.4-a : Correspondance entre propriété manquante et propriété antagoniste ajoutée.

• **L'utilisation d'une macro-construction.** Le test consiste à superposer la figure de l'élève avec la macro-construction solution du professeur appliquée aux données de l'élève.

Notre objectif est de compléter ces moyens d'investigation ou de test par un système de production automatique de contre-exemples dans les cas de figures sous contraintes. La recherche de contre-exemple est un moyen courant du professeur pour valider la solution de l'élève.

Mais qu'est-ce qu'un contre-exemple ? Un contre-exemple est une figure mettant clairement en évidence les insuffisances d'une figure sous contrainte. Au regard du principe énoncé par Jean-Marie Laborde dans [Laborde-95] :

"... une figure n'est acceptée comme contre-exemple que si l'utilisateur perçoit le chemin qui conduit de "sa" configuration à celle proposée comme contre-exemple par l'environnement."

Nous proposons la production de contre-exemples obtenus par ajout à la spécification issue de la construction de l'élève, d'une propriété antagoniste à la propriété manquante. De plus, afin de lier la figure de l'élève au contre-exemple, un préceptoriel doit proposer des animations menant de la première à la seconde.

Les erreurs commises étant détectées, il s'agit en gardant à l'esprit le principe de Jean-Marie Laborde précédemment énoncé :

- 1. Choisir parmi les erreurs commises par l'élève la ou les erreurs à mettre en évidence.
- 2. Choisir la manière de mettre en évidence une erreur.
- 3. Proposer une animation de la figure de l'élève qui mène au contre-exemple.

Au regard d'expérimentations, il apparaît préférable de montrer une seule erreur à la fois. En effet, dans le cas contraire il n'est pas aisé de comprendre le cheminement qui conduit de la figure de l'élève au contre-exemple ou à l'exemple. Il se pose alors le problème du choix du

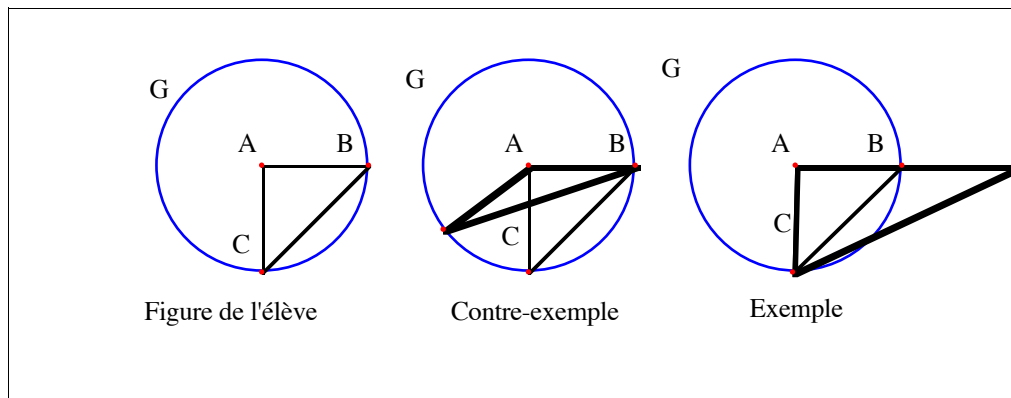


Figure III.4-h : Exemple de construction erronée, contre-exemple et exemple.

type de l'erreur à mettre en évidence. Ce choix est effectué par défaut par le système sous le contrôle toutefois du professeur. L'intérêt de tels comportements par défaut est double. Ils permettent au professeur d'une part de se décharger de ces contraintes de choix et d'autre part de définir par là même ses préférences stylistiques et didactiques.

Ce choix doit consister dans le cas d'une sous-contrainte en une correspondance entre les classes d'erreurs et les propriétés antagonistes ajoutées. Un exemple d'une telle correspondance est donné par le tableau III.4-a.

Exemples.

Exemple 1 : La situation porte sur le problème suivant. L'élève doit construire un triangle ABC rectangle en A. Il répond à cet exercice par la construction suivante et le dessin présenté figure III.4-h:

**Création du cercle G de centre A passant par le point B,
construction d'un point C sur le cercle G.**

Après création des segments $[A, B]$, $[B, C]$ et $[A, C]$, la figure de l'élève est finie, cf figure III.4-h.

Cette figure est erronée même si visuellement elle semble répondre à l'attente du professeur. L'élève a commis ici deux erreurs : l'absence d'orthogonalité en A et la présence d'égalité de longueur des segments $[A, B]$ et $[A, C]$. Pour mettre en évidence la première erreur, nous proposons un contre-exemple obtenu par rotation du point C autour de A. Afin de montrer la particularité entre les segments $[A, B]$ et $[B, C]$, nous affichons un exemple de triangle rectangle quelconque, comme illustré sur la figure III.4-h.

Exemple 2 : La situation concerne la construction d'un carré. L'élève répond avec la construction suivante :

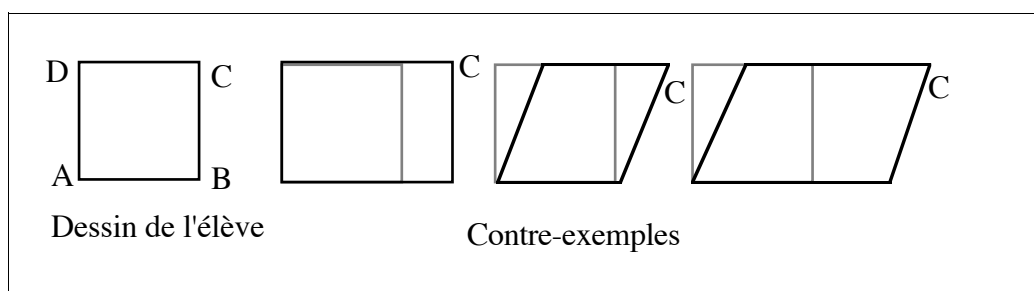


Figure III.4-i : Solution d'un élève et contre-exemples.

Créer deux points A et B quelconques, une droite (d) passant par A, et un point D sur (d). Construire les droites (d') passant par D parallèle à (AB), et (d'') passant par B et parallèle à (d), construire C à l'intersection de (d') et (d'')

Après construction des segments constituant le polygone (ABCD) et l'effacement des droites (d), (d') et (d''), le dessin proposé par l'élève est celui présenté figure III.4-i.

L'élève a construit un parallélogramme à la place du carré demandé, cf. figure III.4-i. Visuellement, il semble avoir répondu à l'exercice, mais l'analyse de la construction montre que la figure produite ne réalise pas deux propriétés géométriques du carré : la perpendicularité et l'égalité des longueurs de deux cotés consécutifs. Sur le dessin, la perpendicularité est obtenue astucieusement grâce à la pixélisation de l'écran, l'égalité des longueurs est obtenue au jugé. Trois types de contre-exemples peuvent être produits. L'animation de C met en évidence l'inégalité des longueurs. La rotation de (d) autour de A montre le manque de perpendicularité. Ces deux animations composées permettent de voir les deux erreurs, et donnent la vraie classe des figures dessinées par l'élève.

Exemple 3 : Sur le dernier exemple, nous décrivons le processus complet. Le problème est le suivant : Construire la médiane du triangle ABC issue du sommet A.

La spécification logique de ce problème est :

point(A), point(B), point(C), point(H), droite(AB), droite(AC), droite(BC), droite(AH), $H \in (BC)$, $|BH| = |CH|$.

1. La construction fournie par l'élève, illustrée en pointillés sur la figure III.4-j, est la suivante :

- création des points A, B et C quelconques,
- création des segments [A, B], [A, C] et [B, C],
- construction de la droite perpendiculaire au segment [B, C] passant par le point A,

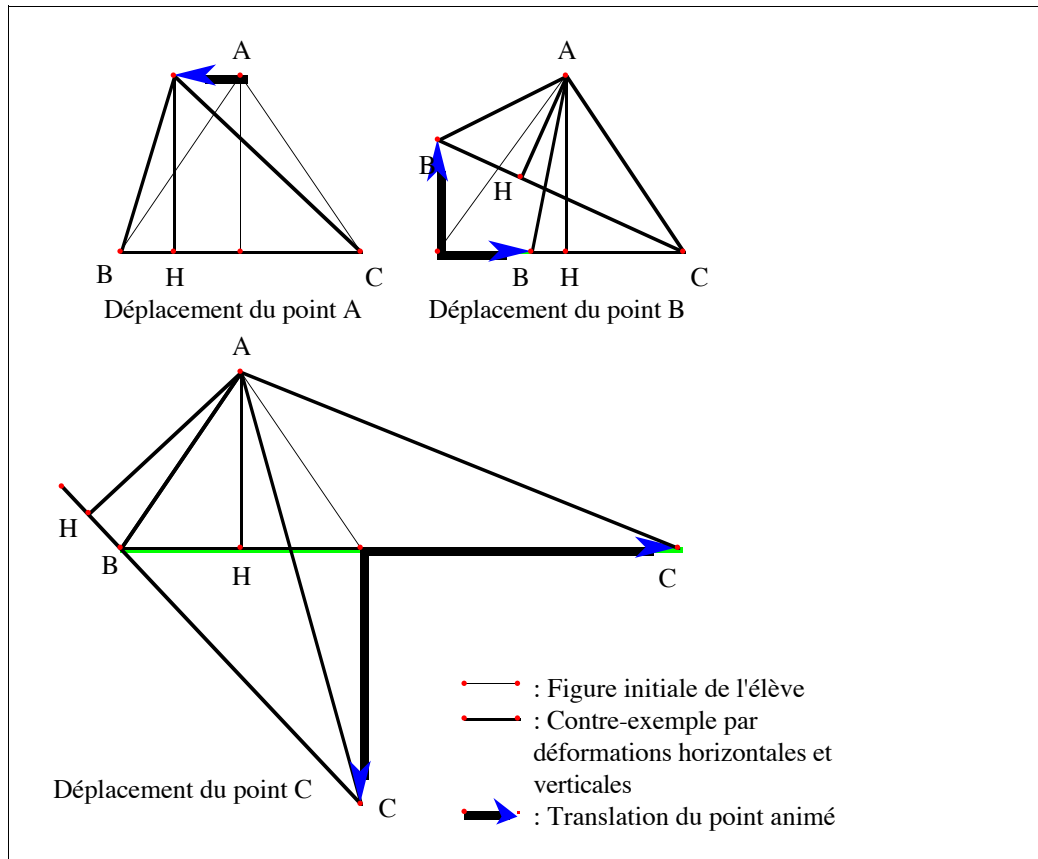


Figure III.4-j : Animations menant aux contre-exemples.

- construction du point d'intersection H de la droite perpendiculaire au segment $[B, C]$ passant par le point A et du segment $[B, C]$.

L'erreur commise par l'élève réside dans la confusion entre la notion de hauteur et la notion de médiane dans un triangle.

2. Si le dessin semble répondre au problème, le point H étant proche du milieu du segment $[B, C]$, c'est parce qu'il s'agit d'un dessin où le triangle ABC semble isocèle en A. La construction de l'élève correspond à la spécification logique suivante :

point(A), point(B), point(C), point(H), droite(BC), droite(AH), $H \in (BC)$, $(AH) \perp (BC)$.

Cette spécification ne répond pas au problème posé. En effet, cette figure est incorrecte d'une part parce qu'elle n'implique pas que le point H soit le milieu du segment $[B, C]$, et d'autre part parce qu'elle implique que la droite (AH) est perpendiculaire à la droite (BC).

3. L'étape de diagnostic consiste à déterminer que le point H n'est pas le milieu du segment $[B, C]$ et que la droite (AH) est perpendiculaire à la droite (BC).

4. Une fois le diagnostic effectué, le système choisit prioritairement de mettre en évidence l'erreur de sous-contrainte commise à l'aide d'une animation des éléments de la figure. Afin de montrer clairement que le point H n'est pas le milieu du segment [B, C], le système choisit de le contraindre à être au dixième de la distance du segment [B, C], comme cela est explicité dans le tableau .III.4-a La spécification logique obtenue du contre-exemple est la suivante :

<p>point(A), point(B), point(C), point(H), droite(A, B), droite(A, C), droite(B, C), droite(A, H), $H \in (B, C)$, (A, H) \perp (B, C), $BH = 1/10 BC$.</p>

5. L'ajout d'une contrainte à un élément de la figure impliquant la libération d'une autre, le système libère successivement les points de base de la figure afin de permettre la construction des contre-exemples. Ainsi, le point A, ou le point B, ou encore le point C sont déplacés suivant l'axe des abscisses ou l'axe des ordonnées, donnant lieu aux constructions rapportées par la figure III.4-j.

Conclusion : Le préceptuel que nous présentons ici est la première étape d'un effort pour fournir des outils d'aide au professeur. Notre recherche porte maintenant sur la définition d'un langage permettant au professeur de spécifier le type de contre-exemples et d'animations qu'il souhaite.

c) Réalisation d'un environnement de dessin géométrique à main levée

L'objectif de notre travail est la définition d'un environnement d'acquisition graphique de contraintes géométriques. Il s'agit pour le système d'effectuer l'apprentissage automatique de la spécification d'une figure géométrique à partir d'un dessin réalisé interactivement par l'utilisateur. Pour la définition du processus d'apprentissage, nous nous inspirons des principes issus de l'apprentissage automatique de concept et plus précisément de la programmation logique inductive.

Notre approche consiste à exploiter la richesse d'expression du dessin à travers son aspect perceptif. L'algorithme d'inférence d'une spécification logique à partir du dessin procède par un examen successif des éléments composant le dessin. Il s'intéresse aux propriétés géométriques liant de manière implicite et approximative ces divers éléments. L'utilisateur prend part de manière active au processus d'apprentissage. Il se charge de la vérification des propriétés géométriques découvertes, au travers de l'animation de la figure produite à partir de la spécification inférée et d'objets de base définis graphiquement. En cas d'insatisfaction, l'utilisateur peut généraliser, ou au contraire spécialiser la spécification logique inférée, en proposant respectivement des exemples ou contre-exemples graphiques.

Le processus complet d'apprentissage passe par les cinq étapes suivantes.

- 1- Acquisition d'un dessin réalisé interactivement par l'utilisateur.
- 2- Inférence automatique d'une spécification logique à partir du dessin proposé.
- 3- Construction et animation d'une figure géométrique associée à la spécification géométrique.

- 4- Spécialisation de la spécification par proposition d'exemples.
- 5- Généralisation de la spécification par proposition de contre-exemples.

Nous ne décrivons ici que succinctement les étapes -1- et -2- de ce processus. L'étape -3- correspond à l'emploi de l'algorithme décrit lors de la seconde partie de ce manuscrit. Nous renvoyons le lecteur à [Nebon-96] pour les étapes -4- et -5-.

Acquisition d'un dessin réalisé interactivement par l'utilisateur.

Le système doit offrir à l'utilisateur un outil de réalisation à la fois simple, rapide et agréable d'un dessin exprimant de manière implicite et approximative un ensemble de propriétés géométriques souhaitées. Dans la réalisation de Florence Nebon, cf. [Nebon-96], trois outils accessibles par menu sont proposés. Ils permettent, respectivement, l'obtention de points, droites et cercles. Nous préconisons l'utilisation d'un seul outil, le crayon électronique (*crayonel*) qui, selon le comportement de l'utilisateur, fournit un point, une droite ou un cercle avec les règles d'interprétation qui suivent. Excepté pour le dessin d'un cercle, les interprétations du dessin proposé par le crayon sont immédiates :

- Si le *crayonel* est utilisé pour un dessin de taille inférieure à quelques pixels, cette figure est un point. Le point considéré est celui défini par la première position du *crayonel*.

- Si le *crayonel* est utilisé pour un dessin dont les premiers points dépassent quelques pixels et qui semblent présenter une courbure faible, cette figure est une droite. La droite considérée est celle définie comme passant par les deux points particuliers suivants : la première position du *crayonel*, et la dernière position du *crayonel* (ou la position courante du *crayonel* si le dessin de la figure n'est pas terminé)

- Si le *crayonel* est utilisé pour un dessin dont les premiers points semblent présenter une courbure non négligeable, cette figure est un cercle. Le cercle considéré est celui défini comme passant par les trois points caractéristiques suivants : la première position du *crayonel*, la dernière position du *crayonel* (ou la position courante du *crayonel* si le dessin de la figure n'est pas terminé), et la position du *crayonel* où l'utilisateur s'est attardé le plus.

L'interprétation du dessin proposé par le *crayonel* respecte la règle suivante : si le pointeur parcourt une figure géométrique point, droite ou cercle, alors cette figure est découverte exactement.

L'interprétation du dessin proposé par le *crayonel* lors du dessin d'un cercle est la plus difficile. La règle précédente interdit la définition d'un cercle par centre et rayon. Elle implique que le cercle soit défini comme circonscrit à trois points pris parmi ceux occupés par le *crayonel*. D'autres choix sont possibles pour ces trois points, par exemple le premier point, le dernier emplacement du *crayonel* et l'emplacement le plus éloigné de ces deux emplacements. Expérimentalement, même en faisant varier la norme utilisée pour calculer cet éloignement maximum, ce choix est moins efficace que celui proposé (qui est beaucoup plus humain).

En ce qui concerne la représentation graphique et les manipulations ultérieures des figures géométriques obtenues, nous renvoyons le lecteur à [Nebon-96].

Inférence automatique d'une spécification logique à partir du dessin proposé.

```

Début
.   initialisation : S=∅
.   Pour chaque élément E du dessin
.   .   Pour chaque élément Fi, Fi ≠ E du dessin
.   .   Recherche l'ensemble des propriétés géométriques pouvant
.   .   .   lier E et Fi, ajouter ces propriétés à S.
.   .   fin.
.   fin.
.   donner S
fin.

```

Figure III.4-k : Algorithme d'inférence.

Le principe général de l'algorithme d'exploitation du dessin consiste en un examen successif des différents éléments constituant le dessin et une recherche des propriétés géométriques susceptibles de lier les éléments entre eux.

Le dessin est considéré comme approximatif sur deux plans. D'une part, les objets qui sont définis sur le dessin sont pris avec une certaine approximation. D'autre part, les relations géométriques apparaissant sur le dessin sont considérées avec une certaine marge d'erreur. Pour la représentation de ces "approximations", l'arithmétique CLP(Intervalle) est avantageuse.

L'algorithme d'inférence d'une spécification logique S à partir d'un dessin peut se résumer comme à la figure III.4-k. Les propriétés envisagées sont :

- l'incidence entre un point et une droite ou entre un point et un cercle,
- le parallélisme entre deux droites,
- la perpendicularité entre deux droites,
- l'égalité de distance entre les rayons de deux cercles, ou entre quatre points pris deux à deux décrivant deux segments, et
- la relation de milieu entre un point et un bipoint pris comme un segment.

Conclusion : Les travaux engagés ici sont très prometteurs et fournissent des outils de dessin très agréables. Le domaine des réalités virtuelles est proche.

Conclusion de la troisième partie :

L'ensemble des applications de cette troisième partie tend vers la définition d'environnements informatisés d'apprentissage humain de la géométrie tels qu'ils apparaissent dans le chapitre III-4. Ces applications reposent effectivement sur le travail de la partie II. Les chapitres III-1, III-2 et III-3, qui portent également sur des applications de la partie II, ne sont pas directement tournés vers l'enseignement mais s'en approchent. Nous pensons que dans un avenir proche, les EIAH de géométrie comporteront des démonstrateurs automatiques et des modules de calcul formel. Alors, les travaux du III-1, III-2 et III-3 se trouveront naturellement dans l'objectif initial de ce travail : la définition d'EIAH de géométrie.

Conclusion générale du manuscrit

Avant d'aborder les perspectives de notre travail, résumons rapidement les résultats contenus dans ce manuscrit.

La première partie porte sur la géométrie de la règle et du compas. Elle contient une analyse de 512 situations de théorèmes classiques de la géométrie euclidienne. Cette analyse est effectuée en vue d'obtenir des constructions rationnelles telles qu'elles ont été présentées dans les chapitres I-2 et I-3. Les résultats expérimentaux et formels de cette étude sont très positifs, cf. chapitre I-4. Sur le plan expérimental, ils montrent que la complexité algébrique des situations géométriques n'est pas intrinsèque. La vaste majorité des 512 situations considérées peuvent s'exprimer sous la forme de figures rationnelles (492 situations sur 512). Seules 20 situations sur 512 ne sont pas rationnelles. Parmi ces 20 situations, 10 sont dans $Q[\sqrt{3}]$ à cause de triangles équilatéraux. C'est un cas très particulier d'extension algébrique dont on peut tirer profit. Sur le plan formel, cette étude révèle des heuristiques qui permettent l'obtention de ces figures rationnelles. Ces heuristiques sont reprises et développées au chapitre III-2.

La seconde partie contient l'essentiel du travail formel informatique de cette thèse. Il aboutit à la définition d'un environnement de programmation CLP(Géométrie) dont l'objectif est double : représenter de manière exacte le domaine des nombres constructible (cf. chapitre II-2) et construire automatiquement des figures géométriques respectant une spécification logique (cf. chapitre II-3). Ce double objectif est atteint avec succès sur la classe des spécifications constructives, (cf. chapitre II-4). Pour la classe générale des problèmes non-linéaires, l'obtention d'une représentation normale fournit une solution au premier problème. Ce résultat permet la définition d'un algorithme de vérification de contraintes redondantes exposé du chapitre III-2. Pour le problème général de la construction automatique, le chapitre II-4 montre que nous obtenons des résultats satisfaisants. Notre algorithme de construction automatique possède de plus de bonnes perspectives d'améliorations que nous reprenons en perspective.

Ces résultats intéressent la communauté CLP car les deux problèmes généraux de la résolution des systèmes de contraintes non-linéaires et de la vérification des contraintes redondantes sont très difficiles. Pour le moment ils n'ont pas reçu de solutions pratiques satisfaisantes. Notre approche concentrée sur la classe particulière et très importante des contraintes géométriques apporte une solution. De plus, cette solution est complète pour le problème de la vérification de contraintes redondantes.

Une autre approche pour la résolution du problème de satisfaction des systèmes de contraintes quadratique a été proposée par Gilles Pesant avec Quad-CLP(R). Cette solution est orthogonale à la notre. Elle propose de prendre une approximation des contraintes quadratiques par des contraintes linéaires. Malheureusement ces deux travaux ne peuvent pas être rapprochés. Le côté "approximation" de Quad-CLP(R) ne permet pas d'utiliser l'aspect "exact" de notre travail CLP(Géométrie).

La dernière partie porte sur la définition d'applications reposant sur l'environnement de programmation CLP(Géométrie).

En utilisant les résultats de la seconde partie, nous présentons une première application au chapitre III-1 qui porte sur la représentation des expressions arithmétiques avec racines carrées. Cette application permet l'obtention d'une écriture normale d'une expression donnée avec un nombre minimum de racines carrées distinctes et respectant au mieux l'expression initiale.

L'analyse des 512 situations géométriques de la première partie donne la matière d'une seconde application portant sur la vérification de systèmes de contraintes non-linéaires redondantes au chapitre III-2. Ce chapitre reprend les résultats du chapitre II-2 sur l'obtention d'une représentation normale des nombres constructibles, ce qui lui assure la correction et la complétude. De plus en profitant du travail effectué au chapitre I-4 sur la recherche de situations rationnelles, 4 heuristiques sont proposées pour rationaliser les spécifications géométriques. Parmi ces 4 heuristiques, 2 peuvent être immédiatement automatiser. Dans ce dernier cas, la complexité de notre approche permet une utilisation immédiate.

L'application précédente (vérification de contraintes redondantes) est poussée jusqu'à la démonstration numérique automatique en géométrie au chapitre III-3. Une nouvelle méthode de démonstration automatique numérique est proposée qui utilise la méthode de vérification de systèmes de contraintes redondants du chapitre III-2, l'arithmétique exacte du chapitre II-2 et une méthode d'estimation originale, (cf. section III.3-b). Cette méthode de démonstration automatique est évaluée sur 4 exemples.

L'ensemble des applications précédentes tendent vers l'objectif initial de ce travail : la réalisation d'environnements informatisés d'apprentissage humain de la géométrie (EIAH de géométrie). Cet objectif se concrétise principalement au chapitre III-4 avec la présentation de trois propositions d'EIAH de géométrie :

- un EIAH de géométrie dynamique avec une approche déclarative pour la spécification des figures géométriques,
- un EIAH d'aide à l'enseignant, pour la détection d'erreur dans une figure d'élève et la construction de contre-exemples sous forme d'animation de la figure de l'élève,
- un EIAH de dessin géométriques à main levé par inférence automatique de la spécification de la figure.

En particulier la première réalisation "GéoSpécif" utilise pleinement les capacités de CLP(Géométrie). Et c'est normal, c'est la réalisation de cette application qui a nécessité le travail présent.

Parmi les perspectives que ce travail suscite, celles qui nous paraissent les plus intéressantes sont :

sur le plan théorique,

- la possibilité d'étendre la représentation normale des nombres constructibles aux corps finis (cf. conclusion du II-2). Ceci permet d'éviter la complexité de la représentation des nombres rationnels en précision infinie. Cependant, cette extension aux corps finis est délicate. D'une part la notion d'ordre n'est pas définie pour les corps finis. D'autre part l'égalité est définie modulo l'ordre du corps et par conséquent deux nombres peuvent avoir le même représentant.

- la possibilité d'étendre l'algorithme de résolution de systèmes de contraintes non-linéaires. Sur le plan algébrique cette extension peut se faire en propageant les relations quadratiques. Sur le plan géométrique elle peut s'effectuer en ajoutant les cercles où apparaissent des angles droits. La correction de la résolution est préservée grâce à la vérification automatique des contraintes redondantes (cf. conclusion du II-3). Les améliorations "géométriques" de l'algorithme de construction automatique ont de plus l'avantage d'être explicables simplement en termes géométriques.

- la possibilité d'obtenir des démonstrations automatiques numériques exactes de théorèmes en géométrie euclidienne (cf. III-2 et III-3). De plus des démonstrations approchées sont envisageables. Avec une démarche probabiliste, en acceptant la probabilité d'une erreur due à un cas particulier, la complexité est celle de la vérification sur un exemple quelconque. Avec une démarche formelle, en reprenant la démarche de Jiwaie Hong, la complexité reste polynomial et évite l'emploi des algorithmes usuels de complexité exponentiel, type Gröbner.

sur le plan pratique,

- la possibilité de résoudre des contraintes générales en prenant en compte le fait que dans la pratique seulement de "petites" extensions apparaissent.

- la possibilité de définir des environnements d'apprentissage de la géométrie plus "intelligents" en intégrant d'une part un algorithme de spécification par inférence automatique à partir d'un dessin à main levée (cf. III.4-c), d'autre part un algorithme de construction et d'animation automatique de figures géométriques (cf. III.4-a), un démonstrateur automatique de propriétés géométriques (cf. III-3), et enfin une aide au diagnostique et à la production de contre-exemples (cf. III.4-b). Ces travaux sont tous définis dans ce manuscrit et déjà réalisés (sauf III-3). D'autres travaux auxquels nous n'avons pas participé existent aussi : production d'une construction à la règle et au compas à partir d'une construction algébrique, production de preuves compréhensibles etc... Il "faut" maintenant les organiser les uns avec les autres pour arriver à un EIAH de géométrie encore un peu plus intelligent.

Pour une conclusion plus globale.

L'enseignement de la géométrie passe aujourd'hui par l'informatique. Le succès de Cabri-Géomètre en est la preuve. Notre travail montre que l'informatique peut fournir des outils de qualité pour compléter la panoplie des EIAH de géométrie. Elle permet apporter deux éléments importants : la déclarativité dans la définition des figures et l'exactitude numérique de ces

figures. Ces deux éléments prolongent l'effort de Cabri-Géomètre pour définir une géométrie dynamique. Ils enrichissent le concept de manipulation directe en assurant la correction des figures.

L'un des enseignements que l'informatique peut tirer de ce travail est le nécessaire effort multidisciplinaire pour tout projet où l'informaticien n'est engagé à priori que comme un exécutant. La tâche de l'informaticien ne peut se restreindre à l'informatique seule. Le travail que nous présentons, chargé d'une réflexion importante sur la géométrie, en est un exemple modeste.

Mais ce n'est pas tout à notre avis, non seulement l'informaticien doit produire un travail multidisciplinaire, mais aussi il doit mener son travail sur deux axes orthogonaux, formels et expérimentaux. Notre travail montre l'importance d'une double démarche formelle et expérimentale pour "attaquer" un problème particulier. L'effort formel de la partie II, et la découverte d'une représentation normale, ne serait rien sans les résultats expérimentaux de la partie I. L'analyse de 512 problèmes de géométrie montre que cette représentation est applicable dans le contexte de l'enseignement.

Enfin sur la nature des problèmes que pose l'enseignement, il n'est pas étonnant que ceux-ci soient difficiles scientifiquement. Ils correspondent au corpus des sciences, de la Grèce antique jusqu'à aujourd'hui. C'est un domaine d'un grand intérêt pour l'informatique car il lui permet de s'enrichir de l'intérieur, en lui posant des problèmes à résoudre dans un contexte riche de plusieurs siècles.

Bibliographies

Bibliographie personnelle:

[Allen, Bouhineau, Trilling-93]

Richard Allen and Denis Bouhineau and Laurent Trilling

Construction and manipulation of geometric figures

WCLP'93-Third Workshop on Constraint Logic Programming, Marseille,
Mars 1993.

[Bouhineau-95a]

Denis Bouhineau,

Vers une approche déclarative pour les logiciels de dessins géométriques,

Actes des Quatrièmes journées EIAO de Cachan, Environnements Interactifs
d'Apprentissage avec Ordinateur, ed Eyrolles, Mars 1995.

[Bouhineau-95b]

Denis Bouhineau

Verification of non-linear overconstraints in Euclidean geometry with linear programming

Proceedings from International Workshop on Over Constraints Systems
OCS'95 in association with the First international Conference on Constraint
programming CP'95, Cassis, Sept, 1995.

[Bouhineau-96]

Denis Bouhineau,

*Solving Geometrical Constraint Systems Using CLP Based on Linear
Constraint Solver,*

Third International Conference on Artificial Intelligence and Symbolic
Mathematical Computation, AISMC-3, Steyr, Austria, LNCS 1098, sept 1996.

[Bouhineau, Channac-96]

Denis Bouhineau, Stéphane Channac,

La programmation logique avec contraintes pour l'aide à l'enseignant

Proceedings of the Third International Conference on Intelligent Tutoring System, ITS'96, Montreal, Canada, Claude Frasson and Gilles Gauthier and Alan Lesgold Eds, Springer-Verlag, LNCS 1086, pp 333-342, Juin 1996.

Bibliographie internationale:

[Allen, Idt, Trilling-93]

Richard Allen, Jeanne Idt, Laurent Trilling,

Constrained based automatic construction and manipulation of geometric figures,

Proceedings of the 13th IJCAI Conference, Chambéry, Morgan Kaufmann Publishers, Los Altos, Aout, 1993.

[Allen, Nicolas, Trilling-90]

Richard Allen, Pierrick Nicolas et Laurent Trilling,

Figure Correctness in an expert System for Teaching Geometry,

Proceedings of the eight biennial conference of the Canadian society for computer studies of intelligence, Ottawa, 1990.

[Allen, Trilling-95]

Richard Allen and Laurent Trilling,

Dynamic Geometry and Declarative Geometric Programming,

M AA (Mathematical Association of America) and AMS (American Mathematical Society) Joint Meeting, Contributed Papers, San Francisco, Janvier,1995

[Balbiani, Dugat, Fariñas del Cerro, Lopez-94]

Philippe Balbiani, Vincent Dugat, Luis Fariñas del Cerro, Anne Lopez,

Eléments de géométrie mécanique,

Editions Hermès, Paris, 1994.

[Baulac-90]

Yves Baulac,

Un micro-monde de géométrie - Cabri- Géomètre.,

Université Joseph Fourier - Grenoble, 1990

[Baulac, Bellemain, Laborde-92]

Baulac, Y., Bellemain, F., Laborde, J.M.,
Cabri, the interactive geometry note,
Brooks/Cole publishing company, Pacific grove, CA, 1992.

[Baulac, Laborde-89]

Yves Baulac et Jean-Marie Laborde,
Sur l'interface d'un Cahier de brouillon informatique pour la géométrie.,
LSD2 - IMAG, 1989

[Bellemain-92]

F. Bellemain,
Conception, réalisation et expérimentation d'un logiciel d'aide à l'enseignement de la géométrie, Cabri-Géomètre.,
Université Joseph Fourier, 1992

[Benhamou-94]

Frédéric Benhamou,
Interval Constraint Logic Programming.,
Constraint Programming : Basics and Trends, pp 1-21, May , 1994.

[Benhamou et Touraivane-95]

Frédéric Benhamou and Touraivane,
Prolog IV : langage et algorithmes.,
Journée Francophones de la Programmation Logique, 1995.

[Bonnet-93]

Jean-François Bonnet,
Témoignage : j'ai rencontré des Cabri-élèves,
Université d'été, Apprentissage et enseignement de la géométrie avec ordinateur : Utilisation du logiciel Cabri-Géomètre en classe, ed LSD2-IMAG, IREM/IUFM de Grenoble, 1993.

[Borning-81]

Borning, A.,
The programming language aspect of ThingLab, a constraint-oriented simulation laboratory,
ACM TOPLAS, vol 3, no.4, 1981.

[Borodin, Fagin, Hopcroft, Tompa-85]

Borodin, A., Fagin, R., Hopcroft, J.E., Tompa, M.,
Decreasing the nested depth of expression involving square roots,
Journal of Symbolic Computation, 1 pp 169-188, 1985.

[Bourbaki-69]

Nicolas Bourbaki,
Eléments d'histoire des mathématiques,
Hermann, Paris, 1969.

[Buthion-75]

Michel Buthion,

Un programme qui résout formellement des problèmes de constructions géométriques.,

Université Pierre et Marie Curie - Paris VI, 1975.

[Cabri-93]

Actes de l'Université d'été : Apprentissage , et enseignement de la géométrie avec ordinateur. Utilisation du logiciel Cabri-Géomètre en classe,

édition Lsd2-IMAG, IREM/IUFM de Grenoble, Juillet, 1993.

[Carrega-89]

Jean-Claude Carrega,

Théorie des corps, la règle et le compas,

réédition, Herman, Paris, 1989.

[Cederberg-89]

Judith N., Cederberg,

A Course in Modern Geometries,

Springer-Verlag, New-York, 1989.

[Chou-88]

Shang-Ching Chou,

Mechanical Geometry Theorem Proving,

Reidel Publishing, Norwell, 1988.

[Chou et Gao-92]

S.C. Chou and X.S. Gao,

Proving Geometry Statement of Constructive Type,

CADE LNCS 607, D. Kapur Eds, 1992.

[Chou, Gao et Zhang-94]

S.C. Chou, X.S. Gao, et X.S. Zhang

Machine proofs in geometry.

World Scientific, Singapore, 1994.

[Cohen-90]

Jacques Cohen,

Constraint logic programming languages

Communication of the ACM vol. 33, no. 7 page 52--68, juillet 1990.

[Colmerauer-90]

Alain Colmerauer,

An introduction to Prolog III,

Communication of the ACM vol. 33, no. 7 page 69--90, juillet 1990.

[Colmerauer-93]

Alain Colmerauer,

Naïve Solving of Non-linear Constraints,

Constraint Logic Programming : Selected Research, The MIT Press, Frédéric Benhamou et Alain Colmerauer éditeurs, pp 89-112, 1993.

[Dahan-Damedico, Peiffer-86]

Alain Dahan-Dalmedico, Jean Peiffer,

Une histoire des mathématiques,

Collections Points, serie science, Mars 1986.

[Davis-77]

Davis, P.J.,

Proof, Completeness, Transcendental, and Sampling,

Journal of the ACM, vol. 24, No. 2, pp 298--310, 1977.

[Dedron, Itard-59]

Pierre Dedron et Jean Itard,

Mathématiques et mathématiciens

Editions Magnard, 1959.

[Deng, Zhang, Yang-90]

Mike Deng, Jingzhong Zhang, Lu Yang

The parallel numerical method of mechanical theorem proving

Theoretical Computer Science No.74 pp. 253--271, 1990.

[Desmoulins-94]

Cyrille Desmoulins,

Etude et réalisation d'un système tuteur pour la construction de figures géométriques

thèse de Doctorat de l'Université Joseph Fourier, Grenoble, Février 1994.

[Desmoulins-95]

Cyrille Desmoulins,

La détection de solutions particulières dans TALC : une approche basée sur des extensions de l'énoncé du problème,

Quatrièmes Journées EIAO de Cachan, Environnements Interactifs d'Apprentissage avec Ordinateurs, Eyrolles, Mars, 1995

[Dincbas, Van Hentenryck, Simonis, Aggoun, Graf, et Berthier-88],

M. Dincbas, Pascal Van Hentenryck, H. Simonis, A. Aggoun, T. Graf, et F. Berthier,

The Constraint Logic Programming Language CHIP,

Proceedings of the International Conference on 5 Fifth Generation Computer Systems, Tokyo, Japan, December, 1988.

[Dufourd-90]

Jean-François Dufourd,

Programmation et résolution de problèmes de constructions géométriques
Actes des journées du GRECO-Programmation, Bigre-Globule, no. 70,
1990.

[Fortune, Van Wyk-93]

Steven Fortune et Christopher J. Van Wyk,

Efficient Exact Arithmetic for Computational Geometry
Proceedings of the 9th ACM Symposium on Computational Geometry, pp
163--172, San Diego, February 1993.

[Gelernter-63]

H. Gelernter,

Computer and Thought, Realization of a geometry theorem proving machine,
Mac Graw-Hill, eds E.A. Feigenbaum and J. Feldman, pp 134--152, New-
York, 1963.

[Guillot, Jolivet, Leperd, Moulin-91]

Guillot, P., Jolivet, L., Leperd, J.L., Moulin C.,

Démonstration automatique en géométrie
l'Université d'été Informatique et Enseignement de la géométrie, Irem de
Toulouse editeur, 1991.

[Jaffar et Lassez-87]

Joxan Jaffar et Jean-Louis Lassez

Constraint Logic Programming,
Proceedings of the fourteenth ACM Symposium of the principles of
Programmings languages, Munich, pp 111-119, 1987.

[Jaffar, Michaylov, Stuckey, et Yap-92]

Joxan Jaffar, S. Michaylov, P-J. Stuckey, et R. Yap,

The CLP(R) Language and System,
ACM Trans. on Programming Languages and Systems, vol. 14, No. 3, pp
339--395, 1992.

[Kramer-92]

Glenn A. Kramer,

Solving Geometric Constraint systems, a Case Study in Kinematics,
The MIT Press, 1992.

[Hong-86]

Hong, Jiawei,

Proving by Example and Gap Theorems,
pp 107--116, 27th Annual Symposium on Foundations of Computer Science,
Toronto, Ontario, Canada, IEEE, 1, oct, 1986.

[Laborde-86]

Jean-Marie Laborde,

Projet d'un Cahier de Brouillon Informatique de Géométrie.,
LSD2 - IMAG, 1986

[Laborde-93]

Colette Laborde,

Apprendre à voir et manier l'objet géométrique au delà du tracé dans Cabri-Géomètre,

Université d'été, Apprentissage et enseignement de la géométrie avec ordinateur~: Utilisation du logiciel Cabri-Géomètre en classe, ed LSD2-IMAG, IREM/IUFM de Grenoble, 1993.

[Laborde-95]

Jean-Marie Laborde,

Des connaissances abstraites aux réalités artificielles, le concept de micromonde Cabri.,

Actes des IVemes Journées EIAO de Cachan, ed Eyrolles, 1995.

[Landau-92]

Landau, S.,

Simplification of nested radicals,

SIAM Journal of Computing Vol 21.N1 pp85-110, Feb 1992.

[Lebesgue-89]

Lebesgue, H.

Leçons sur les constructions géométriques,
réédition, Gauthier-Villard, Paris, 1989.

[Michelucci, Moreau-94]

Dominique Michelucci et J-M. Moreau

Lazy Arithmetic

Rapport interne, Ecole des Mines de St-Etienne, 1995.

[Marti-95]

Philippe Marti,

Une Implémentation d'un Système Coopératif pour la Résolution de Contraintes sur les Réels.,

IVemes Journées Francophones de Programmation en Logique, pp 373-376, Mai, 1995.

[Mutafian-80]

Claude Mutafian,

Equations algébriques et théorie de Galois,
Vuibert, 1980.

[Muller-89]

Jean-Michel Muller,

Arithmétique des ordinateurs, opérateurs et fonctions élémentaires
Masson, 1989.

[Naudin, Quitté-85]

Naudin, P., Quitté, C.,

Algorithmique algébrique,

Masson, Collection Logique mathématiques informatique, 1985.

[N'Dong et Van Caneghem-95]

Stéphane N'Dong et Michel Van Caneghem,

Résolution de problèmes continus en programmation logique avec contraintes avec les intervalles.,

IVemes Journées Francophones de Programmation en Logique, pp 79-93, Mai 1995.

[Older et Benhamou-93a]

William J. Older and Frédéric Benhamou,

Applying Interval Arithmetic to Integer and Boolean Constraints.,

Bell-Northern Research, P.O. Box 3511, Station C K1Y 4H7, Ottawa, Ontario, June, 1993.

[Older et Benhamou-93b]

William J. Older and Frédéric Benhamou,

Programming in CLP(BNR).,

Proceedings of the Second Workshop on Principles and Practice of Constraint Programming (PPCP'93), Janvier 1993.

[Older et Vellino-90]

William J. , Older and André Vellino

Extending Prolog with Constraint Arithmetic on Real Intervals.,

Proceedings of the Canadian Conference on Electrical and Computer Engineering, 1990.

[Older et Vellino-93]

Older, W., Vellino, A.,

Constraint arithmetic on real Intervals

Constraint logic programming, selected research, edited by F. Benhamou and A. Colmerauer, the Mit Press, 1993.

[Pesant et Boyer-94]

Gilles Pesant and Michel Boyer,

Quad- Clp(R) : Adding the Power of Quadratic Constraints.,

Proceedings of the Second Workshop on Principles and Practice of Constraint Programming (PPCP'94), 1994.

[Pesant-95]

Gilles Pesant,

Une approche géométrique aux contraintes arithmétiques quadratiques en programmation logique avec contraintes.,

Thèse de l'Université de Montréal, 1995.

- [Py-90]
Py, D.,
Mentoniez, a geometry I.T.S. for drawing and prof setting
Journal of Artificial Intelligence in Education, Vol 1, No 3, 1990.
- [Robinson-65]
Robinson, J.A.,
A Machine-Oriented logic based on the resolution principle,
J. ACM 12, pages 23-41, 1965.
- [Rolfe-87]
Rolfe, T.,
On a fast integer square root algorithm,
ACM SIGNUM Newsletter, vol.22, No. 4, pp 6--11, 1987.
- [Sakai et Aiba-89]
Sakai K., et Aiba Akira,
CAL : A Theoretical Background of Constraint Logic Programming and its Application,
J. Symbolic Computation, 8 pp 589-603, 1989.
- [Sketchpad-90]
Geometer's Sketchpad,
Key Curriculum Press, Berkeley CA, 1990.
- [Schreck-92]
Schreck, P.,
Automatisation des construction géométriques à la règle et au compas
Thèse de l'Université Louis Pasteur, Strasbourg, 1992.
- [Tarski-51]
Tarski, A.,
A decision method for elementary algebra and geometry
University of California Press, second edition, Los Angeles, 1951.
- [Trilling-95]
Laurent Trilling,
On dynamic geometry and Logic Geometric Programming,,
International Workshop on Constraints for graphics and Visualisation CGV'95 in association with the First international Conference on Constraint programming CP'95, CassisSeptembre,1995
- [Trilling-95]
Laurent Trilling,
Programmation géométrique, impérative et logique,
pp 147-164, JFLA'96, Journées francophones des langages applicatifs, INRIA Collection Didactique, Montreal, Canada, Février 1996.

[Van Hentenryck-92]

Pascal Van Hentenrych,

Constraint satisfaction using constraint logic programming.,
Artificial Intelligence, vol. 58, pp 113-159, 1992.

[Van Hentenryck-95]

Pascal Van Hentenrych,

Constraint Solving for Combinatorial Search Problems : A Tutorial.,
Principles and Practice of Constraint Programming-CP'95, Cassis, France,
Springer Verlag LNCS 976, Ugo Montanari, Francesca Rossi Eds, Septembre
1995.

[Van Hentenryck, Michel-95]

P. Van Hentenryck, L. Michel,

The Design and Implementation of Newton.
CCP'95, Venice, Italy, May 1995.

[Wang-96]

Dongming Wang,

Geometry : from SMC to AI
Third International Conference on Artificial Intelligence and Symbolic
Mathematical Computation, AISMC-3, Steyr, Austria, Springer Verlag LNCS
1098, sept 1996.

[Wolfram-91]

Wolfram, S.,

Mathematica: a system for doing mathematics by computer,
Addison-Wesley, 1991.

[Wu-94]

Wu, Wen-Tsun,

Mechanical Theorem Proving in Geometries,
Springer-Verlag Wien New-York, 1994.

[Yap, Dubé-94]

Chee Yap, and Thomas Dubé,

The exact computation paradigm,
Computing in Euclidean Geometry, World Scientific Press, eds D.Z. Du,
F.K. Hwang, 1994.

[Zippel-85]

Zippel, R.,

Simplification of expression involving radicals,
Journal of Symbolic Computation, pp 189-210, (1) 1985.

Bibliographie locale :

(principalement de l'équipe PLIAGE LSR-IMAG,
mais aussi du Projet Cabri, Leibniz/LSR-IMAG)

[Balacheff-95]

Nicolas Balacheff,

*TéléCabri : principe d'une machine partenaire du formateur dans un contexte
d'apprentissage distant utilisant la téléprésence,*

Page Web de Présentation du Groupe EIAH, 1995.

[Capponi-94]

Bernard Capponi,

Constructions, contraintes,

communication personnelle, juin 94.

[Channac-95]

Stéphane Channac,

*Etude et réalisation d'un constructeur de figure géométriques basé sur un
résolveur de contraintes quadratiques approchées,*

Stage de DEA Informatique, Université Joseph Fourier, Septembre, 1995

[Lixi-96]

Patrick Lixi,

Démonstration automatique de constructions géométriques.

Stage de Magistère de Mathématiques, UJF, Grenoble, Sept. 1996.

[Nebon-96]

Florence Nebon,

Inférence automatique de spécifications logiques à partir de dessins,

INPG-CUEFA, Mémoire d'ingénieur CNAM, Grenoble, Mars 1996.

[Rouchon-94]

Rouchon, P.,

*Etude et réalisation d'un vérificateur de contre-propriétés en programmation
logique avec contraintes sur les intervalles.*

Rapport de stage de maîtrise, Université Joseph Fourier, 1994.

Annexes

- Annexe 1-

construction rationnelle du milieu d'un segment à la règle et au compas (exprimée avec la syntaxe Cabri réduite)

"A" :3
"B" :4
5: Segment 3 4
point qlcq :6
7: Ligne 3 6
8: Cercle 6 3
9: Intersection 7 8
10: Ligne 9 4
11: Intersection 10 8
12: Cercle 11 6
13: Cercle 9 6
14: Intersection 12 13
15: Ligne 14 6
16: Ligne 6 11
17: Point sur 7
18: Cercle 6 17
19: Intersection 16 18
20: Cercle 19 6
21: Cercle 17 6
22: Intersection 20 21
23: Ligne 6 22
24: Intersection 5 23

- Annexe 2-

construction du milieu d'un segment au compas (exprimée avec la syntaxe Cabri réduite)

"A" 3:
"B" 4:
5: Segment 3 4
6: Cercle 4 3
7: Cercle 3 4
8: Intersection 6 7
9: Cercle 8 3
10: Intersection 6 9
11: Cercle 10 4
12: Intersection 6 11
13: Cercle 12 3
14: Intersection 7 13
15: Cercle 14 3
16: Intersection 7 13
17: Cercle 16 3
"M" 18: Intersection 15 17

- Annexe 3-

liste des complexités.

Colonne 0 : le numéro du théorème dans [Chou-88]

Colonne 1 : le nombre d'équations du second degré dans ce théorème

Colonne 2 : le nombre d'extensions dans la construction proposée par [Chou-88] dans ce théorème

Colonne 3 : le nombre d'extensions dans la construction proposée par [Chou-88] modifiée localement (cf chapitre III-2)

Colonne 4 : le nombre d'extensions dans la construction proposée par [Chou-88] modifiée globalement (cf chapitre III-2)

				9	0	0	0	0	18	0	0	0	0	
1	0	0	0	0	10	5	3	0	0	19	5	5	0	0
2	0	0	0	0	11	5	3	0	0	20	0	0	0	0
3	0	0	0	0	12	5	3	0	0	21	1	1	0	0
4	0	0	0	0	13	5	3	0	0	22	0	0	0	0
5	0	0	0	0	14	5	3	0	0	23	0	0	0	0
6	6	6	0	0	15	0	0	0	0	24	0	0	0	0
7	5	5	0	0	16	0	0	0	0	25	3	0	0	0
8	0	0	0	0	17	0	0	0	0	26	0	0	0	0

27	0	0	0	0	76	0	0	0	0	125	4	3	2	1
28	0	0	0	0	77	5	3	0	0	126	2	0	0	0
29	4	0	0	0	78	0	0	0	0	127	0	0	0	0
30	2	0	0	0	79	6	3	2	0	128	3	1	1	1
31	2	0	0	0	80	8	2	0	0	129	2	2	1	1
32	2	0	0	0	81	0	0	0	0	130	4	0	0	0
33	0	0	0	0	82	0	0	0	0	131	4	1	1	0
34	0	0	0	0	83	1	1	0	0	132	2	2	1	1
35	5	1	1	1	84	0	0	0	0	133	0	0	0	0
36	3	1	0	0	85	0	0	0	0	134	0	0	0	0
37	2	0	0	0	86	1	1	0	0	135	3	3	3	0
38	0	0	0	0	87	4	1	1	1	136	1	1	1	0
39	1	1	1	1	88	1	1	0	0	137	3	1	1	0
40	1	1	1	1	89	1	1	1	1	138	3	1	1	0
41	0	0	0	0	90	4	2	0	0	139	3	1	1	0
42	0	0	0	0	91	1	1	0	0	140	2	0	0	0
43	10	1	1	1	92	2	0	0	0	141	2	0	0	0
44	12	1	1	1	93	3	0	0	0	142	0	0	0	0
45	10	1	1	1	94	3	2	1	0	143	4	0	0	0
46	0	0	0	0	95	1	1	1	0	144	3	3	0	0
47	0	0	0	0	96	5	1	0	0	145	0	0	0	0
48	4	2	0	0	97	2	2	1	0	146	0	0	0	0
49	0	0	0	0	98	3	0	0	0	147	0	0	0	0
50	0	0	0	0	99	4	1	0	0	148	0	0	0	0
51	0	0	0	0	100	1	0	0	0	149	0	0	0	0
52	0	0	0	0	101	2	0	0	0	150	0	0	0	0
53	0	0	0	0	102	2	0	0	0	151	2	0	0	0
54	0	0	0	0	103	2	0	0	0	152	2	2	2	0
55	0	0	0	0	104	5	1	1	1	153	1	1	1	0
56	0	0	0	0	105	2	1	0	0	154	0	0	0	0
57	0	0	0	0	106	5	2	1	1	155	0	0	0	0
58	0	0	0	0	107	4	2	0	0	156	0	0	0	0
59	0	0	0	0	108	3	3	1	0	157	1	0	0	0
60	0	0	0	0	109	4	2	0	0	158	2	0	0	0
61	0	0	0	0	110	1	1	1	0	159	0	0	0	0
62	0	0	0	0	111	4	2	2	0	160	2	0	0	0
63	2	2	0	0	112	2	2	1	0	161	0	0	0	0
64	1	1	0	0	113	3	2	2	0	162	5	3	3	0
65	9	0	0	0	114	1	1	0	0	163	5	3	3	0
66	6	0	0	0	115	4	1	0	0	164	0	0	0	0
67	0	0	0	0	116	5	1	0	0	165	0	0	0	0
68	0	0	0	0	117	3	2	1	0	166	0	0	0	0
69	0	0	0	0	118	4	0	0	0	167	2	0	0	0
70	4	2	2	0	119	5	2	1	1	168	2	0	0	0
71	3	3	0	0	120	0	0	0	0	169	0	0	0	0
72	6	0	0	0	121	0	0	0	0	170	2	0	0	0
73	2	2	0	0	122	0	0	0	0	171	0	0	0	0
74	0	0	0	0	123	4	2	0	0	172	2	2	2	0
75	1	1	0	0	124	3	2	0	0	173	0	0	0	0

174	2	0	0	0	223	5	0	0	0	272	3	1	0	0
175	4	0	0	0	224	0	0	0	0	273	3	1	0	0
176	1	1	1	0	225	5	0	0	0	274	3	1	0	0
177	0	0	0	0	226	0	0	0	0	275	3	1	0	0
178	0	0	0	0	227	0	0	0	0	276	3	1	0	0
179	3	0	0	0	228	0	0	0	0	277	4	2	0	0
180	4	0	0	0	229	0	0	0	0	278	9	1	0	0
181	2	0	0	0	230	2	0	0	0	279	0	0	0	0
182	0	0	0	0	231	0	0	0	0	280	0	0	0	0
183	2	0	0	0	232	3	0	0	0	281	3	0	0	0
184	5	0	0	0	233	0	0	0	0	282	3	1	0	0
185	0	0	0	0	234	0	0	0	0	283	3	0	0	0
186	0	0	0	0	235	2	0	0	0	284	3	0	0	0
187	0	0	0	0	236	5	0	0	0	285	3	0	0	0
188	0	0	0	0	237	4	0	0	0	286	3	0	0	0
189	2	0	0	0	238	2	0	0	0	287	4	0	0	0
190	2	0	0	0	239	4	0	0	0	288	3	1	0	0
191	2	0	0	0	240	10	4	0	0	289	9	1	0	0
192	2	0	0	0	241	4	0	0	0	290	9	1	0	0
193	2	0	0	0	242	2	0	0	0	291	6	0	0	0
194	2	0	0	0	243	10	1	1	1	292	4	1	0	0
195	3	1	0	0	244	6	0	0	0	293	3	1	0	0
196	2	0	0	0	245	2	0	0	0	294	5	1	0	0
197	2	0	0	0	246	0	0	0	0	295	3	0	0	0
198	4	0	0	0	247	0	0	0	0	296	3	1	0	0
199	4	0	0	0	248	2	0	0	0	297	3	0	0	0
200	4	0	0	0	249	0	0	0	0	298	5	1	1	0
201	4	0	0	0	250	3	1	0	0	299	3	0	0	0
202	4	0	0	0	251	0	0	0	0	300	3	1	1	0
203	3	1	1	0	252	0	0	0	0	301	3	1	0	0
204	2	0	0	0	253	0	0	0	0	302	4	2	0	0
205	0	0	0	0	254	0	0	0	0	303	4	2	0	0
206	2	0	0	0	255	0	0	0	0	304	5	1	0	0
207	2	0	0	0	256	0	0	0	0	305	4	2	0	0
208	6	1	1	1	257	1	0	0	0	306	5	1	0	0
209	2	0	0	0	258	1	1	1	0	307	5	3	0	0
210	2	0	0	0	259	2	0	0	0	308	4	0	0	0
211	2	0	0	0	260	0	0	0	0	309	8	0	0	0
212	8	0	0	0	261	0	0	0	0	310	6	2	0	0
213	0	0	0	0	262	0	0	0	0	311	3	1	0	0
214	2	0	0	0	263	0	0	0	0	312	0	0	0	0
215	6	0	0	0	264	0	0	0	0	313	3	1	0	0
216	4	0	0	0	265	3	1	0	0	314	2	0	0	0
217	8	0	0	0	266	3	1	0	0	315	9	1	0	0
218	2	0	0	0	267	3	1	0	0	316	2	2	0	0
219	4	1	1	0	268	3	1	0	0	317	3	1	0	0
220	3	1	1	0	269	5	1	0	0	318	3	1	0	0
221	2	0	0	0	270	3	1	0	0	319	2	0	0	0
222	1	1	1	0	271	3	1	0	0	320	0	0	0	0

321	0	0	0	0	370	2	2	0	0	419	2	0	0	0
322	0	0	0	0	371	2	2	0	0	420	3	0	0	0
323	0	0	0	0	372	4	2	2	0	421	0	0	0	0
324	0	0	0	0	373	3	2	0	0	422	0	0	0	0
325	2	0	0	0	374	2	0	0	0	423	0	0	0	0
326	0	0	0	0	375	2	0	0	0	424	3	2	1	0
327	0	0	0	0	376	7	0	0	0	425	3	0	0	0
328	0	0	0	0	377	3	3	0	0	426	3	0	0	0
329	0	0	0	0	378	3	3	1	0	427	1	1	0	0
330	0	0	0	0	379	3	3	1	0	428	5	0	0	0
331	4	2	0	0	380	1	1	0	0	429	4	0	0	0
332	0	0	0	0	381	0	0	0	0	430	0	0	0	0
333	1	1	1	0	382	2	0	0	0	431	0	0	0	0
334	0	0	0	0	383	2	0	0	0	432	3	1	1	0
335	0	0	0	0	384	0	0	0	0	433	2	0	0	0
336	0	0	0	0	385	0	0	0	0	434	2	0	0	0
337	0	0	0	0	386	0	0	0	0	435	0	0	0	0
338	0	0	0	0	387	0	0	0	0	436	0	0	0	0
339	4	0	0	0	388	0	0	0	0	437	0	0	0	0
340	0	0	0	0	389	0	0	0	0	438	0	0	0	0
341	0	0	0	0	390	3	3	0	0	439	0	0	0	0
342	0	0	0	0	391	4	0	0	0	440	2	0	0	0
343	0	0	0	0	392	2	0	0	0	441	0	0	0	0
344	0	0	0	0	393	5	1	1	1	442	0	0	0	0
345	0	0	0	0	394	3	1	1	0	443	0	0	0	0
346	0	0	0	0	395	3	2	1	0	444	0	0	0	0
347	2	0	0	0	396	5	0	0	0	445	4	0	0	0
348	0	0	0	0	397	0	0	0	0	446	0	0	0	0
349	5	0	0	0	398	3	2	1	0	447	0	0	0	0
350	0	0	0	0	399	0	0	0	0	448	2	0	0	0
351	0	0	0	0	400	4	0	0	0	449	4	0	0	0
352	2	0	0	0	401	4	0	0	0	450	4	0	0	0
353	0	0	0	0	402	0	0	0	0	451	4	0	0	0
354	0	0	0	0	403	0	0	0	0	452	5	1	1	0
355	0	0	0	0	404	0	0	0	0	453	2	1	1	1
356	0	0	0	0	405	2	0	0	0	454	2	0	0	0
357	1	1	1	0	406	4	0	0	0	455	0	0	0	0
358	0	0	0	0	407	2	1	1	0	456	5	1	1	0
359	0	0	0	0	408	3	0	0	0	457	4	2	1	0
360	0	0	0	0	409	3	2	1	0	458	5	2	2	0
361	0	0	0	0	410	3	1	0	0	459	4	0	0	0
362	0	0	0	0	411	0	0	0	0	460	3	0	0	0
363	1	1	1	0	412	0	0	0	0	461	5	1	1	0
364	4	0	0	0	413	2	0	0	0	462	4	1	0	0
365	0	0	0	0	414	0	0	0	0	463	0	0	0	0
366	0	0	0	0	415	3	1	0	0	464	0	0	0	0
367	3	0	0	0	416	0	0	0	0	465	0	0	0	0
368	1	0	0	0	417	4	0	0	0	466	0	0	0	0
369	3	2	0	0	418	3	0	0	0	467	3	0	0	0

468	0	0	0	0	484	5	0	0	0	500	2	0	0	0
469	0	0	0	0	485	4	0	0	0	501	8	0	0	0
470	5	1	0	0	486	8	0	0	0	502	0	0	0	0
471	3	1	0	0	487	8	0	0	0	503	0	0	0	0
472	0	0	0	0	488	10	0	0	0	504	4	1	1	0
473	0	0	0	0	489	4	0	0	0	505	4	1	1	0
474	0	0	0	0	490	8	0	0	0	506	0	0	0	0
475	0	0	0	0	491	6	0	0	0	507	0	0	0	0
476	5	0	0	0	492	5	2	2	1	508	2	0	0	0
477	0	0	0	0	493	2	0	0	0	509	3	0	0	0
478	2	2	2	0	494	0	0	0	0	510	5	0	0	0
479	0	0	0	0	495	5	1	0	0	511	2	0	0	0
480	2	0	0	0	496	3	1	0	0	512	3	1	0	0
481	0	0	0	0	497	3	1	0	0					
482	0	0	0	0	498	2	0	0	0					
483	5	0	0	0	499	3	0	0	0					

Liste des situations non rationnelles :

- triangles équilatéraux : 35, 39, 40, 43, 44, 45, 87, 89, 104, 119
- autres : 106, 125, 128, 129, 132, 208, 243, 393, 453, 492.

- Annexe 4-

annexe citée dans le texte :

démonstration d'un théorème au chapitre I-4 en utilisant le schéma de la démonstration de la "normalité" de la représentation du chapitre II-2.

Le schéma de réduction utilisé (transformation d'équations dans K_n en système d'équation dans K_{n-1} , et résolution récursive) permet de démontrer le résultat admis en première partie que nous reprenons ici

Théorème : $\sqrt{P_n} \notin Q[\sqrt{P_1}, \sqrt{P_2}, \dots, \sqrt{P_{n-1}}]$ où P_i est le i ème nombre entier premier.

[Preuve] Nous allons démontrer la proposition F_n suivante dont le théorème est une conséquence immédiate par récurrence sur n entier positif :

(F_n) $\sqrt{\prod P_{ix}/P_{jy}} \notin Q[P_{k1}, \dots, P_{kn}]$ où les P_l sont des nombres premiers distincts (plus généralement les P_l peuvent être pris premiers entre eux deux à deux) tels que $P_{ix}, P_{jy} \neq P_{kz}$ pour tout $ix, jy, et kz$

cas $n=1$, (F_1) s'écrit $\sqrt{P_{ix}/P_{jy}} \notin Q[P_{k1}]$ avec P_{k1} qui ne divise pas $\prod P_{ix}/P_{jy}$.

Notons $m = \sqrt{\prod P_{ix}/P_{jy}}$, m est rationnel,

(F_1) $\Leftrightarrow \sqrt{m} \notin K[P_{k1}]$

(F_1) \Leftrightarrow il n'y a pas a, b rationnels tels que $\sqrt{m} = a + b\sqrt{P_{k1}}$

(F_1) \Leftrightarrow il n'y a pas a, b rationnels tels que $m = (a + b\sqrt{P_{k1}})^2$ en mettant au carré

(F1) \Leftrightarrow il n'y a pas a, b rationnels tels que $m = a^2 + b^2 \cdot Pk_1 + 2a \cdot b \cdot \sqrt{Pk_1}$ (par simple développement)

(F1) \Leftrightarrow il n'y a pas a, b rationnels tels que $m = a^2 + b^2 \cdot Pk_1$ et $2a \cdot b = 0$ car 1 et $\sqrt{Pk_1}$ sont linéairement dépendant

(F1) \Leftrightarrow il n'y a pas a, b rationnels tels que $m = a^2 + b^2 \cdot Pk_1$ et ($a=0$ ou $b=0$) en utilisant la nullité du produit.

(F1) \Leftrightarrow il n'y a pas a, b rationnels tels que $m = a^2$ ou $m = b^2 \cdot Pk_1$ (par distributivité du ou et remplacement des valeurs)

(F1) \Leftrightarrow il n'y a pas a, b rationnels tels que $m = a^2$ ou $m/Pk_1 = b^2$

(F1) \Leftrightarrow m n'a pas de racines carrées dans Q et m/Pk_1 n'a pas de racine carrée dans Q comme par définition m n'a pas de racines carrées dans Q et par hypothèse de divisibilité m/Pk_1 n'a pas de racine carrée dans Q la propriété (F1) est vérifiée.

soit $n > 1$, nous supposons la propriété (Fk) vérifiée pour tout entier k strictement inférieur à n , montrons que la propriété est vérifiée pour n . (Fn) s'écrit $\sqrt{\prod P_{ix}/P_{jy}} \notin Q[Pk_1, \dots, Pk_n]$ où les P_i sont des nombres premiers distincts tels que $P_{ix}, P_{jy} \neq P_{kz}$ pour tout $ix, jy, et kz$

Notons $m = \prod p_{ix}/P_{jy}$, m est rationnel,

(Fn) $\Leftrightarrow \sqrt{m} \notin Q[Pk_1, \dots, Pk_n]$

(Fn) \Leftrightarrow il n'y a pas a, b dans $Q[Pk_1, \dots, Pk_{n-1}]$ tels que $\sqrt{m} = a + b\sqrt{Pk_n}$

(Fn) \Leftrightarrow il n'y a pas a, b dans $Q[Pk_1, \dots, Pk_{n-1}]$ tels que $m = (a + b\sqrt{Pk_n})^2$

(Fn) \Leftrightarrow il n'y a pas a, b dans $Q[Pk_1, \dots, Pk_{n-1}]$ tels que $m = a^2 + b^2 \cdot Pk_n + 2a \cdot b \cdot \sqrt{Pk_n}$

(Fn) \Leftrightarrow il n'y a pas a, b dans $Q[Pk_1, \dots, Pk_{n-1}]$ tels que $m = a^2 + b^2 \cdot Pk_n$ et $2a \cdot b = 0$

(Fn) \Leftrightarrow il n'y a pas a, b dans $Q[Pk_1, \dots, Pk_{n-1}]$ tels que $m = a^2 + b^2 \cdot Pk_n$ et (a ou b nuls)

(Fn) \Leftrightarrow il n'y a pas a, b dans $Q[Pk_1, \dots, Pk_{n-1}]$ tels que $m = a^2$ ou $m = b^2 \cdot Pk_n$

(Fn) \Leftrightarrow il n'y a pas a, b dans $Q[Pk_1, \dots, Pk_{n-1}]$ tels que $m = a^2$ ou $m/Pk_n = b^2$

(Fn) \Leftrightarrow m et m/Pk_n n'ont pas de racines carrées dans $Q[Pk_1, \dots, Pk_{n-1}]$

comme par hypothèse de récurrence m et m/Pk_n n'ont pas de racines carrées dans $Q[Pk_1, \dots, Pk_{n-1}]$ la propriété (Fn) est vérifiée. (l'hypothèse de récurrence s'applique à m/Pk_n , car $m/Pk_n = \prod P_{ix}/P_{jy}/Pk_n$ et $P_{ix}, P_{jy}, Pk_n \neq P_{kz}$ pour tout $ix, jy, et kz = 1 \dots n-1$)

Par récurrence la propriété est donc vérifiée. Le théorème en découle immédiatement.[]

Conséquence du théorème : $Q[\sqrt{P_1}, \sqrt{P_2}, \dots, \sqrt{P_n}]$ est de dimension 2^n sur Q

- Annexe 5-

liste des 72 pb. liste citée dans le texte II.4-e

A B C	" " C/2 H/2	" " " Hc/2
" " H	" " " Ha/2	" " Ha/2 Hb/2
" " Ha	" " " Hc/2	" " " Hc/2
" " Hc	" " H/2 Ha/2	" H Ha

" " Hb	" " " Hc/2	" " A/2 B/2
" " B/2 C/2	" " Ha/2 Hc/2	" " " C/2
" " " Ha/2	" B/2 C/2 H/2 Ha/2	" " " H/2
" " " Hb/2	" " " " Hb/2	" " " Hc/2
" " " Hc/2	" " " Ha/2 Hb/2	" " C/2 H/2
" " Ha/2 Hb/2	" " " Hb/2 Hc/2	" " " Hc/2
" " Hb/2 Hc/2	" " H/2 Ha/2 Hb/2	" " H/2 Hc/2
" Ha Hb	" " " " Hc/2	" A/2 B/2 C/2 H/2
" " B/2 C/2	" " " Hb/2 Hc/2	" " " " Hb/2
" " " H/2	" " Ha/2 Hb/2 Hc/2	" " " H/2 Hb/2
" " " Hb/2	" H/2 Ha/2 Hb/2 Hc/2	" " " " Hc/2
" " " Hc/2	H Ha Hb	" " " Hb/2 Hc/2
" " H/2 Hb/2	" " A/2 B/2	" " H/2 Hb/2 Hc/2
" " Hb/2 Hc/2	" " " Hb/2	" B/2 C/2 H/2 Hb/2
" Hb Hc	" " B/2 C/2	" " " Hb/2 Hc/2
" " B/2 C/2	" " " Hb/2	" " H/2 Hb/2 Hc/2
" " " H/2	" " " Hc/2	A/2 B/2 C/2 H/2 Ha/2
" " " Ha/2	" " Hb/2 Hc/2	Hb/2
" " " Hc/2	" A/2 B/2 C/2 Ha/2	" " " Ha/2 Hb/2
" " C/2 H/2	" " " Ha/2 Hb/2	Hc/2
" " " Ha/2	" " " " Hc/2	" " H/2 Ha/2 Hb/2
" " " Hc/2	" " Ha/2 Hb/2 Hc/2	Hc/2
" " H/2 Ha/2	Ha Hb Hc	

- Annexe 6-

preuves des théorèmes

théorème de l'orthocentre : Soit un triangle ABC et H le point de concours des hauteurs issus de C et B, alors H est sur la hauteur issue de A

[[preuve]] modélisation : soient A (0,0), B(U1,0), C(U2,U3), le point H(X1,X2) est construit selon :

$$(CH \perp AB) : X1 - U2 = 0 \text{ (F1)}$$

$$(BH \perp AC) : X1.U2 - U1.U2 + X2.U3 = 0 \text{ (F2)}$$

à démontrer :

$$(AH \perp BC) : X1.U1 - X1.U2 + X2.U3 = 0 \text{ (T)}$$

Le système F1, F2 est triangulaire en X1, X2. L'élimination des variables se fait selon X2 avec F2, puis selon X1 avec F1.

Selon l'estimation $\mathbf{K}(1)$, nous avons :

T : (X^1) , coefficient maximal : 1

T/F2 : (X^2) , coefficient maximal : 3

T/F2/F1 : (X^4) , coefficient maximal : 6

Selon l'estimation $\mathbf{K}(1(1(1\dots)))$, pour les variables prises dans l'ordre suivant X2, X1, U3, U2, U1 nous avons :

T : $(X2, X1, U3, U2, U1)$, coefficient maximal : 1

T/F2 : $(0, X1^2, U3^2, U2^2, U1^2)$, coefficient maximal : 3

T/F2/F1 : $(0, 0, U3^2, U2^4, U1^2)$, coefficient maximal : 6

Selon l'estimation $\mathbf{K}(1(1(1(\dots))),1(1(1(\dots))),1(1(1(\dots))),\dots)$, pour les variables prises dans l'ordre suivant X2, X1, U3, U2, U1 nous avons :

T : $(X2(0,U3,0,0),X1(0,U2,U1),0,0,0)$, coefficient maximal : 1

T/F2 : $(0,X1(U3,U2,U1),U3(U2,U1),U2(U1),0)$, coefficient maximal : 2

T/F2/F1 : $(0,0,U3(U2^2,U1),U2^2(U1),0)$, coefficient maximal : 4

[[]]

théorème du centroïde : soit un triangle ABC et G le point de concours des deux médianes issues de A et B, alors G est sur la médiane issue de C.

[[preuve]] modélisation : soient A (0,0), B(0,U1), C(U2,U3), le point G(X1,X2) est construit selon :

H sur la médiane issue de B : $X2.U2-X1.U3+2X1.U1-U1.U2 = 0$ (F1)

H sur la médiane issue de A : $X2.U2 -X1.U1+X1.U3 = 0$ (F2)

par triangularisation locale (élimination de X2 dans F1 à partir de F2) :

F1 est remplacé par : $-3X1+U2 = 0$ (F1') les conditions de non dégénérescence imposent $U1 \neq 0$

à démontrer :

H sur la médiane issue de C : $2X2.U2-2X1.U3-X1.U1-U2.U1 = 0$ (T)

Le système F1', F2 est triangulaire en X1, X2. L'élimination des variables se fait selon X2 avec F2, puis selon X1 avec F1'.

Selon l'estimation $\mathbf{K}(1)$, nous avons :

T : (X^1) , coefficient maximal : 2

T/F2 : (X^2) , coefficient maximal : 6

T/F2/F1 : (X^4) , coefficient maximal : 108

Selon l'estimation $\mathbf{K}(1(1(1\dots)))$, pour les variables prises dans l'ordre suivant X2, X1, U3, U2, U1 nous avons :

T : $(X2, X1, U3, U2, U1)$, coefficient maximal : 2

T/F2 : $(0, X1^2, U3^2, U2^2, U1^2)$, coefficient maximal : 6

T/F2/F1 : $(0, 0, U3^2, U2^4, U1^2)$, coefficient maximal : 60

Le point à évaluer (dans l'ordre U1, U3, U2 qui minimise les coefficients): $U1=61$, $U3=(U1^3-1)/(U1-1)+1$, $U2=(U3^3-1)/(U2-1)+1$

Selon l'estimation $\mathbf{K}(1(1(1(\dots))),1(1(1(\dots))),1(1(1(\dots))),\dots)$, pour les variables prises dans l'ordre suivant X2, X1, U3, U2, U1 nous avons :

T : $(X2(0,0,U2,0),X1(U3,0,U1),0,U2(U1),0)$, coefficient maximal : 2

T/F2 : (0,X1(U3,U2,U1),0,U2²(U1),0), coefficient maximal : 4
T/F2/F1 : (0,0,U3(U2²,U1),U2²(U1),0), coefficient maximal : 16

[[]]

théorème de Pappus : soient trois points A1, A2, A3 alignés, trois autres points B1, B2, B3 alignés, alors les points C1=(A2B3)n(A3B2), C2=(A1B3)n(A3B1), C3=(A1B2)n(A2B1) sont alignés.

[[preuve]] modélisation : soient Ai(0,Ui), Bi(Vi,0), les points Ci(Xi,Yi) sont construits selon :

C3 sur (A1B2) : V2.X3+U1.Y3-U1.V2=0 (F1)

C3 sur (A2B1) : V1.X3+U2.Y3-U2.V1 = 0 (F2)

par triangularisation locale (élimination de X3 dans F2 à partir de F1) :

F2 est remplacé par : -V1.U1.Y3+V1.U1.V2+V2.U2.Y3-V2.U2.V1 = 0 (F2') les conditions de non dégénérescence imposent V2≠0

C2 sur (A1B3) : V3.X2+U1.Y2-U1.V3 = 0 (F3)

C2 sur (A3B1) : V1.X2+U3.Y2-U3.V1 = 0 (F4)

par triangularisation locale (élimination de X2 dans F4 à partir de F3) :

F4 est remplacé par : -V1.U1.Y2+V1.U1.V3+V3.U3.Y2-V3.U3.V1 = 0 (F4') les conditions de non dégénérescence imposent V3≠0

C1 sur (A2B3) : V3.X1+U2.Y1-U2.V3 = 0 (F5)

C1 sur (A3B2) : V2.X1+U3.Y1-U3.V2 = 0 (F6)

par triangularisation locale (élimination de X1 dans F6 à partir de F5) :

F6 est remplacé par : -V2.U2.Y1+V2.U2.V3+V3.U3.Y1-V3.U3.V2 = 0 (F6')

à démontrer :

C1, C2, C3 alignés : X1.Y2+Y1.X3+X2.Y3-X3.Y2-Y3.X1-X2.Y1 = 0 (T)

Le système F1, F2', F3, F4', F5, F6' est triangulaire en X3, Y3, X2, Y2, X1, Y1. L'élimination des variables se fait selon X3 avec F1, puis selon Y3 avec F2', puis selon X2 avec F3, puis selon Y2 avec F4', puis enfin selon X1 avec F5, et selon Y1 avec F6'.

Selon l'estimation **K**(1), nous avons :

T : (X¹), coefficient maximal : 1

T/F1 : (X²), coefficient maximal : 3

T/F1/F2' : (X⁴), coefficient maximal : 48

T/F1/F2'/F3 : (X⁸), coefficient maximal : 3⁵2⁴.

T/F1/F2'/F3/F4' : (X¹⁶), coefficient maximal : 3⁵2¹².

T/F1/F2'/F3/F4'/F5 : (X³²), coefficient maximal : 3²¹2¹².

T/F1/F2'/F3/F4'/F5/F6' : (X⁶⁴), coefficient maximal : 3²¹2⁴⁴.

Selon l'estimation **K**(1(1(1(...))))), pour les variables prises dans l'ordre suivant X3, Y3, X2, Y2, X1, Y1, U3, V3, U2, V2, U1, V1 nous avons :

T : (X3,Y3,X2,Y2,X1,Y1,0,0,0,0,0,0), coefficient maximal : 1

T/F1: (0,Y3²,X2,Y2,X1,Y1,0,0,0,V2,U1,0), coefficient maximal : 1

T/F1/F2': (0,0,X2,Y2,X1,Y1,0,0,U2²,V2³,U1³,V1²), coefficient maximal : 4

T/.../F3 : (0,0,0,Y2²,X1,Y1,0,V3,U2²,V2³,U1⁴,V1²), coefficient maximal : 8

T/.../F4':(0,0,0,0,X1,Y1,U3²,V3³,U2²,V2³,U1⁶,V1⁴), coefficient maximal : 64

T/.../F5:(0,0,0,0,0,Y1²,U3²,V3⁴,U2³,V2³,U1⁶,V1⁴), coefficient maximal : 128

T/.../F6':(0,0,0,0,0,0,U3⁴,V3⁶,U2⁵,V2⁵,U1⁶,V1⁴), coefficient maximal : 1024

Selon l'estimation $\mathbf{K}(1(1(1(\dots))),1(1(1(\dots))),1(1(1(\dots))),\dots)$ pour les variables prises dans l'ordre suivant X3, Y3, X2, Y2, X1, Y1, U3, V3, U2, V2, U1, V1 nous avons (en abrégé, les 0 sont supprimés):

T : (X3(Y2,Y1), Y3(X2,X1), X2(Y1), Y2(X1)), coefficient maximal : 1

T/F1 : (Y3(Y2,X2,Y1,X1,V2,U1), X2(Y1,V2), Y2(X1,Y1,V2,U1), Y1(V2,U1)), coefficient maximal : 1

T/F1/F2' : (X2(Y1,U2,V2²,U1,V1), Y2(X1,Y1,U2,V2²,U1²,V1), X1(U2,V2,U1,V1),Y1(U2,V2²,U1²,V1), U2(V2²,U1,V1), V2²(U1²,V1)), coefficient maximal : 2

T/.../F3 : (Y2(X1,Y1,U2,V2²,U1²,V1), X1(U2,V2,U1,V1), Y1(V3,U2,V2²,U1²,V1), U2(V2²,U1,V1), V2²(U1²,V1)), coefficient maximal : 4

T/.../F4' : (X1(U3,V3,U2,V2,U1²,V1²), Y1(U3,V3²,U2,V2²,U1³,V1²), U3(V3,U2,V2²,U1³,V1²), V3(U2,V2²,U1³,V1²), U2(V2²,U1²,V1²), V2²(U1³,V1²)), coefficient maximal : 8

T/.../F5 : (Y1(U3,V3³,U2²,V2²,U1³,V1²), U3(V3²,U2²,V2²,U1³,V1²), V3²(U2²,V2²,U1³,V1²), U2(V2²,U1²,V1²), V2²(U1³,V1²)), coefficient maximal : 16

T/.../F6' : (U3²(V3⁴,U2³,V2³,U1³,V1²), V3⁴(U2³,V2³,U1³,V1²), U2²(V2³,U1²,V1²), V2²(U1³,V1²)), coefficient maximal : 64

[[]]

Résumé

Les contributions théoriques de ce travail comportent : 1) la définition d'une représentation exacte et normale des nombres constructibles (expressions arithmétiques avec racines carrées imbriquées), 2) un algorithme pour tester l'égalité, la différence et l'inégalité entre ces nombres et 3) une méthode pour la vérification de contraintes redondantes vis à vis d'un système de contraintes quadratiques. L'ensemble de ce travail est intégré dans un environnement CLP(Géométrie) reprenant l'approche de la programmation logique avec contraintes.

La principale contribution expérimentale porte sur un ensemble de 512 théorèmes de géométrie proposés par Chou. Elle consiste en la réduction du nombre d'extensions quadratiques nécessaires pour représenter ces situations géométriques grâce à un choix soigneux des constructions employées. Cette réduction s'accompagne du résultat suivant : la grande majorité des 512 théorèmes proposés par Chou peuvent être prise en compte de manière exacte avec un surcout limité, en pratique les nombres rationnels sont suffisants.

Le lien étroit de ce travail avec les préoccupations liées à la réalisation de tuteurs homme-machine font de cette approche un outils intéressant pour l'enseignement de la géométrie. Trois applications sont données dans ce domaine. La principale concerne une approche déclarative pour la définition et la manipulation de figures géométriques.

Summary

The theoretical contributions of this work include : 1) exact and normal representation of constructible numbers (arithmetic expressions with multiple nested square roots of rational numbers), 2) algorithm for equality, disequality and inequality for such real numbers and 3) results for checking if a given constraint is redundant with respect to a system of quadratic constraints. This work is embeded in an environment CLP(Géométrie) using the Constraint Logic Programming approach.

The main experimental contribution concern a set of 512 theorems in geometry proposed by Chou. It is the reduction in the number of quadratic extensions needed for the representation of geometrical situation by a careful choice of constructions. The result of this reduction is that the vast majority of the 512 theorems proposed by Chou can be handled with exact representation without prohibitive cost, practically only rational numbers are necessary in most cases.

The close linkage of this work with man-machine tutoring make the proposed approach an interesting tool for teaching geometry. Three applications are demonstrated in this area. The main one concern a declarative approach for the definition and the manipulation of geometrical figures.

mots clés: Contraintes quadratiques, non-linéaires; Nombres algébriques; Environnement Informatisé d'Apprentissage Humain; Géométrie Euclidienne, règle-compas; Prolog III.