N° D'ORDRE : 8385

UNIVERSITÉ PARIS XI
UFR SCIENTIFIQUE D'ORSAY

THÈSE

présentée pour obtenir

Le GRADE de DOCTEUR EN SCIENCES
DE L'UNIVERSITÉ PARIS XI ORSAY

SPÉCIALITÉ : Informatique

par

Julien NARBOUX

Formalisation et automatisation du raisonnement géométrique en Coq.

Soutenue le 26 septembre 2006
devant la commission d'examen composée de :

Jean-François	Dufourd	Président
Marc	Bezem	Rapporteurs
Loïc	Pottier	
Jacques	Fleuriot	Examinateurs
Claude	Marché	
Hugo	Herbelin	Directeur

Merci à Hugo Herbelin, mon directeur de thèse, il m'a beaucoup apporté. Je tiens à le remercier en particulier pour sa disponibilité ainsi que pour la rigueur avec laquelle il a relu mes articles et ma thèse.

Merci à Marc Bezem et à Loïc Pottier d'avoir accepté d'être les rapporteurs de cette thèse, et pour les remarques et suggestions qu'ils m'ont faites.

Merci à Marc Bezem, Jean-François Dufourd, Jacques Fleuriot, Claude Marché et Loïc Pottier d'avoir accepté de faire partie du jury, j'en suis très honoré.

Merci à Frédérique Guilhot pour les discussions que nous avons eues et pour ses remarques judicieuses à propos de mes articles et de cette thèse.

Merci à Shang-Ching Chou pour les échanges que nous avons eus par courrier électronique et pour avoir produit à ma demande une version électronique de sa thèse.

Merci à Laurent Théry pour avoir adapté mon implantation de la méthode des aires à la nouvelle tactique `field`.

Merci aux membres des équipes LogiCal et Proval qui m'ont permis de travailler dans un environnement agréable et stimulant.

Merci à ma famille pour avoir relu mon manuscrit et pour leur soutien.

Merci à Julie pour avoir relu mon manuscrit et surtout pour m'avoir supporté et encouragé au quotidien.

Table des matières

Introduction **1**

I Formalisation **7**

1 Axiomatiques pour la géométrie **9**
1.1 Introduction 9
1.2 Rappels sur les corps 10
1.3 Axiomatiques 11
1.3.1 Hilbert 11
1.3.2 Tarski 15
1.3.3 Heyting 17
1.3.4 von Plato 19
1.3.5 Wu 21
1.3.6 Chou, Gao et Zhang 24
1.4 Résumé et conclusion 27

2 Formalisation de la géométrie de Tarski **29**
2.1 Introduction 29
2.2 Travaux connexes et motivations 30
2.3 Retour sur l'axiomatique de Tarski 31
2.3.1 Description des axiomes 32
2.3.2 Historique des axiomatiques de Tarski 40
2.4 Aperçu de la formalisation 42
2.4.1 Deux lemmes cruciaux 43
2.4.2 Un exemple de preuve 43
2.5 A propos des cas dégénérés 50
2.6 Logique classique vs. logique intuitionniste. 51
2.7 Conclusion et perspectives 52

II Automatisation **53**

3 Démonstration automatique en géométrie **55**
3.1 Introduction . . . 55
3.2 Les méthodes algébriques . . . 56
3.2.1 La méthode des bases de Gröbner . . . 56
3.2.2 La méthode de Wu . . . 56
3.2.3 La décomposition cylindrique . . . 56
3.3 Les méthodes sans coordonnées . . . 57
3.3.1 La méthode des angles . . . 57
3.3.2 La méthode des vecteurs . . . 57
3.3.3 La méthodes des aires de Chou, Gao et Zhang . . . 57
3.4 Conclusion . . . 64

4 Formalisation de la méthode des aires en Coq **65**
4.1 Introduction . . . 65
4.2 Choix du langage . . . 66
4.3 Choix de l'axiomatique . . . 68
4.4 Propositions nécessaires à la tactique . . . 71
4.5 La tactique proprement dite . . . 72
4.5.1 Tactique d'initialisation . . . 72
4.5.2 Tactique de simplification. . . . 73
4.5.3 Tactiques d'unification. . . . 74
4.5.4 Tactique d'élimination. . . . 75
4.5.5 Tactique d'élimination des points libres. . . . 76
4.5.6 Tactique conclusion. . . . 76
4.6 Un exemple détaillé . . . 77
4.7 Exemples . . . 79
4.7.1 Ceva . . . 79
4.7.2 Menelaus . . . 80
4.7.3 Pascal . . . 81
4.7.4 Pappus . . . 82
4.7.5 Desargues . . . 83
4.7.6 Centre de gravité . . . 84
4.7.7 Droite de Gauss . . . 85
4.8 Conclusion . . . 86

III Visualisation **87**

5 *GeoProof*, géométrie dynamique et preuve formelle **89**
5.1 Introduction . . . 89
5.2 Présentation de *GeoProof* . . . 93
5.3 Démonstration automatique . . . 96

5.3.1 Méthode de démonstration automatique intégrée . . . 96
5.3.2 Avec Coq . . . 102
5.4 Preuve interactive . . . 103
5.5 Perspectives . . . 104
5.6 Conclusion . . . 107

6 Preuves diagrammatiques pour la réécriture 109
6.1 Introduction . . . 109
6.2 Représentation diagrammatique en réécriture abstraite . . . 111
6.2.1 Extension aux disjonctions. . . . 115
6.2.2 Langage des formules représentées . . . 117
6.2.3 A propos de la négation . . . 117
6.2.4 Définitions et propriétés usuelles . . . 117
6.3 Preuves diagrammatiques . . . 120
6.3.1 Règles d'inférence . . . 123
6.4 Correction et complétude du système . . . 127
6.4.1 Logique intuitionniste vs classique . . . 127
6.4.2 Calcul considéré . . . 128
6.4.3 Correction . . . 128
6.4.4 Complétude . . . 130
6.5 Extension aux preuves par induction . . . 134
6.5.1 L'induction classique . . . 134
6.5.2 L'induction bien fondée . . . 135
6.6 Implantation en Coq . . . 140
6.6.1 Règles d'inférence . . . 140
6.6.2 Règles implicites . . . 141
6.7 Quelques preuves diagrammatiques. . . . 143
6.7.1 Propriétés de confluence . . . 143
6.8 Conclusion et perspectives . . . 148

Perspectives 149

A Tous les triangles sont-ils isocèles ? 153

B La géométrie de Tarski 155
B.1 Axiomes . . . 155
B.2 Propriétés de `Cong` . . . 157
B.3 Propriétés de `Bet` . . . 158
B.4 Propriétés de `Cong` et `Bet` . . . 159
B.5 Transitivité de `Bet` . . . 161
B.6 Prédicat de non appartenance à un segment . . . 162
B.7 Propriétés du milieu . . . 164
B.8 Orthogonalité et existence du milieu . . . 165

C Manuel de référence de *GeoProof* **169**
C.1 Installation . . . 174
C.1.1 Windows . . . 174
C.1.2 Linux . . . 174
C.1.3 MacOSX . . . 174
C.2 Outils de construction . . . 174
C.3 Outils d'exploration . . . 177
C.3.1 Description en langue (pseudo-)naturelle . . . 178
C.3.2 Expressions . . . 178
C.3.3 Punaise . . . 180
C.3.4 Trace . . . 180
C.4 Attributs . . . 181
C.5 Outils de sélection . . . 182
C.6 Préférences . . . 183
C.7 Importation/Exportation . . . 184
C.8 Démonstration automatique . . . 184
C.9 Démonstration interactive . . . 186

D Réécriture abstraite **187**

Bibliographie **197**

Notations

$\Rightarrow \Longleftrightarrow \wedge \vee \exists \forall$	les connecteurs logiques
$\equiv$	l'égalité de distances (Tarski)
$\cong$	l'égalité de distances (Hilbert)
$\angle \cong \angle$	l'égalité des angles (Hilbert)
$\beta_T\ A\,B\,C$	B se situe entre A et C
$\mathcal{S}_{ABC}$	l'aire orientée du triangle ABC
$\mathcal{P}_{ABC}$	la différence de Pythagore de A, B et C
$\overline{AB}$	la distance orientée entre A et B
$\angle ABC$	l'angle non orienté ABC
AB	la distance entre A et B
(AB)	la droite passant par A et B
$[AB]$	le segment ayant pour extrémités A et B
$[AB)$	la demie-droite d'extrémité A passant par B
$\overrightarrow{AB}$	le vecteur ayant pour extrémités A et B
$(AB) \perp (CD)$	la droite (AB) est perpendiculaire à la droite (CD)
$(AB) \parallel (CD)$	la droite (AB) est parallèle à (CD)
$(AB) /\!/ (CD)$	la droite (AB) est strictement parallèle à la droite (CD)
$\xrightarrow{*}$	la clôture réflexive et transitive de $\longrightarrow$
$\xrightarrow{+}$	la clôture transitive de $\longrightarrow$
$\xrightarrow{=?}$	la clôture réflexive de $\longrightarrow$
$\xrightarrow{=}$	l'égalité
$\leftrightarrow$	la clôture symétrique de $\longrightarrow$
$\underline{x}$	x est libre (diagrammes)
$\vdash_{LK}$	calcul des séquents classique
$\vdash_{LJ}$	calcul des séquents intuitionniste
$\vdash_{=}$	calcul des séquents avec égalité
$\vdash_{\mid=}$	calcul des séquents avec égalité aux feuilles
$\vdash_{\mathcal{D}}$	calcul diagrammatique

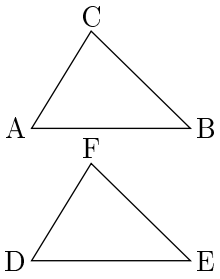
Introduction

Cette thèse s'inscrit dans le cadre de l'effort de recherche actuel à propos de la formalisation des mathématiques.

Nous nous intéressons à la mécanisation des preuves en géométrie. De Euclide à Tarski en passant par Hilbert, la géométrie a joué un rôle central dans l'histoire de la preuve mathématique. De nos jours, la géométrie est aussi souvent utilisée afin d'enseigner ce qu'est une preuve.

Euclide est considéré comme l'un des précurseurs de la *méthode axiomatique*. Dans *les Éléments* (300 av. J.-C.), en partant de quelques propositions supposées vraies qu'il appelle *postulats*, Euclide déduit tous les résultats qui avaient été découverts pendant les deux ou trois siècles précédents, et cela uniquement au moyen de règles logiques. *Les Éléments* d'Euclide ont longtemps été considérés comme un modèle de raisonnement mathématique, en un sens, *les Éléments* constituent le premier système formel mathématique.

Mais, lorsque l'on étudie précisement les preuves d'Euclide, on peut se rendre compte qu'il ne se conforme pas aussi strictement qu'on le voudrait à la méthode axiomatique. Certaines étapes de certaines preuves, même si elles paraissent évidentes, ne peuvent pas être déduites du système d'axiomes qu'il définit. Le raisonnement repose parfois sur l'intuition. En particulier la position relative des points et des droites est souvent *implicitement* admise. Bien qu'il soit l'un des précurseurs de la méthode axiomatique, Euclide n'atteint pas les standards modernes de rigueur mathématique.



Par exemple, dans *les Élements*, Euclide présente ce qu'il croit être une démonstration de SAS[1] : si deux côtés et l'angle formé par ces deux côtés d'un triangle sont égaux à ceux d'un autre triangle, les deux triangles sont égaux.
Démonstration : *déplacer ABC pour que le point A coïncide avec le point D et la droite (AB) coïncide avec (DE). Le point B coïncidera avec E car $AB = DE$. Aussi, la droite (AC) coïncidera avec (DF) car $\angle BAC = \angle EDF$. Le point C coïncidera avec F car $AC = DF$. La droite (BC) coïncidera avec (EF), car les deux droites ne peuvent inclure l'espace. Finalement, $BC = EF$, car les extrémités des segments coïncident. D'où $\angle ACB = \angle DFE$ et $\angle ABC = \angle DEF$.*

La faille dans ce raisonnement réside dans l'utilisation du terme déplacer. Rien dans les postulats d'Euclide ne dit que l'on peut utiliser cette *méthode de superposition*.

Ces étapes de raisonnement qui reposent sur l'intuition, sont potentiellement dangereuses et peuvent mener à des erreurs, un exemple bien connu dû à W. W. Rouse Ball est la « preuve » que tous les triangles sont isocèles. Nous reproduisons sur la figure 1 cet exemple. Nous laissons au lecteur le plaisir de trouver la faille, mais la solution figure en annexe A en page 153.

Dans le but de combler les imprécisions qui figurent dans le livre d'Euclide, les fondements de la géométrie ont fait l'objet de nombreuses recherches à la fin du XIXe siècle. En 1899, Hilbert propose un nouvel ensemble d'axiomes pour la géométrie. Le bénéfice principal apporté par cette nouvelle axiomatique est qu'elle permet de se passer de l'intuition. Le but de Hilbert est de proposer un développement parfaitement rigoureux des mathématiques[2] :

> « Depuis cinq ans, j'étudie les fondements des mathématiques en élaborant une théorie nouvelle de la démonstration. Je voudrais réduire tout énoncé mathématique à la présentation concrète d'une formule obtenue rigoureusement et donner ainsi aux notions et déductions mathématiques une forme irréfutable montrant bien l'ensemble de la science. »
> David Hilbert, *Les fondements des mathématiques*. Conférence faite en 1927, in *Les fondements de la géométrie* 1899. Edition critique, P. Rossier. Dunod 1971, appendice IX, p. 261.

Dans *les fondements de la géométrie*, Hilbert ne tente pas de définir ce qu'est un point ou une droite. Ces notions sont implicitement définies par les axiomes à propos des relations qui les lient.

[1] Side-Angle-Side

[2] Nous verrons au chapitre 2 qu'il n'y parvient qu'en partie car certaines preuves ne sont en fait pas parfaitement rigoureuses

Soit ABC un triangle quelconque.
Soit $\mathcal{D}$ la médiatrice de $[BC]$ et $\mathcal{D}'$ la bissectrice intérieure de l'angle $\angle BAC$.
Si $\mathcal{D} \parallel \mathcal{D}'$ alors ABC est isocèle en A.
Sinon, $\mathcal{D}$ et $\mathcal{D}'$ se coupent en un point I. Il y a deux cas :
Si I est dans le triangle ABC. Soit H (resp. G) le pied de la perpendiculaire à la droite (AB) (resp. (AC)) passant par I. Les triangles AIG et AIH sont égaux car ils ont un côté commun et deux angles égaux, donc $AH = AG$ et $IH = IG$. De plus $IB = IC$ car I est sur la médiatrice de $[BC]$. De même les triangle IGC et IHB sont égaux (un angle droit et deux côtés égaux), donc $HB = GC$. Comme on a $AH = AG$ et $HB = GC$ par addition, on a $AB = AC$.
Sinon I est en dehors du triangle ABC. La preuve est similaire, excepté qu'à la fin on réalise une soustraction plutôt qu'une addition.

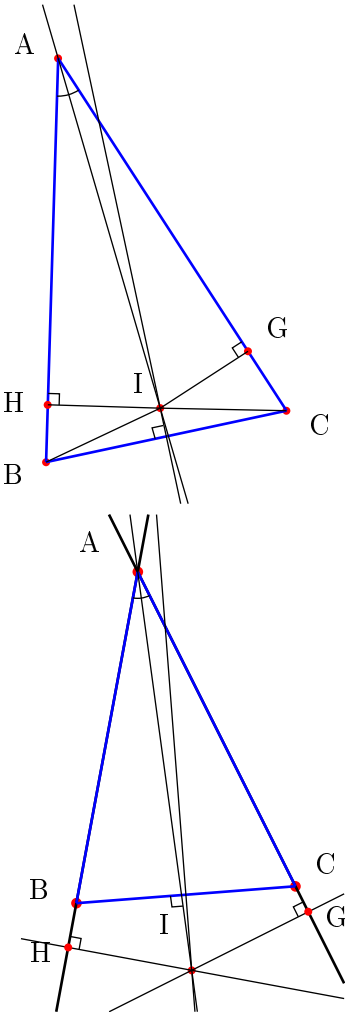


Fig. 1 – Tous les triangles sont isocèles (W. W. Rouse Ball, 1892).

> « Nous pensons trois systèmes différents de choses ; nous nommons les choses du premier système des points ; nous les désignons par des majuscules A, B, C,... ; nous nommons droites les choses du deuxième système et nous les désignons par des minuscules a, b, c,... ; nous appelons plans les choses du troisième système et nous les désignons par les caractères grecs α,β,γ... »
> David Hilbert, *Les fondements de la géométrie*, 1899.

Si pour vérifier une preuve, on peut se passer de l'intuition, il devient alors possible de vérifier mécaniquement des preuves. Le fait d'être une preuve étant alors par définition décidable, il est possible d'utiliser un ordinateur pour vérifier des preuves. C'est dans ce contexte que s'inscrit cette thèse. La *vérification* mécanisée des preuves en géométrie fait l'objet de la première partie.

De plus, dès lors que dans une démonstration on peut se passer de l'intuition, on peut aussi imaginer générer automatiquement des théorèmes. Poincaré l'a tout de suite remarqué (même si c'était dans le but de défendre le rôle de l'intuition dans les mathématiques) :

> « Ainsi c'est bien entendu, pour démontrer un théorème, il n'est pas nécessaire ni même utile de savoir ce qu'il veut dire. On pourrait remplacer le géomètre par le piano à raisonner imaginé par Stanley Jevons ; ou, si l'on aime mieux, on pourrait imaginer une machine où l'on introduirait les axiomes par un bout pendant qu'on recueillerait les théorèmes à l'autre bout, comme cette machine légendaire de Chicago où les porcs entrent vivants et d'où ils sortent transformés en jambons et en saucisses. Pas plus que ces machines, le mathématicien n'a besoin de comprendre ce qu'il fait. »
> Henri Poincaré, *Les mathématiques et la logique*. Revue de Métaphysique et de Morale, 1905.

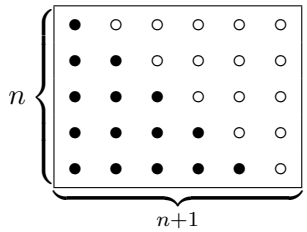
Cette question a aujourd'hui beaucoup évolué, et on sait maintenant que pour certains types de porcs on peut construire des machines et pour d'autres non.

Un exemple important d'une telle machine est la procédure de décision de Tarski pour sa géométrie, *i.e.* la théorie des corps réels clos [Tar51].

Depuis, la géométrie a été l'un des domaines où la déduction automatique a été la plus fructueuse. Des conjectures ont été prouvées pour la première fois par des méthodes automatiques. Ce thème fait l'objet de la deuxième partie de cette thèse.

Enfin, comme nous avons pu le voir à propos de l'exemple des triangles isocèles, l'utilisation d'un raisonnement qui repose sur une figure peut parfois induire en erreur, mais d'un autre côté, une figure peut souvent éclairer le lecteur. La définition de la preuve comme *potentiellement mécanisable*, aussi fructueuse soit elle, ne doit pas faire perdre de vue le critère de lisibilité

d'une preuve qui lui permet de devenir *convaincante*. Il existe des preuves qui peuvent être considérées comme *convaincantes* sans pour autant être formelles. Voici une « preuve » diagrammatique[3] que la somme des entiers de 1 à n vaut $\frac{n(n+1)}{2}$:



On voudrait donc pouvoir décider quand une figure peut être considérée comme une preuve correcte ou pas. Dans la troisième partie de cette thèse, nous nous intéressons à cette question à propos des preuves diagrammatiques que l'on peut réaliser dans le domaine de la réécriture abstraite.

Cette thèse est ainsi composée de trois parties. Dans la première partie nous faisons d'abord un tour d'horizon des principales axiomatiques de la géométrie, puis nous présentons notre formalisation de la géométrie de Tarski. Dans la seconde partie, après avoir présenté les principales procédures de décision en géométrie, nous décrivons notre implantation, dans l'assistant de preuve Coq, de la méthode des aires de Chou, Gao et Zhang. Dans la troisième partie, nous présentons d'abord la conception d'un outil de visualisation et d'interaction graphique dans le cadre de la preuve formelle en géométrie, puis nous proposons un système formel diagrammatique pour réaliser des preuves diagrammatiques dans le cadre de la réécriture abstraite.

[3] Pour d'autres exemples similaires et l'étude de leur formalisation, voir la thèse de Mateja Jamnik [Jam01].

Première partie

Formalisation

Chapitre 1

Axiomatiques pour la géométrie

1.1 Introduction

Dans le langage courant le terme géométrie est utilisé pour décrire l'ensemble des définitions et théorèmes qui sont exposés ou découlent des *Éléments* d'Euclide[1]. Dans cette thèse nous nous intéressons à la formalisation de la géométrie, il faut donc donner une définition plus précise du terme géométrie. Avant de nous lancer dans une formalisation il faut faire un choix, celui de l'axiomatique et de la géométrie car, nous allons le voir, il existe de nombreuses géométries et pour chacune d'entre elles plusieurs axiomatiques différentes.

Dans ce chapitre, nous réalisons un tour d'horizon des principales axiomatiques en géométrie. Afin de définir des géométries, il y a deux approches possibles : l'approche *algébrique* ou *axiomatique*.
La première consiste d'une manière générale à définir la géométrie à partir d'objets (les points) appartenant au produit cartésien C^n d'un corps C. Les points sont donc définis par leurs coordonnées.
Le seconde approche, utilisée par Hilbert, Euclide et Tarski consiste à définir des axiomes qui ont un sens « géométrique » et à se passer du concept de corps.
Mais comme certaines théories géométriques sont catégoriques (c'est à dire que tous leurs modèles sont isomorphes), on peut les caractériser par leur modèle (unique à isomorphisme près). La seconde approche rejoint alors la première puisque ces modèles sont définis de manière algèbrique. Nous avons choisi d'organiser notre présentation sous l'angle des diverses axiomatiques.

[1] Cette définition informelle ne rend pas compte de l'existence des géométries non-euclidiennes. Nous l'adoptons tout de même puisque les géométries non-eucliennes ne seront pas abordées dans cette thèse.

Nous donnons une présentation de six axiomatiques différentes. Mais avant de rentrer dans le détail de ces axiomatiques nous commençons par quelques définitions à propos des corps, celles-ci seront utilisées pour définir certains *modèles* de ces géométries.

1.2 Rappels sur les corps

Les notions suivantes de corps seront utilisées dans la suite. Nous commençons par la définition du concept de corps ordonné. Géométriquement l'ordre permet d'exprimer le fait pour un point d'être situé entre deux autres ou bien le fait d'être situé dans un demi-plan.

Définition 1 (corps ordonné)**.** *Un corps ordonné est un corps muni d'une relation d'ordre total $\leq$ vérifiant les propriétés suivantes :*

- $\forall ab\ a \leq b \Rightarrow a + c \leq b + c$
- $\forall ab\ 0 \leq a \wedge 0 \leq b \Rightarrow 0 \leq ab$

Attention, tous les corps ne peuvent pas être ordonnés, l'exemple le plus connu est le corps des nombres complexes $\mathbb{C}$.

Définition 2 (corps hilbertien ou pythagoricien)**.** *On dit qu'un corps C est « hilbertien » si la somme des carrés de deux de ses éléments admet toujours une racine carrée.*

$$\forall xy \in C, \exists z \in C, x^2 + y^2 = z^2$$

L'étymologie du qualificatif vient du fait que si x et y sont dans un corps pythagoricien alors l'hypoténuse du triangle rectangle de côtés x et y l'est aussi. Notons que cette définition équivaut à :

$$\forall x \in C, \exists z \in C, 1 + x^2 = z^2$$

Exemple. *L'extension de $\mathbb{Q}$ avec toutes les racines carrées de p pour p premier dans $\mathbb{Z}$ est un corps hilbertien appelé corps de Hilbert (noté $\mathbb{H}$).*

Définition 3 (corps euclidien)**.** *On dit qu'un corps ordonné C est « euclidien » si tout élément positif ou nul admet une racine carrée.*

$$\forall x \in C, \exists y \in C, x \geq 0 \Rightarrow x = y^2$$

Remarque 1. *Tout corps euclidien est hilbertien, mais l'inverse n'est pas vrai, $\mathbb{H}$ fournit un contre exemple car $\sqrt{2}$ n'a pas de racine carrée dans $\mathbb{H}$.*

Exemple. *Le plus petit corps euclidien contenant $\mathbb{Q}$ est le corps des nombres constructibles à la règle et au compas.*

Définition 4 (corps réel clos)**.** *On dit qu'un corps ordonné C est un corps réel clos lorsque tous les éléments positifs ou nuls de C possèdent une racine carrée et tout polynôme de degré impair à coefficients dans C admet au moins une racine dans C.*

Exemple. $\mathbb{R}$ *est un corps réel clos.*

Nous avons défini quelques corps qui serviront à décrire les modèles de certaines géométries, mais pour cela nous avons aussi besoin de la notion de plan cartésien :

Définition 5 (plan cartésien)**.** *La plan cartesien sur un corps C, est l'ensemble C^2 des paires ordonnées d'éléments de C appelés points. Les droites sont les ensembles de points définis par l'équation $ax + by + c = 0$ avec a,b et c dans C et $a \neq 0$ ou $b \neq 0$.*

1.3 Axiomatiques

Dans cette partie nous présentons la géométrie sous l'angle de l'axiomatique employée pour la définir. Nous nous intéressons d'abord aux deux axiomatiques les plus connues : celles de Hilbert et de Tarski. Ensuite nous décrivons deux axiomatiques constructives. Enfin nous présentons deux axiomatiques bien particulières qui servent de fondements à deux méthodes de démonstration automatique : les axiomatiques de Wu et de Chou, Gao et Zhang. Le deuxième sert de base au développement que nous décrivons dans le chapitre 4. Dans le chapitre suivant nous reviendrons sur l'axiomatique de Tarski plus en détail et présenterons la formalisation que nous en avons faite dans l'assistant de preuve Coq.

1.3.1 Hilbert

Cette partie présente l'ensemble d'axiomes proposé par Hilbert en 1899. Ces axiomes ont pour but de créer des fondements rigoureux pour la géométrie d'Euclide. Le texte *Grundlagen der Geometrie* (*Les fondements de la géométrie* [Hil71]) remplace les axiomes d'Euclide, par un ensemble de 20 axiomes (21 dans la version originale). Les principales innovations de cet ensemble d'axiomes résident dans l'utilisation de l'axiome de Pasch et dans le fait que la méthode de superposition d'Euclide que nous évoquions en introduction disparaît.

Dans sa version plane, l'axiomatique de Hilbert repose sur deux sortes d'objets : les points et les droites. Dans sa version tridimensionnelle il faut ajouter les plans. Il n'est pas précisé ce que sont les points, les droites, et les plans ni même qu'un ensemble de points peut former une droite. Ces notions sont définies uniquement par les axiomes qui définissent les relations entre ces notions.

Les relations sont les suivantes :

- L'appartenance d'un point à une droite.
- L'appartenance d'un point à un plan.
- Le fait pour un point d'être situé entre deux autres.
- La congruence des segments (notée $\cong$).
- La congruence des angles (notée $\cong$).

Dimension 2

Groupes I : Axiomes d'appartenance. Les axiomes d'appartenance traitent des points, des droites et des plans et de leurs intersections.

I1 Pour tout couple de deux points A et B, il existe une droite contenant A et B.

I2 Il n'existe pas plus d'une droite à laquelle appartiennent deux points A et B.

I3 Toute droite contient au moins deux points ; il existe au moins trois points non alignés.

Notons que l'unicité des objets est énoncée explicitement alors que ce n'est pas le cas dans les *Éléments* d'Euclide .

Groupe II : Axiomes d'ordre. Nous présentons maintenant les axiomes d'ordre qui définissent le fait pour un point d'être entre deux autres. Cette notion sert aussi à définir des prédicats qui décrivent si un point se trouve d'un côté ou de l'autre d'une droite, si un angle est plus grand qu'un autre ou si une distance est plus grande qu'une autre.

II1 Si un point B est entre A et C, B est aussi entre C et A et il existe une droite contenant les points A, B et C.

II2 Étant donnés deux points A et C, il existe un point B sur la droite (AC) tel que C est entre A et B.

II3 Étant donnés trois points sur une droite, un et un seul est entre les deux autres.

II4 (Pasch) Étant donnés trois points non alignés A,B et C, et une droite m qui ne contient ni A ni B ni C, si m contient un point du segment $[AB]$[2], alors m contient aussi un point du segment $[AC]$ ou un point du segment $[BC]$ (voir figure 1.1).

[2]Notons que l'expression « m contient un point du segment $[AB]$ » peut-être formalisée par « Il existe P appartenant à m tel que P est entre A et B ». Il n'est donc pas nécessaire ici de définir explicitement la notion de segment.

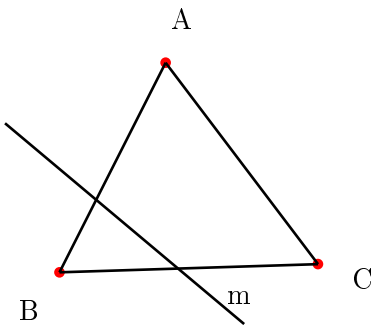


FIG. 1.1 – Axiome de Pasch

Groupe III : Axiomes de congruence.

III1 Si A et B sont deux points d'une droite a et A' est un point d'une droite a' (éventuellement égale à a) alors on peut trouver un point B' sur a' tel que le segment $[AB]$ soit congruent (ou égal) au segment $[A'B']$.

III2 Si deux segments sont congruents à un troisième, ils sont congruents entre eux. Tout segment est congruent à lui-même.

III3 Soient $[AB]$ et $[BC]$ deux segments sans point commun portés par la droite a, $[A'B']$ et $[B'C']$ deux segments de la droite a' eux aussi sans point commun, si $AB \cong A'B'$ et $BC \cong B'C'$, alors $AC \cong A'C'$.

III4 Étant donné un angle $\angle BAC$ et une demi-droite $[DF)$, il existe une unique demi-droite $[DE)$, sur un côté donné de la droite (DF) telle que $\angle BAC \cong \angle EDF$.

III5 Deux angles congruents à un même troisième sont congruents entre eux. Tout angle est congruent à lui-même.

III6 (SAS) Étant donné des triangles ABC et DEF. Si $AB \cong DE$, $AC \cong DF$ et $\angle BAC \cong \angle EDF$ alors $BC \cong EF$, $\angle ABC \cong \angle DEF$ et $\angle ACB \cong \angle DFE$.

Groupe IV : Axiome des parallèles. Le groupe IV ne contient qu'un seul axiome, le fameux axiome d'Euclide :

IV : Axiomes d'Euclide Soient une droite a et un point A extérieur à a ; il existe au plus une droite qui passe par A et qui ne coupe pas a.

Groupe V : Axiomes de continuité

V1 (Axiome de la mesure ou axiome d'Archimède) Étant donné un segment $[CD]$ et une demi-droite $[AB)$, il existe n points $A_1, \ldots, A_n$ sur (AB), tels que $A_jA_{j+1} \cong CD$, $1 \leq j < n$. De plus, B est entre A_1 et A_n.

V2 (Dedekind) Supposons que l'ensemble des points d'une droite est l'union de deux ensembles non vides X et Y de points, de telle manière que aucun point de X n'est entre deux points de Y et vice versa alors il existe un unique point P tel que pour tout $x \in X$ et $y \in Y$, P coïncide avec x ou y ou bien P est entre x et y.

Les axiomes d'Archimède et de Dedekind sont les homologues géométriques des axiomes qui portent le même nom à propos des réels.

Les deux principes suivants sont dérivables à partir de l'axiome de Dedekind :

Principe d'intersection cercle-cercle :
Si un cercle possède un point à l'intérieur et un point à l'extérieur d'un autre cercle alors ces deux cercles s'intersectent en deux points.

Principe d'intersection cercle-droite :
Si une extrémité d'un segment est à l'intérieur d'un cercle et l'autre extrémité est à l'extérieur du cercle, alors le segment intersecte le cercle.

Dimension 3

Pour obtenir l'axiomatique de Hilbert en dimension 3, il faut ajouter un troisième type d'objet : les plans. De plus il faut rajouter des axiomes concernant ces plans et reformuler deux axiomes.

Dans le groupe I, on ajoute les cinq axiomes suivants :

I4 Étant donnés trois points non alignés, il existe un plan contenant les trois points. Tout plan contient au moins un point.

I5 Étant donnés trois points non alignés, il existe un seul plan contenant les trois points.

I6 Si deux points d'une droite appartiennent à un même plan alors tous les points de la droite appartiennent à ce plan.

I7 Si deux plans ont un point commun alors ils en ont au moins un autre.

I8 Il existe au moins quatre points non coplanaires.

L'axiome de Pasch s'énonce de la manière suivante :

Étant donnés trois points non alignés A,B et C, et une droite m du plan ABC mais qui ne contient ni A ni B ni C, si m contient un point du segment $[AB]$, alors m contient aussi un point du segment $[AC]$ ou un point du segment $[BC]$

L'axiome d'Euclide s'énonce de la manière suivante :

Soient une droite a et un point A extérieur à a ; dans le plan déterminé par a et A, il existe au plus une droite qui passe par A et qui ne coupe pas a.

Commentaires

L'axiome d'Archimède est indépendant des autres axiomes de Hilbert, les géométries qui vérifient l'axiome d'Archimède sont ainsi appelées archimèdiennes et les autres non-archimèdiennes. Notons que la plupart des résultats prouvés par Hilbert ne dépendent pas de l'axiome d'Archimède. L'axiome de Dedekind est utile aux propriétés d'intersection des cercles.

Proposition 1. *Les* groupes *d'axiomes de l'axiomatique de Hilbert sont indépendants.*

Modèles

Nous résumons ici quelques résultats concernant les modèles de la géométrie de Hilbert, pour plus de détails voir [Har00].

Proposition 2 (géométrie affine)**.** *Pour tout corps C le plan cartésien sur C satisfait les axiomes d'appartenance (I1-I3) ainsi que l'axiome des parallèles (IV).*

Proposition 3 (géométrie affine ordonnée)**.** *Pour tout corps* ordonné *C le plan cartésien sur C satisfait les axiomes d'appartenance, des parallèles ainsi que les axiomes d'ordre (I-II-IV) et de congruence III2-III5.*

Proposition 4. *Soit C un corps ordonné, la plan cartésien sur C satisfait III1 ssi C est pythagoricien.*

Proposition 5. *Les propriétés suivantes sont équivalentes :*
- *C est euclidien*
- *Le plan cartésien sur C satisfait le principe d'intersection cercle-cercle.*
- *Le plan cartésien sur C satisfait le principe d'intersection cercle-droite.*

L'axiomatique de Hilbert est catégorique (tous ses modèles sont isomorphes).

Proposition 6. *Les modèles de l'axiomatique de Hilbert sont tous isomorphes à $\mathbb{R}^2$ le plan réel.*

1.3.2 Tarski

Alfred Tarski a travaillé sur l'axiomatisation et sur les méta-mathématiques de la géométrie euclidienne de 1926 à 1983.

TAB. 1.1 – L'axiomatique de Tarski [Tar59].

Identité	$\forall xy, \beta_T\ x\,y\,x \Rightarrow x = y$
Transitivité	$\forall xyzu, \beta_T\ x\,y\,u \wedge \beta_T\ y\,z\,u \Rightarrow \beta_T\ x\,y\,z$
Connectivité	$\forall xyzu, \beta_T\ x\,y\,z \wedge \beta_T\ x\,y\,u \wedge x \neq y \Rightarrow \beta_T\ x\,z\,u \vee \beta_T\ x\,u\,z$
Réflexivité	$\forall xy, xy \equiv yx$
Identité	$\forall xyz, xy \equiv zz \Rightarrow x = y$
Transitivité	$\forall xyzuvw, xy \equiv zu \wedge xy \equiv vw \Rightarrow zu \equiv vw$
Pasch	$\forall txyzu, \exists v, \beta_T\ x\,t\,u \wedge \beta_T\ y\,u\,z \Rightarrow \beta_T\ x\,v\,y \wedge \beta_T\ z\,t\,v$
Euclide	$\forall txyzu, \exists vw, \beta_T\ x\,u\,t \wedge \beta_T\ y\,u\,z \wedge x \neq u \Rightarrow$ $\beta_T\ x\,z\,v \wedge \beta_T\ x\,y\,w \wedge \beta_T\ v\,t\,w$
5 segments	$\forall xx'yy'zz'uu'x \equiv yx'y' \wedge yz \equiv y'z' \wedge$ $xu \equiv x'u' \wedge yu \equiv y'u' \wedge$ $\beta_T\ x\,y\,z \wedge \beta_T\ x'\,y'\,z' \wedge x \neq y \Rightarrow zu \equiv z'u'$
Construction	$\forall xyuv, \exists z, \beta_T\ x\,y\,z \wedge yz \equiv uv$
Dimension	$\exists xyz, \neg\beta_T\ x\,y\,z \wedge \neg\beta_T\ y\,z\,x \wedge \neg\beta_T\ z\,x\,y$
Dimension	$\forall xyzuv, xu \equiv xv \wedge yu \equiv yv \wedge zu \equiv zv \wedge u \neq v$ $\Rightarrow \beta_T\ x\,y\,z \vee \beta_T\ y\,z\,x \vee \beta_T\ z\,x\,y$
Continuité	$\exists z, \forall xy, \phi \wedge \psi \Rightarrow \beta_T\ z\,x\,y \Rightarrow \exists u, \forall xy, \phi \wedge \psi \Rightarrow \beta_T\ x\,u\,y$ où ϕ est une formule dans laquelle y,z et u n'apparaissent pas libres et ψ est une formule dans laquelle x,z et u n'apparaissent pas libres.

L'axiomatique de Tarski se base sur la logique du premier ordre, et deux prédicats. Le premier β_T exprime le fait qu'un point appartient à un segment, le second $\equiv$ exprime la congruence de segments. Dans cette formalisation de la géométrie seuls les points sont traités comme des individus et sont représentés par des variables. En particulier il n'y a aucun symbole pour représenter des figures géométriques (des ensembles de points). Cette géométrie permet tout de même d'exprimer la plupart des résultats habituels qui sont formulés en termes de droites, cercles, segments, triangles, etc. Ceci est dû au fait que ces notions peuvent être représentées indirectement par l'ensemble des points qui les définit.

Nous reproduisons sur le tableau 1.1 l'axiomatique de Tarski telle qu'elle a été publiée en 1959 [Tar59]. Nous détaillerons la signification de ces axiomes dans le chapitre suivant.

Propriétés méta-théoriques

La géométrie de Tarski possède les propriétés suivantes :
- elle est cohérente : on ne peut pas démontrer A et $\neg A$.
- elle est complète : pour toute formule F alors soit F est dérivable soit $\neg F$ est dérivable.
- elle est décidable : il existe un algorithme qui permet de dire si une

formule est dérivable ou non.
- elle n'est pas axiomatisable de manière finie comme une théorie de premier ordre sur le même langage.
- elle est catégorique, ses modèles sont tous isomorphes au carré R^2 d'un corps réel clos R.

Indépendance. On dit que des axiomes sont indépendants, si aucun d'entre eux ne peut être dérivé au moyen des autres. Pour prouver que des axiomes $A_1 \dots A_n$ sont indépendants pour chacun des axiomes A_i il faut exhiber un modèle qui vérifie $A_1 \dots A_{i-1}$, $A_{i+1} \dots A_n$ et $\neg A_i$.

Gupta [Gup65] a prouvé l'indépendance des axiomes de Tarski, excepté l'axiome de Pasch et la réflexivité de la congruence qui restent des problèmes ouverts.

Comparaison avec l'axiomatique de Hilbert. L'axiomatique de Tarski n'a pas que l'avantage d'avoir de bonnes propriétés méta-théoriques, elle est aussi d'une extrême simplicité. Comme nous le verrons elle peut être réduite à 11 axiomes. Cette simplicité est réelle, les deux prédicats sur lesquels elle repose ont une signification géométrique évidente, et les axiomes qu'elle utilise sont aussi très simples.

Notons qu'il est possible d'avoir une axiomatique basée uniquement sur un seul prédicat [Pie99] mais c'est au prix d'un complexification de l'axiomatique. En effet dans ce cas on obtient 24 axiomes dont certains sont presque aussi longs que l'axiomatique de Tarski dans son ensemble. On pourrait croire que l'axiomatique de Tarski est plus simple car elle a su utiliser des axiomes plus complexes qui contiennent plus de puissance déductive. Ce n'est pas le cas, les axiomes utilisés par Tarski ont tous été utilisés dans des travaux précédents sur les axiomatiques. Ce qui peut expliquer la simplicité de cette axiomatique est que les conditions de non dégénérescence sont réduites au maximum, les axiomes peuvent ainsi être utilisés dans les cas limites.

Du point de vue pratique de la formalisation, ce point est crucial car il simplifie grandement les développements. Nous reviendrons sur ce point dans le chapitre suivant.

Un autre avantage de l'axiomatique de Tarski par rapport à l'axiomatique de Hilbert réside dans le fait qu'il suffit de changer deux de ses axiomes pour obtenir des géométries dans d'autres dimensions. Dans l'axiomatique de Hilbert, nous avons vu que pour passer de la dimension deux à la dimension trois il a fallu ajouter un type d'objet (les plans) et des axiomes.

1.3.3 Heyting

Nous présentons dans cette partie les axiomatiques de Heyting pour la géométrie projective et la géométrie affine plane [Hey59]. Ces axiomatiques

ont la particularité d'être intuitionnistes, elles ne supposent pas que l'égalité de deux points est décidable.

L'axiomatique de Heyting, comme celle de Hilbert, est basée sur deux types d'objets que l'on peut interpréter par d'une part les points et d'autre part les droites. Elle admet aussi deux relations, la première notée $\sharp$ exprimant la différence (*apartness*) des points et la deuxième l'appartenance d'un point à une droite notée $\in$ (*incidence*). Les points sont dénotés par des lettres majuscules et les droites par des minuscules. Les variables libres sont implicitement quantifiées universellement.

Heyting définit deux axiomatiques. La première concerne la géométrie projective et la seconde la géométrie affine plane.

Géométrie projective

Axiomes de « différence » : le premier groupe définit le prédicat de différence de deux points. L'égalité est par définition la négation de la différence. Notons que comme nous sommes dans un contexte constructif $A \neq B$ n'implique pas que $A \sharp B$.

S1 $A \sharp B \Rightarrow B \sharp A$.

S2 $\neg A \sharp B \iff A = B$.

S3 $A \sharp B \Rightarrow \forall C, C \sharp A \vee C \sharp B$.

Axiomes géométriques :

P1 $A \sharp B \Rightarrow \exists l, A \in l \wedge B \in l$

P2 $A \sharp B \wedge A \in l \wedge A \in m \wedge B \in l \wedge B \in m \Rightarrow l = m$

Définition 6. *On dit que A est en dehors de l, noté $A \,\omega\, l$ ssi :*

$$\forall B, B \in l \Rightarrow B \sharp A$$

Définition 7. *On dit que l est distincte de m, noté $l \sharp_l m$ ssi :*

$$\exists A, A \in l \wedge A \,\omega\, m$$

P3 $l \sharp_l m \Rightarrow \exists A, A \in l \wedge A \in m$

P4 $A \sharp B \wedge A \in l \wedge B \in l \wedge C \,\omega\, l \wedge A \in m \wedge C \in m \Rightarrow B \,\omega\, m$

P5

- **i** $\exists A\, B, A \sharp B$
- **ii** $\exists A\, B\, C, A \in l \wedge B \in l \wedge C \in l \wedge A \sharp B \wedge A \sharp C \wedge B \sharp C$
- **iii** $\exists A, A \,\omega\, l$

Géométrie affine plane

Axiomes de non appartenance : ils sont les mêmes que précédemment : S1,S2 et S3.

Les définitions 6 et 7 sont identiques.

Axiomes géométriques :

A1 $l\sharp_l m \wedge A\,\omega\,l \Rightarrow \exists p, A \in p \wedge \forall X, (X \in l \wedge X \in p) \iff (X \in l \wedge X \in m)$

A2 $A\sharp B \wedge A \in l \wedge A \in m \wedge B \in l \wedge B \in m \Rightarrow l = m$

Définition 8. *On dit que l et m sont sécantes ssi :*

$$l\sharp_l m \wedge \exists A, A \in l \wedge A \in m$$

A3 l et m sont sécantes $\Rightarrow \forall p, \exists A, (A \in l \wedge A \in p) \vee (\exists B, B \in m \wedge B \in p)$

A4 $A\sharp B \wedge A \in l \wedge B \in l \wedge C\,\omega\,l \wedge A \in m \wedge C \in m \Rightarrow B\,\omega\,m$

A5 $P\,\omega\,l \wedge \neg(\exists X, X \in l \wedge X \in m) \wedge P \in m \wedge Q \in l \Rightarrow Q\,\omega\,m$

Définition 9. *On dit que l est parallèle à m noté $l /\!/ m$ ssi :*

$$\forall A, A \in l \Rightarrow A\,\omega\,m$$

A6 $\forall l, \exists m, l /\!/ m$

A7

i $\exists l$

ii $\exists ABCD, A\sharp B \wedge A\sharp C \wedge A\sharp D \wedge B\sharp C \wedge B\sharp D \wedge A \in l \wedge B \in l \wedge C \in l \wedge D \in l$

iii $A\sharp B \Rightarrow \exists l, A \in l \wedge B\,\omega\,l$

iv $A \in l \Rightarrow \exists m, A \in m \wedge l\sharp_l m$

1.3.4 von Plato

L'axiomatique de von Plato est une autre axiomatique constructive de la géométrie. Elle introduit les concepts de droites convergentes et de non appartenance d'un point à une droite. Ces notions correspondent aux concepts classiques[3] de parallélisme et d'appartenance d'un point à une droite. Nous donnons ici l'axiomatique de von Plato pour la géométrie affine plane ordonnée. Comme celles de Heyting et de Hilbert, l'axiomatique de von Plato est basée sur des points et des droites. Nous dénotons les points par des majuscules et les droites par des minuscules.

Cette axiomatique est basée sur cinq prédicats qui intuitivement ont les significations suivantes :

$A\sharp B$ A et B sont des points distincts.

[3] Ce mot est à prendre dans les deux sens du terme.

$l\sharp_l m$ l et m sont des droites distinctes.
$Undir(l,m)$ l et m ne sont pas orientées dans la même direction.
$LApt(A,l)$ le point A est situé strictement à gauche de l.
$LCon(l,m)$ l coupe m par la gauche.

Von Plato suppose aussi l'existence de quatres fonctions de construction :
ln(A,B) la droite passant par A et B.
pt(l,m) le point d'intersection des droites l et m.
par(l,A) la droite parallèle à l passant par A.
rev(l) la droite dans la direction inverse de l.

Définition 10. *Le point A est situé strictement à droite de l ssi :*

$$RApt(A,l) \equiv LApt(A, rev(l))$$

Définition 11. *La droite l coupe m par la droite ssi :*

$$RCon(l,m) \equiv LCon(l, rev(m))$$

Définition 12. *On dit que l et m sont convergentes ssi :*

$$Con(l,m) \equiv Undir(l,m) \wedge Undir(l, rev(m))$$

Définition 13. *On dit que A est en dehors de l ssi :*

$$Apt(A,l) \equiv LApt(A,l) \vee RApt(A,l)$$

Axiomes à propos de la différence

$$\neg A\sharp A$$
$$A\sharp B \Rightarrow A\sharp C \vee B\sharp C$$
$$\neg a\sharp_l a$$
$$a\sharp_l b \Rightarrow a\sharp_l c \vee b\sharp_l c$$
$$\neg Undir(a,a)$$
$$Undir(a,b) \Rightarrow Undir(a,c) \vee Undir(b,c)$$
$$Con(l,m) \Rightarrow Con(l,n) \wedge Con(m,n)$$

Axiomes de raffinement

$$Undir(l,m) \vee Undir(l, rev(m))$$
$$Con(l,m) \Rightarrow LCon(l,m) \vee RCon(l,m)$$

Principes d'exclusion du raffinement

$$\neg(LApt(A,l) \wedge RApt(A,l))$$
$$\neg(LCon(l,m) \wedge RCon(l,m))$$

Axiomes de construction

$$A\sharp B \Rightarrow Inc(A, ln(A,B)) \wedge Inc(B, ln(A,B))$$
$$Con(l,m) \Rightarrow Inc(pt(l,m), l) \wedge Inc(pt(l,m), m)$$
$$Inc(A, par(l,A))$$
$$EqLn(l, rev(l))$$
$$Opp(ln(A,B), ln(B,A))$$
$$Dir(par(l,a), l)$$

Unicité des constructions

$$A\sharp B \wedge l\sharp_l m \Rightarrow \begin{array}{c} LApt(A,l) \vee LApt(B,l) \vee \\ LApt(A,m) \vee LApt(B,m) \vee \\ RApt(A,l) \vee RApt(B,l) \vee \\ RApt(A,m) \vee RApt(B,m) \end{array}$$

$$A\sharp B \wedge LApt(A,l) \Rightarrow LApt(B,l) \vee LCon(ln(A,B), l)$$

Axiomes de substitution

$$LApt(A,l) \Rightarrow A\sharp B \vee LApt(B,l)$$
$$LApt(A,l) \wedge Undir(l,m) \Rightarrow l\sharp_l m \vee RApt(A,m)$$
$$LCon(l,m) \Rightarrow Undir(m,n) \vee LCon(l,n)$$

1.3.5 Wu

Nous décrivons l'axiomatique de Wu. Cette axiomatique permet de caractériser la géométrie pour laquelle la méthode de démonstration automatique de Wu est complète. Nous évoquerons cette axiomatique dans le chapitre 5. L'axiomatique de Wu utilise des points et des droites comme objets de base.

Les précidats utilisés sont les suivants :

- appartenance d'un point à une droite
- perpendicularité de deux droites (notée $\perp$)
- congruence des segments (notée $\equiv$)

L'axiomatique de Wu comporte sept groupes d'axiomes.

Groupe I Incidence

Les axiomes d'incidence sont les même que ceux de Hilbert.

Définition 14. *On dit que deux droites sont parallèles si elles n'ont aucun point en commun. Cette relation est notée* $/\!/$.

Groupe II Axiome des parallèles

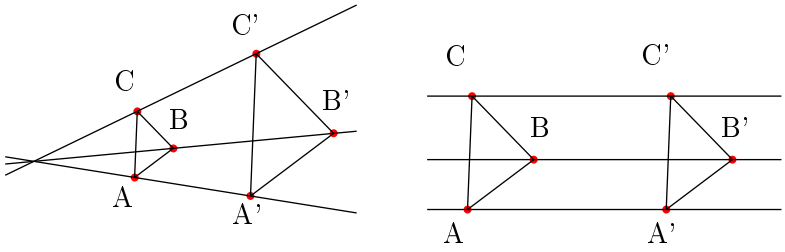
Etant donné un point A et une droite l il existe une et une seule droite parallèle à l passant par A.

Groupe III Axiome de l'infini

Il existe une infinité de points distincts.

Groupe IV Axiome de Desargues

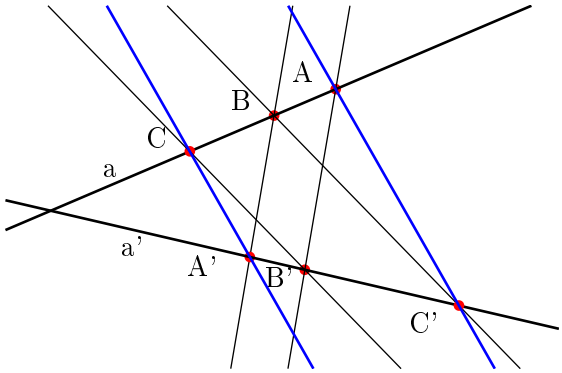
Axiome 1 Soient ABC et $A'B'C'$ deux triangles tels que $(AB) /\!/ (A'B')$, $(AC) /\!/ (A'C')$ et $(BC) /\!/ (B'C')$, alors soit les droites (AA'), (BB') et (CC') sont parallèles deux à deux soit elles sont concourantes.



Axiome 2 Soient ABC et $A'B'C'$ deux triangles tels que $(AB) /\!/ (A'B')$, $(AC) /\!/ (A'C')$, si (AA'), (BB') et (CC') sont parallèles deux à deux ou sont concourantes alors $(BC) /\!/ (B'C')$.

Groupe V Axiome de Pascal

Soit a et a' deux droites distinctes, A,B,C et A',B',C' des points distincts de a et a' respectivement. Si $(BC') /\!/ (B'C)$ et $(AB') /\!/ (A'B)$ alors $(AC') /\!/ (A'C)$.



Groupe VI Axiomes des perpendiculaires

Axiome VI.1 Si $a \perp a'$ alors $a' \perp a$.

Axiome VI.2 Étant donné un point O et une droite a, il existe une et une seule droite perpendiculaire à a passant par O.

Axiome VI.3 Si $a' \perp a$ et $a'' \perp a$ alors $a' /\!\!/ a''$.

Définition 15 (isotropique)**.** *Si $a \perp a$ alors on dit que a est isotropique.*

Axiome VI.4 Par n'importe quel point il passe au moins une droite non isotropique.

Axiome VI.5 Soit ABC un triangle et trois droites a,b et c passant par A , B et C respectivement telles que $a \perp BC$, $b \perp AC$ et $c \perp AB$, alors a, b et c sont concourantes.

Groupe VII Axiomes de congruence

Définition 16. *La médiatrice d'un segment $[AB]$ est la droite l passant par le milieu de $[AB]$ perpendiculaire à la droite (AB). Si la droite (AB) est isotropique alors c'est la droite (AB) elle-même, sinon l est non-isotropique.*

Axiome VII.1 Si a est la médiatrice des segments $[AA']$ et $[BB']$ alors $AB \equiv A'B'$.

Axiome VII.2 Si $AB \equiv CD$ et $CD \equiv EF$ alors $AB \equiv EF$.

Axiome VII.3 Tout couple de droites sécantes et non-isotropiques possède un axe de symétrie[4].

[4]Nous ne donnons pas la définition d'un axe de symétrie, celle-ci requiert de dériver quelques propositions des axiomes précédents nous renvoyons le lecteur à [Cho85]

Axiome VII.4 Soient A, B et C trois points distincts tels que $AB \equiv AC$ et les droites (AB) et (AC) sont non-isotropiques, alors il existe un axe de symétrie des droites (AB) et (AC) qui est aussi la médiatrice du segment $[BC]$.

1.3.6 Chou, Gao et Zhang

L'axiomatique de Chou, Gao et Zhang est une axiomatique très particulière. Elle a été définie afin de servir de fondements à la méthode de démonstration automatique appelée méthode des aires. C'est la méthode que nous avons formalisée dans l'assistant de preuve Coq, nous commentons cette formalisation dans le chapitre 4. Cette axiomatique utilise une approche à mi-chemin entre l'approche algébrique et l'approche purement géométrique. En effet, elle utilise les axiomes d'un corps mais contient aussi des axiomes géométriques tous basés sur deux quantités géométriques : l'aire orientée d'un triangle (notée $\mathcal{S}_{ABC}$) et le ratio de deux distances orientées portées par des droites parallèles (notée $\frac{\overline{\cdot\cdot}}{\overline{\cdot\cdot}}$). La géométrie affine est une conséquence des axiomes de Chou, Gao, Zhang [CGZ94]. Comme celle de Tarski, cette axiomatique est basée uniquement sur des points.

Corps

Le premier groupe d'axiomes est celui d'un corps. Nous supposons que nous avons un corps F de n'importe quelle caractéristique.

$$F \text{ est un corps}$$

Ratios de distances algébrique. Il existe une fonction d'arité quatre, des points dans le corps,

$$\frac{\overline{\cdot\cdot}}{\overline{\cdot\cdot}} : \text{Point} \to \text{Point} \to \text{Point} \to \text{Point} \to F$$

telle que si A, B et P sont colinéaires alors

$$\frac{\overline{AP}}{\overline{AB}} = -\frac{\overline{PA}}{\overline{AB}} = \frac{\overline{PA}}{\overline{BA}} = -\frac{\overline{AP}}{\overline{BA}}$$

et

$$\frac{\overline{PA}}{\overline{AB}} = 0 \iff P = A$$

Aire signée. Il existe une fonction d'arité trois, des points dans le corps.

$$\mathcal{S} : \text{Point} \to \text{Point} \to \text{Point} \to F$$

L'aire orientée d'un triangle est par définition l'opposée de l'aire du même triangle considéré dans le sens inverse.

$$\mathcal{S}_{ABC} = \mathcal{S}_{CAB} = \mathcal{S}_{BCA} = -\mathcal{S}_{BAC} = -\mathcal{S}_{CBA} = -\mathcal{S}_{ACB}$$

Si A, B et C sont non colinéaires alors $\mathcal{S}_{ABC} \neq 0$.

Axiome de Chasles. Si trois points A, B et C sont alignés alors la somme des distances orientées $\overline{AB}$ et $\overline{BC}$ est égale à $\overline{AC}$.

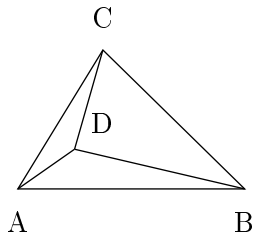
$$\mathcal{S}_{ABC} = 0 \to \overline{AB} + \overline{BC} = \overline{AC}$$

Dimension. Il existe trois points non colinéaires.

$$\exists\, A, B, C : \text{Point}, \mathcal{S}_{ABC} \neq 0$$

Quatre points du plan sont liés par la relation suivante :

$$\mathcal{S}_{ABC} = \mathcal{S}_{DBC} + \mathcal{S}_{ADC} + \mathcal{S}_{ABD}$$



Construction.
Existence :

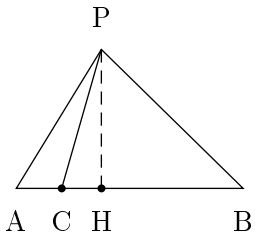
$$\forall r : F\ A \neq B \Rightarrow \exists P : \mathit{Point}, \mathcal{S}_{ABP} = 0 \wedge \frac{\overline{AP}}{\overline{AB}} = r \wedge \frac{\overline{AP}}{\overline{AB}} + \frac{\overline{PB}}{\overline{AB}} = 1$$

Unicité :

$$\forall r, A \neq B \quad \begin{array}{lllll} \wedge & \mathcal{S}_{ABP} & = 0 \wedge & \frac{\overline{AP}}{\overline{AB}} & = r \wedge \frac{\overline{AP}}{\overline{AB}} + \frac{\overline{PB}}{\overline{AB}} = 1 \\ \wedge & \mathcal{S}_{ABP'} & = 0 \wedge & \frac{\overline{AP'}}{\overline{AB}} & = r \wedge \frac{\overline{AP'}}{\overline{AB}} + \frac{\overline{P'B}}{\overline{AB}} = 1 \end{array} \to P = P'$$

Proportions. L'axiome des proportions permet de créer le lien entre l'aire orientée et la distance orientée.

$$A \neq C \rightarrow \mathcal{S}_{PAC} \neq 0 \rightarrow \mathcal{S}_{ABC} = 0 \rightarrow \frac{\overline{AB}}{\overline{AC}} = \frac{\mathcal{S}_{PAB}}{\mathcal{S}_{PAC}}$$



1.4 Résumé et conclusion

Afin de fournir au lecteur un aperçu général des axiomatiques présentées dans ce chapitre, nous donnons un tableau récapitulatif de leurs principales caractéristiques.

Axiomatique	Logique constructive	Objets de base	Prédicats	Nb axiomes
Heyting	✓	points, droites	différence, incidence	13
Hilbert 2	✘	points, droites	incidence point droite, congruence segments, congruence angles, appartenance à un segment	12
Hilbert 3	✘	points, droites, plans	incidence point droite, incidence point plan, congruence segments, congruence angles, appartenance à un segment	17
von Plato	✓	points, droites	différences points, différence droites, différence direction, strictement à gauche, intersecte par la gauche	22
Tarski	✘	points	congruence segments, appartenance à un segment	11
Wu	✘	points, droites	incidence, perpendicularité, congruence segments	17
Zhang	✘	points, corps	aire orientée, ratio de distances orientées	19

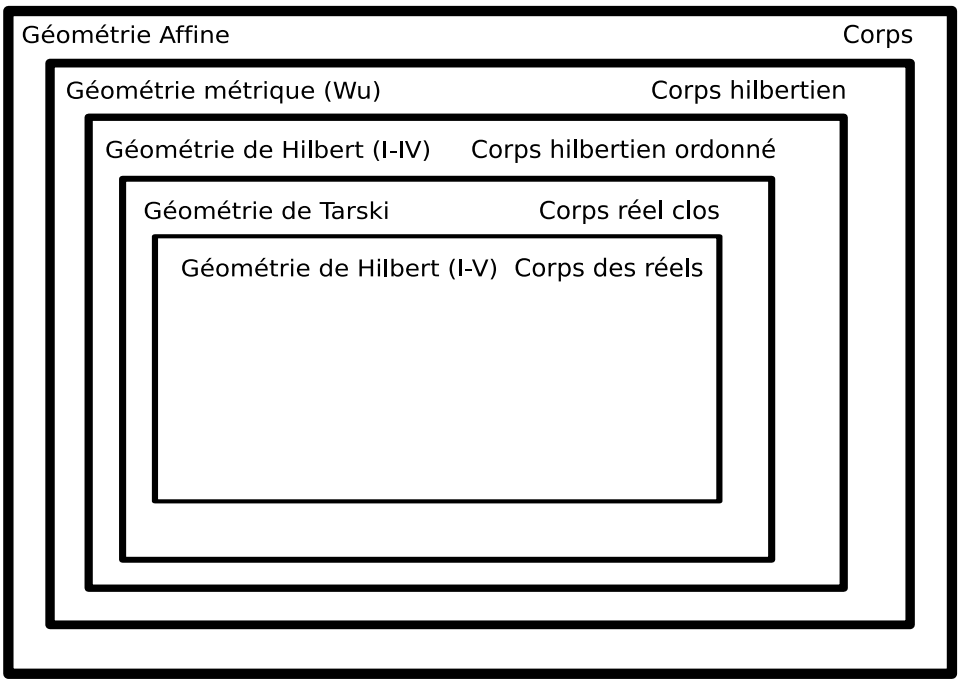


FIG. 1.2 – Diagrame réprésentant les inclusions entres modèles

Le tableau suivant fournit pour quelques axiomatiques, un corps sur lequel on peut construire un plan cartésien qui en est un modèle.

Corps	**Axiomes**
corps simple	Hilbert : I1-I3 IV
corps ordonné	Hilbert : I II IV III2-5
corps hilbertien	Wu
corps hilbertien ordonné	Hilbert : I II III IV
corps euclidien	Hilbert : I II III IV V2
corps réel clos	Tarski
corps des réels $\mathbb{R}$	Hilbert

Nous avons présenté les principales axiomatiques connues de la géométrie. Comme nous pouvons le voir sur le tableau récapitulatif, l'axiomatique de Tarski est l'une des plus simple, c'est celle que nous avons décidé de formaliser. Dans le chapitre suivant nous présentons notre développement de l'axiomatique de Tarski.

Chapitre 2

Formalisation de la géométrie de Tarski

2.1 Introduction

Comme nous l'avons vu en introduction, les preuves dans *les Éléments* d'Euclide ne sont pas aussi rigoureuses qu'on le voudrait. Dans *les fondements de la géométrie*, le but de Hilbert était de développer un traité dans lequel aucune intuition géométrique ne serait nécessaire pour vérifier la preuve d'un théorème. L'objet de ce chapitre est de faire passer les preuves du statut de *potentiellement mécanisable* à *mécanisé*.

La tâche qui consiste à mécaniser *les fondements de la géométrie* de Hilbert a été abordée par d'autres auteurs. Nous nous intéressons ici à la géométrie de Tarski. Nous avons utilisé l'assistant de preuve Coq. Nous n'allons pas détailler ici l'art et la manière d'utiliser Coq, pour cela nous invitons le lecteur à consulter la manuel de référence de Coq [Coq04, HKPM04] et le Coq'Art par Yves Bertot et Pierre Castéran [BC04a].

Au début des années 60, Wanda Szmielew et Alfred Tarski ont débuté un projet de création d'un traité sur les fondements de la géométrie. Un développement systématique de la géométrie euclidienne devait en constituer la première partie. Mais le décès prématuré de Wanda Szmielew a mis malheureusement un terme à ce projet. Finalement, Wolfram Schwabhäuser a repris le flambeau à partir du brouillon de Wanda Szmielew et Alfred Tarski. Il a pu publier le traité en 1983[1].

Nous présentons ici la formalisation des huit premiers chapitres de ce traité [SST83].

Nous décrivons d'abord les différentes versions de l'axiomatique de Tarski.

[1] Nous tirons ces informations de l'introduction à la lettre de Tarski adressée à Schwabhäuser en 1978, publiée par Givant en 1999 [TG99]

Puis nous donnons un aperçu de notre formalisation en donnant un exemple de théorème dont nous détaillons la preuve. Enfin nous comparons notre formalisation avec les développements précédents et traitons des cas dégénérés.

2.2 Travaux connexes et motivations

Outre l'intérêt en soi de la formalisation, nous visons à terme deux applications, la première est l'utilisation d'un assistant de preuve pour enseigner la géométrie [Nar05], la seconde est la preuve de programmes dans le domaine de la géométrie algorithmique.

Dans [DDS00], Christophe Dehlinger, Jean-François Dufourd et Pascal Schreck proposent une formalisation dans l'assistant de preuve Coq de l'axiomatique de Hilbert ainsi que des propositions à propos des trois premiers groupes d'axiomes. Le but de cette formalisation était d'une part de servir de fondation à l'étude de problèmes algorithmiques et d'autre part d'étudier l'aspect intuitionniste ou non des preuves. La conclusion de cette formalisation est qu'il est nécessaire (dans le sens où les auteurs n'ont pas pu l'éviter) d'utiliser la décidabilité de l'égalité entre deux points.

Une seconde formalisation de la géométrie de Hilbert a été réalisée par Laura Meikle et Jacques Fleuriot [MF03]. Cette formalisation a été menée à bien au moyen du système Isabelle [Pau06, NPW], donc comme la logique sous-jacente à ce système est une logique classique, cette formalisation ne s'intéresse pas au problème de savoir si les preuves sont constructives.

Ces deux formalisations ont permis d'arriver à la conclusion que les preuves de Hilbert ne sont pas parfaitement formelles. En particulier, les cas dégénérés, comme lorsque deux points coïncident ou trois points sont alignés, ne sont souvent pas traités explicitement. Une analyse précise des *fondements de la géométrie* montre que d'une édition à l'autre, Hilbert a parfois changé des axiomes mais il n'a pas toujours changé les preuves qui en dépendent. Les preuves peuvent être rendues plus rigoureuse grâce à l'utilisation de l'assistant de preuve. Si on change un axiome, il est alors facile de vérifier quelles sont les preuves qui restent valides[2].

D'autres travaux ont été réalisés à propos de la formalisation de la géométrie. Gilles Khan a formalisé l'axiomatique constructive de von Plato dans le système Coq [Kah95, vP95].

Le plus grand développement concernant la géométrie formelle est celui réalisé par Frédérique Guilhot [Gui05].

Le développement de Frédérique Guilhot est très différent des précédents car il s'inscrit dans un projet pédagogique. Le développement a été créé dans l'idée d'être utilisé dans le cadre de l'enseignement de la géométrie. Les

[2] En pratique, l'utilisation d'un minimum d'automatisation permet de rendre les preuves un peu moins sensibles à des modifications d'axiomes ou de lemmes.

axiomes et définitions utilisés sont adaptés à la façon dont est enseignée la géométrie en France.

Le but de notre formalisation comparée à celle de Frédérique Guilhot [Gui05] est de fournir un développement de bas niveau pour la géométrie.

La formalisation que nous avons réalisée à partir de l'axiomatique de Tarski a l'avantage d'être basée sur une axiomatique très simple : deux prédicats et onze axiomes. Mais cette simplicité a un prix. En effet, cette formalisation n'est pas du tout adaptée au contexte de l'éducation, puisque certaines propriétés pourtant intuitivement très simples requièrent l'introduction préalable de nombreux autres résultats. C'est le cas par exemple pour l'existence du milieu d'un segment qui ne peut être obtenue qu'à la fin du chapitre huit (soit après environ 150 lemmes et 4000 lignes de Coq). Le faible nombre d'axiomes impose un ordonnancement des lemmes qui n'est souvent pas très intuitif et donc les preuves sont plutôt longues et difficiles par rapport à l'énoncé que l'on prouve.

La formalisation de la géométrie dans un assistant de preuve n'a pas seulement l'intérêt de fournir un très haut niveau de confiance en les preuves formalisées, cela permet aussi de combiner des preuves purement géométriques avec d'autres types de preuves. Les assistants de preuve comme le système Coq couvrent un large champ d'applications. On pourra ainsi tout aussi bien réaliser des preuves par induction sur le nombre de points d'un polygone, ou bien utiliser les nombres complexes ou encore réaliser des preuves de programmes en algorithmique géométrique.

Concernant la deuxième application que nous visons, à savoir la preuve de programmes en algorithmique géométrique, on peut citer les formalisations d'algorithmes de calcul d'enveloppe convexe par David Pichardie et Yves Bertot en Coq [PB01] et par Laura Meikle et Jacques Fleuriot en Isabelle [MF05]. Christophe Dehlinger et Jean-François Dufourd ont formalisé le théorème de classification des surfaces en Coq [DD04a, DD04b].

2.3 Retour sur l'axiomatique de Tarski

Alfred Tarski a travaillé sur l'axiomatisation et sur les méta-mathématiques de la géométrie euclidienne par intermittence de 1926 jusqu'à sa mort en 1983. Ainsi, de son travail et de celui de ses étudiants sont nées plusieurs versions de son axiomatique. Cette partie est organisée de la manière suivante : nous donnons d'abord la liste et une description en termes informels de toutes les propositions qui sont apparues comme axiomes dans au moins une version de l'axiomatique de Tarski, puis après avoir résumé l'historique de ces différentes versions, nous présentons la version que nous avons formalisée.

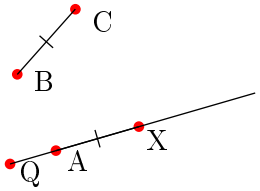


FIG. 2.1 – Axiome de construction d'un segment

2.3.1 Description des axiomes

Nous reproduisons ici la liste des propositions qui apparaissent dans diverses versions de l'axiomatique de Tarski. Nous adoptons la même numérotation que dans [TG99]. Les variables libres sont considérées comme quantifiées universellement.

1 Réflexivité de la congruence des segments

$$AB \equiv BA$$

2 Pseudo-transitivité de la congruence des segments

$$AB \equiv PQ \wedge AB \equiv RS \Rightarrow PQ \equiv RS$$

3 Identité pour la congruence des segments

$$AB \equiv CC \Rightarrow A = B$$

Axiome de construction d'un segment

4 Construction d'un segment

$$\exists X, \beta_T\ Q\ A\ X \wedge AX \equiv BC$$

L'axiome de construction d'un segment permet de construire un point sur une demi-droite $[QA)$ à une certaine distance BC de A.

Axiome des cinq segments

5 Cinq segments

$$A \neq B \wedge \beta_T\ A\ B\ C \wedge \beta_T\ A'\ B'\ C' \wedge$$
$$AB \equiv A'B' \wedge BC \equiv B'C' \wedge AD \equiv A'D' \wedge BD \equiv B'D' \Rightarrow CD \equiv C'D'$$

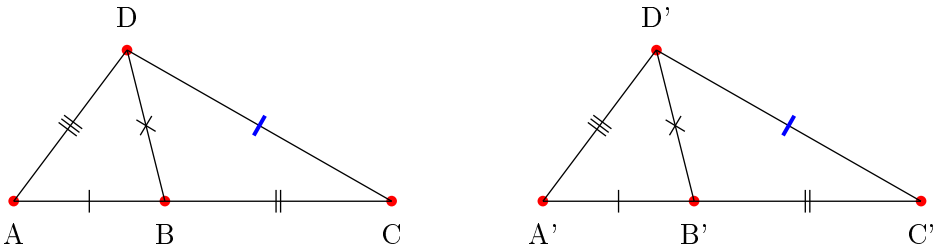


FIG. 2.2 – Axiome des cinq segments

5_1 Cinq segments (variante)

$$A \neq B \wedge B \neq C \wedge \beta_T\ A\,B\,C \wedge \beta_T\ A'\,B'\,C' \wedge \\ AB \equiv A'B' \wedge BC \equiv B'C' \wedge AD \equiv A'D' \wedge BD \equiv B'D' \Rightarrow CD \equiv C'D'$$

Cette variante ne diffère de l'axiome précédent que par la présence de la condition supplémentaire $B \neq C$.

6 Identité pour β_T

$$\beta_T\ A\,B\,A \Rightarrow A = B$$

Axiome de Pasch

L'axiome de Pasch original énonce le fait que si une droite coupe un côté d'un triangle sans passer par l'un des sommets du triangle, alors elle coupe l'un des deux autres côtés.

7 Pasch (intérieur)

$$\beta_T\ A\,P\,C \wedge \beta_T\ B\,Q\,C \Rightarrow \exists X, \beta_T\ P\,X\,B \wedge \beta_T\ Q\,X\,A$$

7_1 Pasch (extérieur)

$$\beta_T\ A\,P\,C \wedge \beta_T\ Q\,C\,B \Rightarrow \exists X, \beta_T\ A\,X\,Q \wedge \beta_T\ B\,P\,X$$

7_2 Pasch (extérieur) (Variante)

$$\beta_T\ A\,P\,C \wedge \beta_T\ Q\,C\,B \Rightarrow \exists X, \beta_T\ A\,X\,Q \wedge \beta_T\ X\,P\,B$$

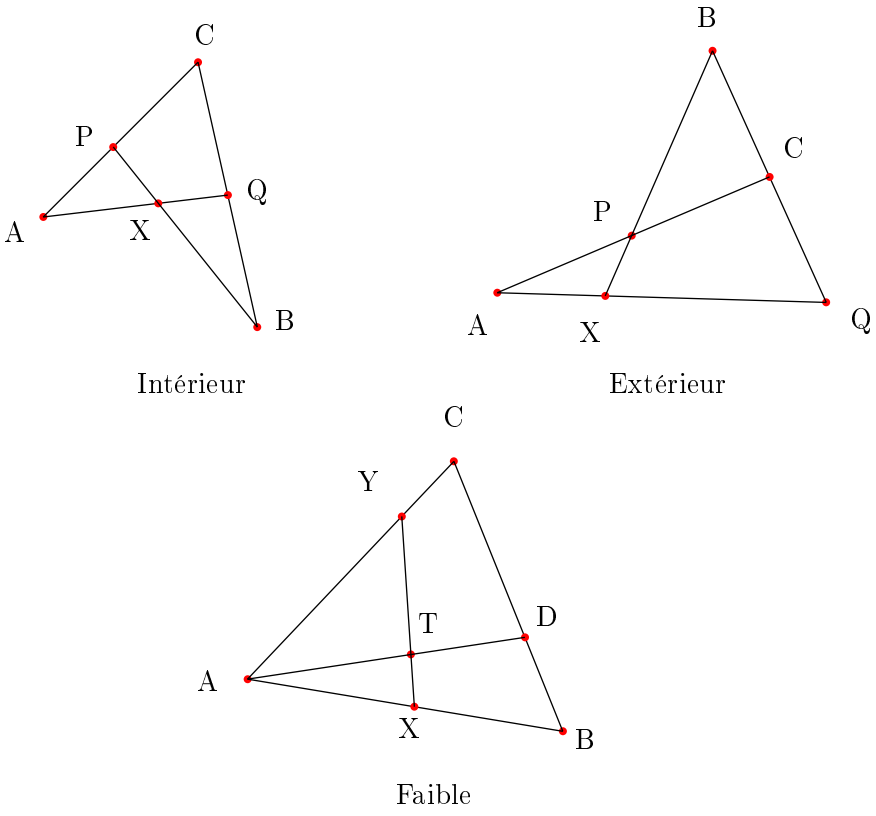


FIG. 2.3 – Axiomes de Pasch

7_3 Pasch faible

$$\beta_T\ A\,T\,D \wedge \beta_T\ B\,D\,C \Rightarrow \exists X, Y, \beta_T\ A\,X\,B \wedge \beta_T\ A\,Y\,C \wedge \beta_T\ Y\,T\,X$$

Axiomes de dimension

Les axiomes de dimension servent à borner la dimension de l'espace. Notons que les axiomes de borne inférieure n sont exactement la négation des axiomes de borne supérieure $n-1$.

8(1) Dimension, borne inférieure 1

$$\exists AB, A \neq B$$

Il existe deux points distincts.

8(2) Dimension, borne inférieure 2

$$\exists ABC, \neg\beta_T\ A\,B\,C \wedge \neg\beta_T\ B\,C\,A \wedge \neg\beta_T\ C\,A\,B$$

Il existe trois points non colinéaires.

8(n) Dimension, borne inférieure n

$$\exists ABCP_1P_2\ldots P_{n-1},\ \begin{array}{l}\bigwedge_{1\leq i<j<n} P_i \neq P_j \wedge \\ \bigwedge_{i=2}^{n-1} AP_1 \equiv AP_i \wedge BP_1 \equiv BP_i \wedge CP_1 \equiv CP_i \wedge \\ \neg\beta_T\ A\,B\,C \wedge \neg\beta_T\ B\,C\,A \wedge \neg\beta_T\ C\,A\,B\end{array}$$

9(0) Dimension, borne supérieure 0

$$A = B$$

Tous les points sont égaux.

9(1) Dimension, borne supérieure 1

$$\beta_T\ A\,B\,C \vee \beta_T\ B\,C\,A \vee \beta_T\ C\,A\,B$$

Tout les points sont sur la même droite.

9(n) Dimension, borne supérieure n

$$\begin{array}{l}\bigwedge_{1\leq i<j\leq n} P_i \neq P_j \wedge \\ \bigwedge_{i=2}^{n} \begin{array}{l} AP_1 \equiv AP_i \wedge \\ BP_1 \equiv BP_i \wedge \\ CP_1 \equiv CP_i \end{array}\end{array} \Rightarrow \beta_T\ A\,B\,C \vee \beta_T\ B\,C\,A \vee \beta_T\ C\,A\,B$$

$9_1(2)$ Dimension, borne supérieure 2 (variante)

$$\exists Y, (ColXY\,A \wedge \beta_T\; B\,Y\,C) \vee (ColXY\,B \wedge \beta_T\; C\,Y\,A) \vee (ColXY\,C \wedge \beta_T\; A\,Y\,B)$$

où $ColABC$ est défini par $\beta_T\; A\,B\,C \vee \beta_T\; B\,C\,A \vee \beta_T\; C\,A\,B$.

Axiome d'Euclide

Le célèbre axiome d'Euclide prend ici des formes inhabituelles.
- La première forme énonce le fait que si un point est situé à l'intérieur d'un angle $\angle ABC$ alors il y a une droite qui intersecte chacun des deux côtés de l'angle.
- La seconde forme énonce l'existence du centre du cercle circonscrit à un triangle non dégénéré.
- La troisième forme correspond au théorème de la droite des milieux.

10 Axiome d'Euclide

$$\beta_T\; A\,D\,T \wedge \beta_T\; B\,D\,C \wedge A \neq D \Rightarrow \exists X, Y \beta_T\; A\,B\,X \wedge \beta_T\; A\,C\,Y \wedge \beta_T\; X\,T\,Y$$

10_1 Axiome d'Euclide (variante)

$$\beta_T\; A\,D\,T \wedge \beta_T\; B\,D\,C \wedge A \neq D \Rightarrow \exists X, Y \beta_T\; A\,B\,X \wedge \beta_T\; A\,C\,Y \wedge \beta_T\; Y\,T\,X$$

10_2 Axiome d'Euclide (seconde forme)

$$\beta_T\; A\,B\,C \vee \beta_T\; B\,C\,A \vee \beta_T\; C\,A\,B \vee \exists X, AX \equiv BX \wedge AX \equiv CX$$

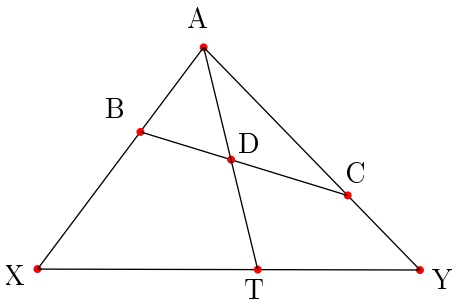
10_3 Axiome d'Euclide (troisième forme)

$$\begin{array}{l} \beta_T\; A\,B\,F \wedge AB \equiv BF \wedge \\ \beta_T\; A\,D\,E \wedge AD \equiv DE \wedge \\ \beta_T\; B\,D\,C \wedge BD \equiv DC \end{array} \Rightarrow BC \equiv FE$$

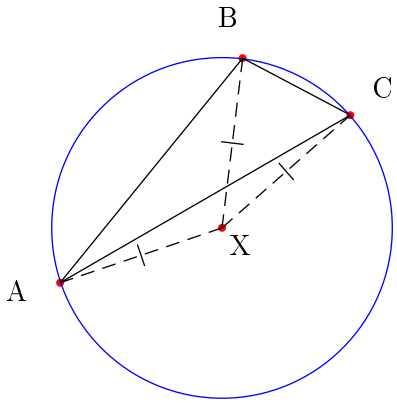
Axiome de continuité

11 Continuité (au second ordre)

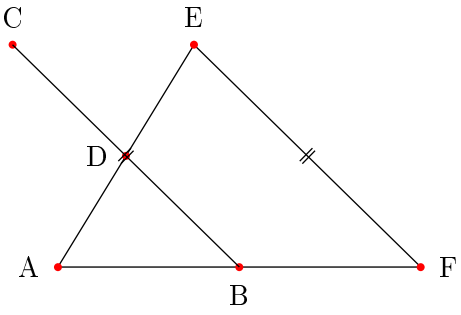
$$\exists a\, \forall xy\, (x \in X \wedge y \in Y \Rightarrow \beta_T\; a\,x\,y) \Rightarrow \exists b\, \forall xy\, x \in X \wedge y \in Y \Rightarrow \beta_T\; x\,b\,y$$



Première forme



Deuxième forme



Troisième forme

FIG. 2.4 – Axiomes d'Euclide

Schéma 11 Continuité (schéma)

$$\exists a \forall xy\,(\alpha \wedge \beta \Rightarrow \beta_T\ a\,x\,y) \Rightarrow \exists b \forall xy\,\alpha \wedge \beta \Rightarrow \beta_T\ x\,b\,y$$

où α etβ sont des formules du premier ordre, telles que a,b et y n'apparaissent pas libres dans α et a,b et x n'apparaissent pas libres dans β.

Quand le schéma[3] d'axiomes est utilisé en lieu et place de l'axiome 11, la géométrie est dite élémentaire. L'axiome 11 est le seul axiome qui n'est pas une formule de la logique du premier ordre[4]. Il est l'analogue géométrique de l'axiome de Dedekind.

Réflexivité et symétrie de β_T

12 Pseudo-réflexivité de β_T

$$\beta_T\ A\,B\,B$$

B est toujours entre A et B.

14 Symétrie de β_T

$$\beta_T\ A\,B\,C \Rightarrow \beta_T\ C\,B\,A$$

Si B est entre A et C alors il est entre C et A.

Axiomes concernant l'égalité

13 Compatibilité de l'égalité avec β_T

$$A = B \Rightarrow \beta_T\ A\,B\,A$$

19 Compatibilité de l'égalité avec $\equiv$

$$A = B \Rightarrow AC \equiv BC$$

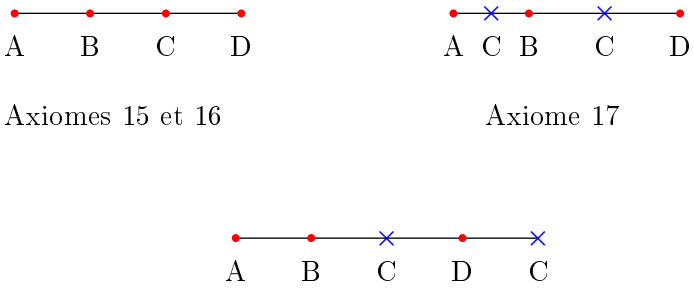
Axiomes de (pseudo-)transitivité pour β_T

15 Transitivité (intérieure) de β_T

$$\beta_T\ A\,B\,D \wedge \beta_T\ B\,C\,D \Rightarrow \beta_T\ A\,B\,C$$

[3] Le terme schéma d'axiomes désigne la description d'une infinité d'axiomes. Le fait pour un énoncé d'être ou non un axiome doit bien sûr toujours être décidable.

[4] Comme nous l'avons vu au chapitre précédent, Hilbert a prouvé que la géométrie élémentaire n'est pas finiment axiomatisable. Cela signifie qu'il n'existe pas de système d'axiomes fini *exprimés dans le même langage* qui est équivalent à la géométrie élémentaire. Comme Florent Kirchner l'a montré [Kir06], il est possible d'obtenir un système d'axiomes fini, mais exprimés dans un langage étendu.



Axiomes 15 et 16 — Axiome 17

Axiome 18

FIG. 2.5 – Transitivité et pseudo-transitivités

16 Transitivité (extérieure) de β_T

$$\beta_T\ A\,B\,C \wedge \beta_T\ B\,C\,D \wedge B \neq C \Rightarrow \beta_T\ A\,B\,D$$

17 Pseudo-transitivité (intérieure) de β_T

$$\beta_T\ A\,B\,D \wedge \beta_T\ A\,C\,D \Rightarrow \beta_T\ A\,B\,C \vee \beta_T\ A\,C\,B$$

18 Pseudo-transitivité (extérieure) de β_T

$$\beta_T\ A\,B\,C \wedge \beta_T\ A\,B\,D \wedge A \neq B \Rightarrow \beta_T\ A\,C\,D \vee \beta_T\ A\,D\,C$$

Axiomes de construction de triangles

20 Unicité de la construction du triangle

$$\begin{array}{ll} AC \equiv AC' \wedge BC \equiv BC' \wedge & \\ \beta_T\ A\,D\,B \wedge \beta_T\ A\,D'\,B \wedge \beta_T\ C\,D\,X \wedge & \Rightarrow C = C' \\ \beta_T\ C'\,D'\,X \wedge D \neq X \wedge D' \neq X & \end{array}$$

20_1 Unicité de la construction du triangle (variante)

$$\begin{array}{ll} A \neq B \wedge & \\ AC \equiv AC' \wedge BC \equiv BC' \wedge & \Rightarrow C = C' \\ \beta_T\ B\,D\,C' \wedge (\beta_T\ A\,D\,C \vee \beta_T\ A\,C\,D) & \end{array}$$

21 Existence de la construction du triangle

$$AB \equiv A'B' \Rightarrow \exists CX, \begin{array}{l} AC \equiv A'C' \wedge BC \equiv B'C' \wedge \\ \beta_T\ C\ X\ P \wedge (\beta_T\ A\ B\ X \vee \beta_T\ B\ X\ A \vee \beta_T\ X\ A\ B) \end{array}$$

2.3.2 Historique des axiomatiques de Tarski

Nous nous appuyons sur [TG99] et sur les notes de bas de page de [Tar51] pour retracer l'historique des différentes versions de l'axiomatique de Tarski. Tarski a commencé à travailler sur son axiomatique dès 1926 et l'a présentée dans son cours à l'université de Varsovie. Elle est été soumise à publication en 1940 puis publiée dans sa première forme en 1967 [Tar67]. Cette version contient 20 axiomes plus un schéma d'axiomes. Une deuxième version un peu plus simple a été publiée dans [Tar51]. Cette première simplification consiste uniquement à considérer une logique munie d'une égalité, les axiomes 13 et 19 sont alors superflus. Cette seconde version fut simplifiée à nouveau par Eva Kallin, Scott Taylor et Alfred Tarski pour arriver à un ensemble de douze axiomes [Tar59]. La dernière simplification obtenue, l'a été par Gupta dans sa thèse [Gup65], il donne la preuve que deux autres axiomes peuvent être éliminés. C'est cette dernière version que nous utilisons dans notre formalisation.

Le tableau 2.1 fournit la liste des axiomes contenus dans chacune de ces axiomatiques. Pour plus de clarté nous résumons la liste des axiomes que nous avons utilisés dans notre formalisation. AU lieu du schéma, nous utilisons l'axiome 11 qui est exprimable en logique d'ordre supérieur.

TAB. 2.1 – L'axiomatique de Tarski ; historique et version formalisée (11 axiomes).

Année :	1940	1951	1959	1965	1983
Référence :	[Tar67]	[Tar51]	[Tar59]	[Gup65]	[SST83]
Axiomes :	1	1	1	1	1
	2	2	2	2	2
	3	3	3	3	3
	4	4	4	4	4
	5_1	5_1	$\rightarrow 5$	5	5
	6	6	6		6
	7_2	7_2	$\rightarrow 7_1$	7_1	$\rightarrow 7$
	8(2)	8(2)	8(2)	8(2)	8(2)
	$9_1(2)$	$9_1(2)$	$\rightarrow 9(2)$	9(2)	9(2)
	10	10	$\rightarrow 10_1$	10_1	$\rightarrow 10$
	11	11	11	11	11
	12	12			
	13				
	14	14			
	15	15	15	15	
	16	16			
	17	17			
	18	18	18		
	19				
	20	$\rightarrow 20_1$			
	21	21			
Nb. d'axiomes :	20	18	12	10	10
	+	+	+	+	+
	1 schéma	1 schéma	1 schéma	1 schéma	1 schéma

Réflexivité	1	$AB \equiv BA$
Transitivité	2	$AB \equiv CD \wedge AB \equiv EF \Rightarrow CD \equiv EF$
Identité	3	$AB \equiv CC \Rightarrow A = B$
Construction	4	$\exists E, \beta_T\ A\,B\,E \wedge BE \equiv CD$
5 segments	5	$AB \equiv A'B' \wedge BC \equiv B'C' \wedge$ $AD \equiv A'D' \wedge BD \equiv B'D' \wedge$ $\beta_T\ A\,B\,C \wedge \beta_T\ A'\,B'\,C' \wedge A \neq B \Rightarrow CD \equiv C'D'$
Identité	6	$\beta_T\ A\,B\,A \Rightarrow (A = B)$
Pasch	7	$\exists X, \beta_T\ A\,P\,C \wedge \beta_T\ B\,Q\,C \Rightarrow \beta_T\ P\,x\,B \wedge \beta_T\ Q\,x\,A$
Dim. inférieure	8(2)	$\exists ABC, \neg\beta_T\ A\,B\,C \wedge \neg\beta_T\ B\,C\,A \wedge \neg\beta_T\ C\,A\,B$
Dim. supérieure	9(2)	$AP \equiv AQ \wedge BP \equiv BQ \wedge CP \equiv CQ \wedge P \neq Q$ $\Rightarrow \beta_T\ A\,B\,C \vee \beta_T\ B\,C\,A \vee \beta_T\ C\,A\,B$
Euclide	10	$\exists XY, \beta_T\ A\,D\,T \wedge \beta_T\ B\,D\,C \wedge A \neq D \Rightarrow$ $\beta_T\ P\,x\,B \wedge \beta_T\ Q\,x\,A$
Continuité	11	$\forall XY, (\exists A, (\forall xy, x \in X \wedge y \in Y \Rightarrow \beta_T\ A\,x\,y)) \Rightarrow$ $\exists B, (\forall xy, x \in X \Rightarrow y \in Y \Rightarrow \beta_T\ x\,B\,y)$.

2.4 Aperçu de la formalisation

La formalisation que nous avons réalisée prouve formellement que les simplifications successives de l'axiomatique de Tarski sont correctes, nous prouvons formellement que les axiomes superflus sont dérivables à partir des autres axiomes.

Les sections suivantes contiennent un rapide tour d'horizon de contenu de chacun des chapitres :

Le premier chapitre contient l'axiomatique ainsi que la définition du prédicat « colinéaire » (noté `Col`).

Tout d'abord on suppose qu'il existe un type pour les objets que l'on manipule, on l'appelle Point :

Parameter *Point* : *Set*.

Ensuite, on suppose que l'on a deux prédicats sur les points, l'un quaternaire et l'autre ternaire[5] :

Parameter *Cong* : $Point \to Point \to Point \to Point \to Prop$.

Parameter *Bet* : $Point \to Point \to Point \to Prop$.

Ensuite nous définissons le prédicat Col afin de simplifier l'énoncé de certains axiomes :

Definition *Col* $(A\ B\ C : Point) : Prop :=$
$Bet\ A\ B\ C \vee Bet\ B\ C\ A \vee Bet\ C\ A\ B$.

Nous ne détaillons pas tous les axiomes, nous donnons seulement l'axiome de continuité :

Axiom *continuity* :
$\forall\ X\ Y : Point \to Prop,$
$(\exists\ A : Point, (\forall\ x\ y : Point, X\ x \to Y\ y \to Bet\ A\ x\ y)) \to$
$\exists\ B : Point, (\forall\ x\ y : Point, X\ x \to Y\ y \to Bet\ x\ B\ y).$

Le second chapitre contient quelques propriétés de base du prédicat de congruence de longueur des segments (noté `Cong`). Il contient aussi la preuve de l'unicité du point construit par l'axiome 4.

Le troisième chapitre contient quelques propriétés du prédicat qui exprime qu'un point appartient à un segment (noté `Bet`). Il contient en particulier les preuves des propositions 12, 14 et 16. C'est dans ce chapitre que l'axiome qui donne une borne inférieure pour la dimension est utilisé pour la première fois.

Le quatrième chapitre contient diverses propriétés de `Cong`, `Col` et `Bet`.

[5] Pour des raisons pratiques, il est d'usage en Coq de définir les fonctions qui prennent plusieurs arguments de manière curryfiée (c'est à dire sour la forme $A \Rightarrow B \Rightarrow C$ plutôt que $A \wedge B \Rightarrow C$). Si le lecteur le désire il peut penser le type de *Cong* comme : (*Point* , *Point* , *Point* , *Point*) $\to$ *Prop*.

Le cinquième chapitre contient quelques propriétés de transitivité de `Bet` et la définition d'un prédicat de comparaison de longueurs (noté `le`) ainsi que les propriétés associées. Il contient en particulier la preuve des propositions 17 et 18.

Le sixième chapitre définit un prédicat de non appartenance à un segment noté `out` qui est utilisé pour prouver quelques propriétés de `Cong`, `Col` et `Bet` notamment des propriétés de transitivité de `Col`.

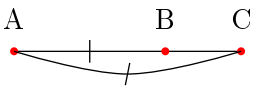
Le septième chapitre définit le milieu, le symétrique d'un point et en prouve quelques propriétés. Il faut noter qu'à ce stade nous ne pouvons pas encore prouver l'existence du milieu.

Le huitième chapitre contient la définition du prédicat « perpendiculaire » (noté `Perp`), quelques propriétés associées ainsi que l'existence du projeté orthogonal. On peut alors enfin prouver l'existence du milieu d'un segment.

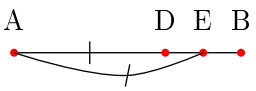
2.4.1 Deux lemmes cruciaux

Dans notre formalisation, nous suivons de près le texte de Schwabhäuser, Szmielew et Tarski excepté au chapitre cinq où nous avons introduit deux lemmes cruciaux qui ne figurent pas dans le texte original. Ces deux lemmes permettent de déduire l'égalité de deux points qui figurent à la même position sur un segment.

$$\forall ABC, \beta_T\ A\,B\,C \wedge AC \equiv AB \Rightarrow C = B$$



$$\forall ABDE, \beta_T\ A\,D\,B \wedge \beta_T\ A\,E\,B \wedge AD \equiv AE \Rightarrow D = E.$$



2.4.2 Un exemple de preuve

Nous reproduisons ici une des preuves non triviales de [SST83] : la preuve due à Gupta que l'axiome 18 peut être dérivé à partir des autres axiomes.

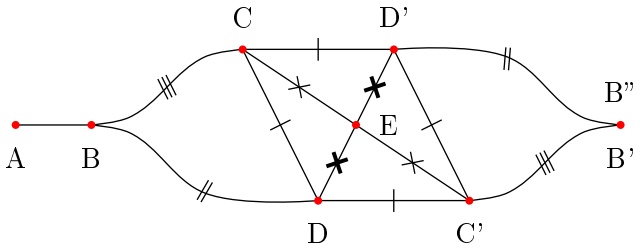


FIG. 2.6 – Preuve de l'axiome 18

Nous donnons la traduction de la preuve telle qu'elle est donnée dans [SST83] avec en parallèle la preuve formelle sous forme de script Coq.

Cet exemple montre que la preuve formelle est d'une taille raisonnable comparée à la preuve informelle. A l'échelle du développement, si nous omettons le premier chapitre qui contient presque uniquement les axiomes, les sept chapitres suivants occupent 41 pages et correspondent à environ 4000 lignes de Coq. Nous obtenons un facteur de *de Bruijn*, (le rapport entre la taille du texte original et de sa formalisation [Wie]), de 100 lignes par page.

Proposition 7 (Gupta). $A \neq B \wedge \beta_T\ A\,B\,C \wedge \beta_T\ A\,B\,D \Rightarrow \beta_T\ A\,C\,D \vee \beta_T\ A\,D\,C$

Preuve:
Soient C' et D' des points tels que[6] *:*

$$\beta_T\ A\,D\,C' \wedge DC' \equiv CD \text{ et } \beta_T\ A\,C\,D' \wedge CD' \equiv CD$$

```
assert (exists C', Bet A D C' /\ Cong D C' C D)...
DecompExAnd H2 C'.
assert (exists D', Bet A C D' /\ Cong C D' C D)...
DecompExAnd H2 D'.
```

Il faut donc montrer que $C = C'$ ou $D = D'$.
Soient B et B'' des points tels que :

$$\beta_T\ A\,C'\,B' \wedge C'B' \equiv CB \text{ et } \beta_T\ A\,D'\,B'' \wedge D'B'' \equiv DB$$

```
assert (exists B', Bet A C' B' /\ Cong C' B' C B)...
DecompExAnd H2 B'.
assert (exists B'', Bet A D' B'' /\ Cong D' B'' D B)...
DecompExAnd H2 B''.
```

[6] La tactique `DecompExAnd H A` suppose que `H` est une hypothèse de la forme $\exists x\ \bigwedge_i P_i(x)$. Elle introduit un point A dans le contexte et décompose la conjonction en différentes hypothèses.

D'après le lemme 2.11[7] *nous pouvons en déduire que* $BC' \equiv B''C$ *et que* $BB' \equiv B''B$.

```
assert (Cong B C' B'' C).
eapply l2_11.
3:apply cong_commutativity.
3:apply cong_symmetry.
3:apply H11.
Between.
Between.
esTarski.
assert (Cong B B' B'' B).
eapply l2_11;try apply H2;Between.
```

Par unicité de la construction, on sait que $B'' = B'$.

```
assert (B''=B').
apply construction_unicity with
(Q:=A) (A:=B) (B:=B'') (C:=B) (x:=B'') (y:=B');Between...
smart_subst B''.
```

Nous savons donc que les points $BCD'C'B'C'DC$ *forment une configuration des cinq segments. Ceci est noté* $FSC \begin{pmatrix} BCD'C' \\ B'C'DC \end{pmatrix}$.

```
assert (FSC B C D' C' B' C' D C).
unfold FSC;repeat split;unfold Col;Between;sTarski.
2:eapply cong_transitivity.
2:apply H7.
2:sTarski.
apply l2_11 with (A:=B) (B:=C) (C:=D') (A':=B') (B':=C') (C':=D);
Between;sTarski;esTarski.
```

D'où $C'D' \equiv CD$ *(car dans le cas* $B \neq C$ *l'axiome des cinq segments permet de conclure et dans le cas* $B = C$ *par hypothèse)*[8].

```
assert (Cong C' D' C D).
cases_equality B C.
(* First case *)
treat_equalities.
eapply cong_transitivity.
apply cong_commutativity.
apply H11.
```

[7] Le lemme 2.11 est le suivant :
$\forall ABCA'B'C'\ \beta_T\ A\,B\,C \Rightarrow \beta_T\ A'\,B'\,C' \Rightarrow AB \equiv A'B' \Rightarrow BC \equiv B'C' \Rightarrow AC \equiv A'C'$

[8] Cette étape utilise la tactique `treat_equalities` qui permet de simplifier le but dans les cas dégénérés, ceci fait l'objet du paragraphe suivant.

```
Tarski.
(* Second case *)
apply cong_commutativity.
eapply l4_16;try apply H3...
```

D'après l'axiome de Pasch, il existe un point E tel que :

$$\beta_T\ C\ E\ C' \wedge \beta_T\ D\ E\ D'$$

```
assert (exists E, Bet C E C' /\ Bet D E D').
eapply inner_pash;Between.
DecompExAnd H13 E.
```

Nous pouvons en déduire que $IFS \begin{pmatrix} ded'c \\ ded'c' \end{pmatrix}$ *et* $IFS \begin{pmatrix} cec'd \\ cec'd' \end{pmatrix}$.

```
assert (IFSC D E D' C D E D' C').
unfold IFSC;repeat split;Between;sTarski.
eapply cong_transitivity.
apply cong_commutativity.
apply H7.
sTarski.

assert (IFSC C E C' D C E C' D').
unfold IFSC;repeat split;Between;sTarski.
eapply cong_transitivity.
apply cong_commutativity.
apply H5.
sTarski.
```

D'où $EC \equiv EC'$ *et* $ED \equiv ED'$.

```
assert (Cong E C E C').
eapply l4_2;eauto.
assert (Cong E D E D').
eapply l4_2;eauto.
```

Supposons que $C \neq C'$. *Il faut montrer que* $D = D'$[9].

```
cases_equality C C'.
smart_subst C'.
assert (E=C).
eTarski.
smart_subst E.
unfold IFSC, FSC,Cong_3 in *;intuition.
```

[9]Notons que cette phrase utilise la décidabilité de l'égalité de deux points

D'après l'hypothèse, on a aussi $C \neq D'$.

```
assert (C<>D').
unfold not;intro.
treat_equalities...
```

D'après l'axiome de construction d'un segment, il existe des points P, Q *et* R *tels que :*

$\beta_T\, C'\, C\, P \wedge CP \equiv CD'$ *et* $\beta_T\, D'\, C\, R \wedge CR \equiv CE$ *et* $\beta_T\, P\, R\, Q \wedge RQ \equiv RP$

```
assert (exists P, Bet C' C P /\ Cong C P C D')...
DecompExAnd H21 P.
assert (exists R, Bet D' C R /\ Cong C R C E)...
DecompExAnd H21 R.
assert (exists Q, Bet P R Q /\ Cong R Q R P)...
DecompExAnd H21 Q.
```

Donc $FSC \begin{pmatrix} D'CRP \\ PCED' \end{pmatrix}$, *d'où* $RP \equiv ED'$ *et donc* $RQ \equiv ED$.

```
assert (FSC D' C R P P C E D').
unfold FSC;unfold Cong_3;intuition...
eapply l2_11.
apply H25.
3:apply H26.
Between.
sTarski.

assert (Cong R P E D').
eapply l4_16.
apply H21.
auto.

assert (Cong R Q E D).
eapply cong_transitivity.
apply H28.
eapply cong_transitivity.
apply H22.
sTarski.
```

On en déduit que $FSC \begin{pmatrix} D'EDC \\ PRQC \end{pmatrix}$,

```
assert (FSC D' E D C P R Q C).
unfold FSC;unfold Cong_3;intuition...
eapply l2_11.
```

```
3:eapply cong_commutativity.
3:eapply cong_symmetry.
3:apply H22.
Between.
Between.
sTarski.
```

d'où d'après le lemme 2.11 $D'D \equiv PQ$ *et* $CQ \equiv CD$ *(car dans le cas où* $D' \neq E$ *l'axiome des cinq segments permet de conclure, dans le cas contraire, nous pouvons déduire que* $D' = D$ *et* $P = Q$*).*

```
cases_equality D' E.
(* First case *)
treat_equalities...
sTarski.
(* Second case *)
eapply l4_16;eauto.
```

D'après le théorème 4.17, comme $R \neq C$ *et que* R, C *et* D' *sont colinéaires nous pouvons conclure que* $D'P \equiv D'Q$*.*

```
assert (R<>C).
unfold not;intro.
treat_equalities...

assert (Cong D' P D' Q).
apply l4_17 with (A:=R) (B:=C) (C:=D').
assumption.
3:apply H32.
unfold Col;left;Between.
sTarski.
```

Comme $C \neq D'$, $\mathrm{Col}\, CD'B$ *et* $\mathrm{Col}\, CD'B'$*. On peut déduire de manière analogue que* $BP \equiv BQ$ *et* $B'P \equiv B'Q$*.*

```
assert (Cong B P B Q).
eapply  l4_17; try apply H20;auto.
unfold Col;right;right;Between.
(* *)
assert (Cong B' P B' Q).
eapply  l4_17 with (C:=B').
apply H20.
unfold Col.
Between.
assumption.
assumption.
```

Comme $C \neq D'$, on a $B \neq B'$ et comme $Col\, BC'B'$ on a $C'P \equiv C'Q$.

```
cases_equality B B'.
subst B'.
unfold IFSC,FSC, Cong_3 in *;intuition.
clean_duplicated_hyps.
clean_trivial_hyps.
assert (Bet A B D').
Between.
assert (B=D').
eTarski.
treat_equalities.
Tarski.

assert (Cong C' P C' Q).
eapply l4_17.
apply H37.
unfold Col;right;left;Between.
auto.
auto.
```

Comme $C \neq C'$ et $Col\, C'CP$ on a $PP \equiv PQ$.

```
assert (Cong P P P Q).
eapply  l4_17.
apply H19.
unfold Col;right;right;Between.
auto.
auto.
```

D'après l'axiome d'identité on a $P = Q$.

```
assert (P=Q).
eapply cong_identity.
apply cong_symmetry.
apply H39.
```

Comme $PQ \equiv D'D$, on a aussi $D = D'$.

```
subst Q.
assert (D=D').
eapply cong_identity with (A:=D) (B:=D') (C:=P).
unfold IFSC,FSC, Cong_3 in *;intuition.
```

C'est ce qu'il fallait montrer.

```
assert (E=D).
eTarski.
unfold IFSC,FSC, Cong_3 in *;intuition.
```

□

2.5 A propos des cas dégénérés

Les différents articles à propos de la formalisation de la géométrie, en particulier ceux à propos de la formalisation des *Fondements de la géométrie de Hilbert* [DDS00, MF03], mettent l'accent sur le problème des cas dégénérés. Nous appelons cas dégénérés tous les cas particuliers de configuration de la figure comme par exemple les cas où l'une des conditions suivantes est vérifiée :

- deux points coïncident,
- trois points sont alignés,
- deux droites sont parallèles...

Les cas dégénérés n'apparaissent souvent pas dans les preuves informelles et alourdissent de façon conséquente la formalisation. La preuve d'un théorème de géométrie dans un cas dégénéré est souvent fastidieuse voire même difficile[10].

Afin de limiter la taille des preuves, nous nous sommes efforcés d'automatiser certaines tâches répétitives. Nous décrivons ici ces éléments d'automatisation. Nous pensons qu'un développement formel ne peut pas être réalisé de façon raisonnable sans concevoir quelques tactiques de démonstration, certes hautement spécialisées, mais qui permettent de traiter certains problèmes répétitifs qui sont souvent omis dans les preuves informelles. Ceci a le double interêt de raccourcir le temps de formalisation et d'augmenter la lisibilité des preuves générées. Il ne faut pas comparer ces mini-tactiques aux puissantes procédures de décision pour la géométrie, et nous ne pouvons pas les utiliser ici car un de nos buts à terme est justement de fournir des fondations pour des implantations de ces procédures de décision.

La tactique principale pour traiter les cas dégénérés est la tactique : `treat_equalities`. L'idée est de propager l'information à propos des cas dégénérés. Par exemple si l'on sait que $A = B$ et que $AB \equiv CD$ alors on peut déduire que $C = D$. C'est très simple mais cela raccourcit les preuves des cas dégénérés de manière assez efficace.

```
Ltac treat_equalities :=
try treat_equalities_aux;
repeat
  match goal with
   | H:(Cong ?X3 ?X3 ?X1 ?X2) |- _ =>
      assert (X1=X2);
      [apply cong_identity with (A:=X1) (B:=X2) (C:=X3);
       apply cong_symmetry;assumption|idtac];
      smart_subst X2
   | H:(Cong ?X1 ?X2 ?X3 ?X3) |- _ =>
```

[10] Il semblerait que les cas dégénérés soient à la géométrie ce que l'α-conversion est au lambda calcul : une source de difficulté importante dans le cadre d'une formalisation.

```
      assert (X1=X2);
      [apply cong_identity with (A:=X1) (B:=X2) (C:=X3);
       assumption|idtac];
      smart_subst X2
   | H:(Bet ?X1 ?X2 ?X1) |- _ =>
      assert (X1=X2);
      [apply between_identity;
       assumption|idtac];
      smart_subst X2
end.
```

Mais le problème des cas dégénérés ne peut pas être totalement évité car parfois les cas dégénérés sont presque plus complexes que le cas général. En revanche, notre expérience a montré que le choix de l'axiomatique est capital. Certains développements que nous avons réalisés dans le cadre de l'axiomatique de Hilbert, nous ont montré que celle-ci produit beaucoup plus de cas dégénérés que l'axiomatique de Tarski. Ceci est sans doute en partie dû au fait que les axiomes sont énoncés de manière moins générique que dans l'axiomatique de Tarski (ils comportent plus de conditions de non dégénéscence) et produisent donc des preuves moins « uniformes ».

2.6 Logique classique vs. logique intuitionniste.

Notre formalisation de la géométrie de Tarski a été réalisée dans le système Coq. Puisque le noyau du système Coq est basé sur le calcul des constructions inductives [CPM90], *i.e.* une logique constructive, quand nous avons besoin de la logique classique il est nécessaire de le préciser au système. C'est le cas dans le développement que nous avons réalisé à partir de l'axiomatique de Tarski. Il apparaît fréquemment dans les preuves des distinctions de cas. Celles-ci utilisent implicitement le fait que l'égalité entre deux points et le prédicat qui exprime le fait que trois points sont colinéaires sont décidables. Il serait bien sûr intéressant de réaliser un développement purement constructif de la géométrie, nous n'avons pas pu nous atteler à cette tâche même si comme nous l'avons vu il existe des axiomatiques constructives que l'on peut facilement formaliser en Coq [Kah95], seules quelques propriétés simples ont été dérivées par von Plato à partir de son axiomatique, il reste à réaliser un travail similaire à celui de Tarski, Szmielew et Schwabhäuser pour des géométries constructives, c'est à dire réaliser un développement systèmatique de la géométrie dans un cadre constructif.

2.7 Conclusion et perspectives

Nous avons présenté notre formalisation de l'axiomatique de Tarski. Nous donnons, en particulier, la preuve formelle que certaines simplifications de l'axiomatique originale de Tarski sont correctes. Mais la conclusion la plus importante est sans doute la suivante : parmi les différentes axiomatiques, celle de Tarski est vraisemblablement la plus adaptée à la formalisation. En effet, nous nous sommes aussi intéressé à la formalisation de l'axiomatique de Hilbert et d'après notre expérience personnelle, les développements réalisés à partir de l'axiomatique de Tarski sont plus faciles car ils sont plus uniformes que ceux réalisés à partir de l'axiomatique de Hilbert. Ils comportent moins de cas dégénérés. Comme de plus l'axiomatique de Tarski admet de bonnes propriétés méta-théoriques, nous pensons donc qu'elle est la plus adaptée à un développement formel de la géométrie (non constructive). Par exemple, contrairement à l'axiomatique de Hilbert, dans le cadre de l'axiomatique de Tarski, il est possible de construire une partie du développement de manière indépendante de la dimension dans laquelle on travaille. Cela signifie qu'en pratique, si nous voulons réaliser un développement en dimension trois, nous pouvons réutiliser une grande partie de notre développement.

Même si la géométrie ne fait plus beaucoup l'objet de l'intérêt des mathématiciens, nous pensons qu'il reste encore de nombreux sujets à explorer. Une extension naturelle de notre travail serait de formaliser les autres chapitres de [SST83] et de prouver tous les axiomes de Hilbert. Ce travail est en cours. Nous envisageons aussi d'enrichir notre formalisation afin de s'en servir comme fondation pour le développement de Frédérique Guilhot. Il serait aussi très intéressant (mais cela constitue un défi à plus long terme) de développer un traité de géométrie constructive similaire au livre de Schwabhäuser, Szmielew et Tarski mais basé sur l'axiomatique de von Plato [vP95] par exemple.

Détail des énoncés prouvés

L'annexe B fournit la liste des énoncés qui ont été prouvés en Coq à partir de notre formalisation de l'axiomatique de Tarski. Le développement Coq complet avec les *preuves* et les liens hypertextes facilitant la navigation peut être consulté à l'adresse suivante :

`http://www.lix.polytechnique.fr/Labo/Julien.Narboux/tarski.html`

Deuxième partie

Automatisation

Chapitre 3

Démonstration automatique en géométrie

3.1 Introduction

La géométrie est l'un des domaines les plus fructueux de la démonstration automatique. De nombreux théorèmes difficiles peuvent être prouvés par des programmes en utilisant des méthodes analytiques ou algébriques. Une des premières procédures de décision pour la géométrie fût introduite par Alfred Tarski : l'élimination des quantificateurs dans la théorie des corps réels clos. Sa méthode fût ensuite améliorée par la méthode de décomposition cylindrique de Collins [Col75]. Il existe de nombreuses techniques de démonstration automatique en géométrie. Dans ce chapitre nous donnons d'abord un bref aperçu des principales méthodes de démonstration automatique en géométrie, puis nous détaillons la méthode des aires de Chou, Gao et Zhang que nous avons formalisée au sein de l'assistant de preuve Coq. Cette formalisation fait l'objet du chapitre suivant.

Comme nous l'avons vu au premier chapitre, il y a deux façons d'aborder la géométrie : avec ou sans coordonnées. Ceci se reflète aussi dans les méthodes de démonstration automatique. Ainsi, nous distinguons deux types de méthodes, d'une part les méthodes algébriques qui sont basées sur une représentation sous forme de contraintes sur les coordonnées et d'autre part les méthodes qui évitent l'utilisation des coordonnées.

3.2 Les méthodes algébriques

3.2.1 La méthode des bases de Gröbner

La méthode des bases de Gröbner a été introduite par Bruno Buchberger en 1965[1] [Buc65]. Depuis, elle a beaucoup été étudiée et est disponible dans la plupart des systèmes de calcul formel. Elle a trouvé de nombreuses applications dans différents domaines des mathématiques et des sciences de l'ingénieur. En ce qui concerne les assistants de preuve, Jérôme Creci a réalisé une implantation dans le système Coq [Cre04] en réutilisant l'algorithme de Buchberger extrait de la preuve de correction réalisée par Laurent Théry [Thé01]. La méthode des bases de Gröbner permet de décider si un polynôme appartient à un idéal. Dans le contexte de la géométrie, il faut donc exprimer les hypothèses et la conclusion avec des polynômes et tester si la conclusion est dans l'idéal engendré par les hypothèses. Cette méthode ne permet pas de traiter des problèmes utilisant des inégalités polynomiales, on ne peut donc pas traiter des problèmes dans une géométrie ordonnée impliquant l'ordre des points sur une droite par exemple.

3.2.2 La méthode de Wu

Introduite en 1978, la méthode de Wu fût la première méthode vraiment fructueuse en géométrie [Wu78], puisque elle permit non seulement de retrouver la preuve de nombreux théorèmes exprimés de manière constructive en géométrie non ordonnée mais aussi de prouver de *nouveaux* théorèmes. Cette méthode est capable de générer automatiquement les conditions de non-dégénérescence d'un théorème. Elle est complète pour la géométrie métrique définie par l'axiomatique de Wu que nous avons décrite page 21. Cette méthode a été améliorée par Shang Ching Chou. Pour plus d'information à propos de cette méthode voir le livre et la thèse de Chou [Cho88, Cho85].

3.2.3 La décomposition cylindrique

La méthode de décomposition cylindrique a été introduite par Collins dans les années 70 [Col75]. C'est une méthode qui possède un champs d'application très large puisqu'elle réalise l'élimination des quantificateurs dans la théorie des corps réels clos. En ce qui concerne la preuve formelle, une implantation est en cours dans le système Coq, pour plus d'informations voir la thèse de Assia Mahboubi [Mah05, Mah06].

[1] Bruno Buchberger a donné le nom de son directeur de thèse à sa méthode.

3.3 Les méthodes sans coordonnées

3.3.1 La méthode des angles

La méthode des angles [CGZ96] permet de générer des preuves lisibles pour de nombreux théorèmes. Elle a cependant le défaut de n'être qu'une heuristique. Elle est basée sur la notion d'angle entre deux droites pris dans le sens trigonométrique. Cette méthode a fait l'objet d'une implantation en prolog par Sean Wilson [WF05]. Le logiciel développé permet en outre de visualiser les preuves sous la forme d'une suite de diagrammes.

3.3.2 La méthode des vecteurs

La méthode des vecteurs [CGZ94] est très similaire à la méthode des aires que nous développons dans la partie suivante. Elle manipule trois quantités géométriques : les vecteurs, leur produit vectoriel et leur produit scalaire. L'idée de la méthode est d'éliminer les points du but dans l'ordre inverse de leur construction.

3.3.3 La méthodes des aires de Chou, Gao et Zhang

La procédure de décision de Chou, Gao et Zhang est la mécanisation d'une méthode connue sous le nom de *méthode des aires*. C'est une méthode à mi-chemin entre les méthodes analytiques et algébriques. L'idée de la méthode consiste à exprimer le but de manière constructive[2] et à traiter les points dans l'ordre inverse de leur construction. Le tableau 3.1 fournit la liste des constructions possibles. Le traitement de chaque point consiste à éliminer toutes les occurrences de ce point dans le but. Ceci est réalisé grâce à une batterie de *lemmes d'élimination*.

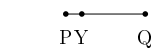
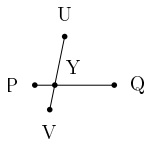
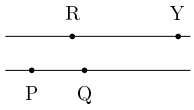
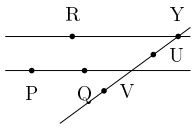
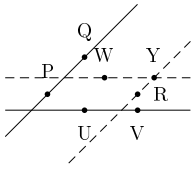
Pour être traitable par la méthode le but doit donc vérifier deux conditions. Premièrement le théorème doit être énoncé comme une séquence de constructions (par exemple on construit des points comme intersection de deux droites ou sur la parallèle à une droite passant par un point, etc.)[3]. Deuxièmement, le but doit être exprimé en utilisant une expression arithmétique comportant trois quantités géométriques :

1. la ratio de distances orientées portées par des droites parallèles ($\frac{\overline{AB}}{\overline{CD}}$) avec (AB) parallèle à (CD) ,
2. l'aire orientée d'un triangle ($\mathcal{S}_{ABC}$) et

[2]Notons que le terme « constructif » est employé ici pour désigner une séquence de constructions géométriques, il ne signifie pas constructif dans le sens de la logique constructive.

[3]Notons que la façon dont la figure est construite à une influence sur les conditions de non-dégénérescence et induit donc des théorèmes légèrement différents.

TAB. 3.1 – Constructions

	Soit Y sur la droite PQ tel que $\frac{\overline{PY}}{\overline{PQ}} = \lambda$. $(P \neq Q)$
	Soit Y l'intersection de PQ et UV. $(PQ \nparallel UV)$
	Soit Y sur la parallèle à PQ passant par R tel que $\frac{\overline{RY}}{\overline{PQ}} = \lambda$. $(P \neq Q)$
	Soit Y l'intersection de UV et de la parallèle à PQ passant par R. $(PQ \nparallel UV)$
	Soit Y l'intersection de la parallèle à PQ passant par R et de la parallèle à UV passant par W. $(PQ \nparallel UV)$

3. la différence de Pythagore (la différence entre la somme des carrés des deux côtés d'un triangle et du carré du troisième côté $\mathcal{P}_{ABC} = \overline{AB}^2 + \overline{BC}^2 - \overline{AC}^2$).

Remarque 2. *Notons que*

$$\mathcal{P}_{ABC} = -2(\overrightarrow{AB}.\overrightarrow{BC})$$

et

$$4\mathcal{S}^2_{ABC} = \overrightarrow{AB} \wedge \overrightarrow{BC} \; . \; \overrightarrow{AB} \wedge \overrightarrow{BC}$$

où . représente le produit scalaire et $\wedge$ représente le produit vectoriel. Ceci explique les similitudes évoquées plus haut entre la méthode des vecteurs et la méthode des aires.

Ces trois quantités géométriques sont suffisantes pour traiter un grande partie de la géométrie comme on peut le voir sur le tableau 3.2 page 61. Elles vérifient des propriétés élémentaires comme $\mathcal{S}_{AAB} = 0$, $\mathcal{S}_{ABC} = -\mathcal{S}_{BAC}$ et $\mathcal{S}_{ABC} = \mathcal{S}_{BCA}$. Ces propriétés sont définies précisément par l'axiomatique que nous avons présentée page 24.

En résumé, les formules traitées par cette méthode sont celles de la forme :

$$\forall A_1, \ldots A_n : \mathit{Point}, C_i(A_o, \ldots, A_p) \rightarrow \ldots \rightarrow C_j(A_q, \ldots, A_r) \rightarrow R = 0$$

où R est une expression arithmétique sur un corps contenant des aires orientées et des ratios et les C_i sont des prédicats exprimant la séquence de constructions. Pour chaque point construit il y a un prédicat C_i exprimant la façon dont il a été construit. Notons que le graphe de dépendance des constructions ne doit pas contenir de cycle.

Pour éliminer un point du but, on va appliquer l'un des lemmes d'élimination qui apparaissent dans le tableau 3.3 page 62. Ce tableau se lit de la façon suivante : pour éliminer un point Y, il faut choisir la ligne correspondant à la façon dont le point Y a été construit, et appliquer la formule donnée dans la colonne correspondant à la quantité géométrique dans laquelle Y apparaît. Les lemmes réécrivent n'importe quelle quantité géométrique contenant une occurrence d'un point Y ($\mathcal{S}_{ABY}$ ou $\frac{\overline{AY}}{\overline{CD}}$ pour n'importe quels points A,B,C et D tels que $(AY) \parallel (CD)$) en une expression sans occurrence de Y[4]. Il y a un lemme pour chaque paire de construction et de quantité géométrique. Pour la géométrie de l'incidence, nous avons cinq façons de construire un point et deux quantités géométriques, donc dix lemmes d'élimination sont nécessaires. Notons qu'il y a plus de constructions que nécessaire (certaines peuvent être exprimées au moyen d'autres). Ceci est utilisé pour simplifier

[4]Notons que toutes les occurrences de Y sont éliminées seulement si les points présents dans la quantité géométrique contenant Y (A,B,C et D) sont différents de Y, ce problème est traité dans l'implantation décrite au chapitre suivant.

les énoncés des théorèmes et pour réduire la taille des preuves en fournissant des lemmes d'élimination pour des constructions non primitives. Les constructions qui font intervenir un paramètre λ peuvent être utilisées pour construire un point à une distance donnée (si λ est instanciée) ou à n'importe quelle distance (si λ reste sous forme d'une variable). Dans ce cas, ces constructions permettent de construire ce que nous appelons des points semi-libres.
Quand tous les points construits ont disparu du but, le résultat est une expression arithmétique contenant des quantités géométriques utilisant uniquement des points libres[5]. Après cette étape les quantités géométriques utilisent uniquement des points libres mais ce ne sont pas nécessairement des quantités indépendantes. Elles peuvent être liées par la relation définie par l'axiome de dimension :

$$\mathcal{S}_{ABC} = \mathcal{S}_{DBC} + \mathcal{S}_{ADC} + \mathcal{S}_{ABD}$$

Dans le cas où elles ne sont pas indépendantes, elles peuvent être décomposées au moyen de trois points non colinéaires. Ces points peuvent être vus comme une base qui ne serait pas nécessairement orthogonale. Nous détaillerons ceci dans le chapitre 4. Si toutes les quantités géométriques sont indépendantes le but peut être vu comme une équation entre deux fractions rationnelles. Celle-ci est facilement décidable.
Les étapes de la méthode peuvent être résumées informellement de la façon suivante :

- exprimer le but de manière *constructive* (comme une suite de constructions) en n'utilisant que des quantitées géométriques ;
- vérifier les conditions de non-dégénérescence ;
- enlever les points construits du but un à un en utilisant les lemmes d'*élimination* ;
- transformer le but en une expression contenant uniquement des quantités géométriques *indépendantes* ;
- *décider* si l'égalité obtenue est universellement vraie ou non.

[5]Un point libre est un point dont le dégré de liberté est égal à la dimension de l'espace dans lequel on travaille, dans *GeoProof* cela correspond aux points que l'on peut déplacer librement.

TAB. 3.2 – Traduction de quelques prédicats usuels en utilisant $\mathcal{S}$, des ratios et $\mathcal{P}$

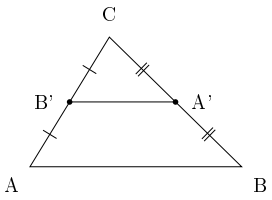
Notion Geométrique	**Formalisation**
A,B et C sont collinéaires	$\mathcal{S}_{ABC} = 0$
$AB \parallel CD$	$\mathcal{S}_{ABC} = \mathcal{S}_{ABD}$
I est le milieu de $[AB]$	$\frac{\overline{AB}}{\overline{AI}} = 2 \wedge \mathcal{S}_{ABI} = 0$
$AB \perp BC$	$\mathcal{P}_{ABC} = 0$
$AB \perp CD$	$\mathcal{P}_{ACD} = \mathcal{P}_{BCD}$
$A = B$	$\mathcal{P}_{ABA} = 0$

Exemple explicatif

Dans cette partie, nous donnons un exemple de preuve en utilisant la méthode des aires.

Considérons le théorème de la droite des milieux :

Soit ABC un triangle, et soient A' et B' les milieux de BC et AC respectivement. La droite $A'B'$ est parallèle à la base AB.
Preuve (en utilisant la méthode des aires). Tout d'abord nous traduisons le but $(A'B' \parallel AB)$ en son équivalent en utilisant l'aire orientée :



$$\mathcal{S}_{A'B'A} = \mathcal{S}_{A'B'B}$$

Ensuite nous éliminons du but les points construits en commençant par les derniers construits. Nous pouvons donc éliminer A' ou B' qui sont chacun des feuilles de l'arbre de dépendances de la construction. Nous choisissons d'éliminer B'. Les quantités géométriques contenant une occurrence de B' sont $\mathcal{S}_{A'B'B}$ et $\mathcal{S}_{A'B'A}$, B' a été construit en utilisant la première construction du tableau 3.3 avec $\lambda = \frac{1}{2}$, on en déduit que :

$$\mathcal{S}_{A'B'A} = \mathcal{S}_{AA'B'} = \frac{1}{2}\mathcal{S}_{AA'A} + \frac{1}{2}\mathcal{S}_{AA'C} = \frac{1}{2}\mathcal{S}_{AA'C}$$

et

$$\mathcal{S}_{A'B'B} = \mathcal{S}_{BA'B'} = \frac{1}{2}\mathcal{S}_{BA'A} + \frac{1}{2}\mathcal{S}_{BA'C}$$

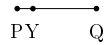
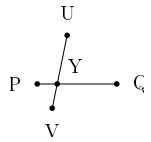
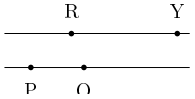
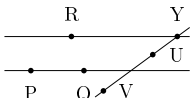
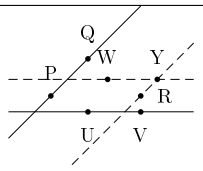
En utilisant ces deux égalités et après simplification le but devient alors

$$\mathcal{S}_{AA'C} = \mathcal{S}_{BA'A} + \mathcal{S}_{BA'C}$$

Maintenant nous pouvons éliminer A' de manière analogue en utilisant les égalités suivantes :

$$\mathcal{S}_{CAA'} = \frac{1}{2}\mathcal{S}_{CAB} + \frac{1}{2}\mathcal{S}_{CAC} = \frac{1}{2}\mathcal{S}_{CAB}$$

TAB. 3.3 – Lemmes d'élimination

Construction	Description	Formule d'élimination	
	(Condition de non dégénérescence)	$\mathcal{S}_{ABY} =$	Si $AY \parallel CD$, $A \neq Y$, $C \neq D$ alors $\frac{\overline{AY}}{\overline{CD}} =$
	Soit Y sur la droite PQ tel que $\frac{\overline{PY}}{\overline{PQ}} = \lambda$. $(P \neq Q)$	$\lambda S_{ABQ} + (1-\lambda)\mathcal{S}_{ABP}$	$\begin{cases} \frac{\overline{AP}}{\overline{PQ}} + \lambda & \text{si } A \in PQ \\ \frac{\overline{CD}}{\overline{PQ}} & \\ \frac{\mathcal{S}_{APQ}}{\mathcal{S}_{CPDQ}} & \text{sinon}^a. \end{cases}$
	Soit Y l'intersection de PQ et UV. $(PQ \nparallel UV)$	$\frac{\mathcal{S}_{PUV}\mathcal{S}_{ABQ} + \mathcal{S}_{QVU}\mathcal{S}_{ABP}}{\mathcal{S}_{PUQV}}$	$\begin{cases} \frac{\mathcal{S}_{AUV}}{\mathcal{S}_{CUDV}} & \text{si } A \notin UV \\ \frac{\mathcal{S}_{APQ}}{\mathcal{S}_{CPDQ}} & \text{sinon.} \end{cases}$
	Soit Y sur la parallèle à PQ passant par R tel que $\frac{\overline{RY}}{\overline{PQ}} = \lambda$. $(P \neq Q)$	$\mathcal{S}_{ABR} + \lambda\mathcal{S}_{APBQ}$	$\begin{cases} \frac{\overline{AR}}{\overline{PQ}} + \lambda & \text{si } A \in RY \\ \frac{\overline{CD}}{\overline{PQ}} & \\ \frac{\mathcal{S}_{APRQ}}{\mathcal{S}_{CPDQ}} & \text{sinon.} \end{cases}$
	Soit Y l'intersection de UV et de la parallèle à PQ passant par R. $(PQ \nparallel UV)$	$\frac{\mathcal{S}_{PUQR}\mathcal{S}_{ABV} - \mathcal{S}_{PVQR}\mathcal{S}_{ABU}}{\mathcal{S}_{PUQV}}$	$\begin{cases} \frac{\mathcal{S}_{AUV}}{\mathcal{S}_{CUDV}} & \text{si } A \notin UV \\ \frac{\mathcal{S}_{APRQ}}{\mathcal{S}_{CPDQ}} & \text{sinon.} \end{cases}$
	Soit Y l'intersection de la parallèle à PQ passant par R et de la parallèle à UV passant par W. $(PQ \nparallel UV)$	$\frac{\mathcal{S}_{PWQR}}{\mathcal{S}_{PUQV}}.\mathcal{S}_{AUBV} + \mathcal{S}_{ABW}$	$\begin{cases} \frac{\mathcal{S}_{APRQ}}{\mathcal{S}_{CPDQ}} & \text{si } AY \nparallel PQ \\ \frac{\mathcal{S}_{AUWV}}{\mathcal{S}_{CUDV}} & \text{sinon.} \end{cases}$

[a] $\mathcal{S}_{ABCD}$ est une notation pour $\mathcal{S}_{ABC} + \mathcal{S}_{ACD}$.

$$\mathcal{S}_{ABA'} = \frac{1}{2}\mathcal{S}_{ABB} + \frac{1}{2}\mathcal{S}_{ABC} = \frac{1}{2}\mathcal{S}_{ABC}$$

$$\mathcal{S}_{CBA'} = \frac{1}{2}\mathcal{S}_{CBB} + \frac{1}{2}\mathcal{S}_{CBC} = 0$$

Le but devient alors :

$$\frac{1}{2}\mathcal{S}_{CAB} = \frac{1}{2}\mathcal{S}_{ABC}$$

La preuve est terminée car $\mathcal{S}_{CAB} = \mathcal{S}_{ABC}$ par hypothèse sur $\mathcal{S}$.

3.4 Conclusion

Nous avons donné un rapide aperçu des principales méthodes de démonstration automatique en géométrie. Le tableau suivant résume les principales propriétés de ces méthodes. Nous avons décrit en détail la méthode des aires de Chou, Gao et Zhang. Dans le chapitre suivant, nous nous intéressons à sa formalisation en Coq.

Méthode	Traite des cas non constructifs	Traite des problèmes impliquant l'ordre	Génère des preuves lisibles	Référence
Bases de Gröbner	✓	✘	✘	[Buc65]
Décomposition cylindrique	✓	✓	✘	[Col75]
Méthode de Wu	✘	✘	✘	[Wu78]
Méthode des aires	✘	✘	✓	[CGZ94]
Méthode des angles	✘	✘	✓	[CGZ94]
Méthode des vecteurs	✘	✘	✓	[CGZ94]

Chapitre 4

Formalisation de la méthode des aires en Coq

Nous présentons dans ce chapitre la formalisation en Coq d'une procédure de décision pour la géométrie affine plane. Parmi les procédures de décision pour la géométrie que nous avons décrites au chapitre précédent, nous avons choisi d'implanter la méthode des aires de Chou, Gao et Zhang. En effet cette procédure génère des preuves à la fois courtes et « lisibles ».

4.1 Introduction

Comme nous avons pu le voir en introduction du chapitre 2, récemment des développements ont été réalisés pour formaliser des éléments de géométrie dans des assistants de preuve [Kah95, DDS00, MF03, Gui05, Nar06c]. Ces différents développements visent deux applications principales : l'enseignement de la géométrie et la preuve d'algorithmes géométriques. Nous pensons que les succès de la démonstration automatique en géométrie peuvent servir ces deux buts. En effet, il est possible de formaliser dans les assistants de preuve un éventail très large de théorèmes, mais pour cela nous avons besoin d'automatisation afin de faciliter la production de preuves formelles.

Le but du travail que nous décrivons dans ce chapitre est de permettre l'utilisation de la méthode de Chou, Gao et Zhang dans l'assistant de preuve Coq [Coq04, HKPM04, BC04a]. Nous décrivons ici l'implantation de la procédure de décision en une tactique Coq.

Comparée à une implantation *ad-hoc* d'un démonstrateur automatique en géométrie, cette approche basée sur un assistant de preuve a l'avantage de fournir un très haut niveau de confiance en la correction des preuves générées puisqu'elles sont vérifiées par le noyau de Coq.

La formalisation d'une procédure de décision au sein de Coq n'a pas seulement l'avantage de simplifier la preuve des théorèmes et d'augmenter le

niveau de confiance, elle permet aussi de combiner des preuves géométriques avec des preuves arbitrairement complexes développées en utilisant toute la puissance de la logique sous jacente à l'assistant de preuve. Par exemple, des théorèmes dont la preuve utilise une induction sur le nombre de points d'un polygone peuvent être formalisés en utilisant le système Coq. Il est aussi possible d'utiliser des arguments géométriques pour prouver la correction d'un programme en algorithmique géométrique. Les problèmes liés au traitement des conditions de non dégénérescence sont cruciaux dans ce contexte ; ceci est aussi mis en valeur dans notre formalisation.

Ce chapitre est organisé de la manière suivante : après avoir motivé nos choix de formalisation, nous décrivons comment la procédure est implantée dans l'assistant de preuve Coq, puis nous donnons quelques exemples de théorèmes démontrés automatiquement.

Le processus de formalisation de la procédure de décision peut être divisé en trois étapes. Il faut d'abord choisir l'axiomatique, ensuite prouver les propositions nécessaires à la tactique et enfin écrire la tactique elle-même. Ces trois étapes sont décrites dans les sections suivantes les unes après les autres, mais pour faciliter le développement formel elles ont en fait été réalisées en parallèle. En effet, certaines des sous-tactiques de la tactique globale implantant la procédure de décision servent d'outils pour faciliter la preuve des propositions nécessaires.

4.2 Choix du langage

Il existe trois méthodes pour ajouter une tactique au système Coq :

– programmer directement en utilisant la langage dans lequel Coq est implanté : Ocaml,

– programmer en utilisant le langage de tactiques de Coq : $\mathcal{L}_{tac}$,

– programmer par réflexion en utilisant Coq comme un langage de programmation.

Notre tactique est implantée principalement en utilisant le langage $\mathcal{L}_{tac}$ qui est intégré au système Coq. Ce langage fournit des primitives pour décrire des tactiques Coq au sein de Coq lui-même (sans utiliser Ocaml). Mais certaines des sous-tactiques que nous utilisons sont implantées en utilisant la méthode de la réflexion. Cette méthode, introduite par Samuel Boutin en 1997 [Bou97], consiste à formaliser un sous ensemble du langage de Coq (ici celui des expressions arithmétiques construites à partir de quantités géométriques noté A dans la suite) au moyen d'un objet de Coq lui même (dans ce cas un type inductif représentant les expressions arithmétiques noté AF). Les calculs qui seraient effectués par une tactique classique (non réflexive) dans le métalangage (Ocaml ou $\mathcal{L}_{tac}$) sont ici réalisés en utilisant le langage Coq lui-même. La figure 4.1 fournit une représentation du processus de ré-

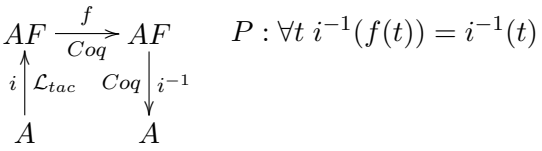


FIG. 4.1 – La réflexion.

flexion dans le cas d'une tactique qui réalise une réécriture. Une tactique réflexive est composée de quatre éléments :

i un fragment de code écrit en $\mathcal{L}_{tac}$ (ou en Ocaml)[1] pour traduire un terme Coq dans l'univers des objets (réification),

f un terme Coq qui résout le problème exprimé dans l'univers des objets,

i^{-1} un terme Coq qui traduit depuis l'univers des objets vers l'univers Coq (pour que la tactique puisse fonctionner il faut que $i^{-1}(i(t)) \longrightarrow t$ mais ce fait n'a pas besoin[2] d'être *prouvé formellement*),

P la preuve de la validité de la transformation réalisée par f.

Cette méthode a l'avantage de produire des tactiques plus efficaces et des preuves plus courtes parce que l'application de la tactique se traduit en une seule étape de calcul (en utilisant la règle de conversion du calcul des constructions inductives). De plus, grâce au travail de thèse de Benjamin Grégoire [Gré03] le processus de conversion peut être *compilé*[3], il est possible de réaliser de cette manière des calculs relativement efficaces. La preuve du théorème des quatre couleurs en Coq réalisée par Georges Gonthier en collaboration avec Benjamin Werner est un exemple significatif qui n'aurait sans doute pas pu être mené à bien sans cette technique [Gon04].

Pour plus d'information sur la méthode de preuve réflexive, nous invitons le lecteur à consulter le chapitre 16 du livre le *Coq'Art* de Yves Bertot et Pierre Castéran [BC04a].
Dans notre développement, nous avons utilisé le méthode de la réflexion pour l'implantation des tactiques d'unification et de simplification. Nous n'avons pas choisi la méthode de la réflexion pour la tactique dans sa globalité pour deux raisons :

1. Nous pensons que l'efficacité ne serait pas améliorée de manière significative et les preuves générées seraient d'une taille comparable. En effet les preuves générées par notre tactique sont principalement une suite d'applications de lemmes d'élimination.

[1] Il est parfois aussi possible d'utiliser la commande `quote` de Coq qui permet dans des cas simples de générer automatiquement ce fragment de code.

[2] Cela ne peut d'ailleurs pas être prouvé puisque i représente du code Ocaml ou $\mathcal{L}_{tac}$ que l'on ne peut pas manipuler dans Coq.

[3] Ceci nécessite la version 8.1 ou supérieure du système Coq.

2. Exprimer la tactique comme un terme Coq, et prouver la correction aurait été une tâche très difficile. Nous faisons un usage intensif des primitives de haut niveau du langage $\mathcal{L}_{tac}$ comme par exemple la reconnaissance de motifs pour des termes et des sous termes, l'effacement d'hypothèses *etc.* Pour utiliser la méthode de réflexion sur la tactique dans sa globalité, toute cette machinerie et la preuve de sa correction auraient dû être réalisées en Coq. Cela reviendrait en partie à implanter $\mathcal{L}_{tac}$ en Coq, exercice intéressant et qui pourrait s'avérer utile dans le contexte actuel qui consiste à mettre de plus en plus de calculs dans les preuves, mais qui aurait demandé un effort conséquent.

4.3 Choix de l'axiomatique

L'axiomatique que l'on va utiliser pour formaliser cette procédure de décision en Coq est inspirée de l'axiomatique de Chou, Gao et Zhang que nous avons présentée page 24. Le tableau 4.1 page ci-contre fournit la liste des axiomes que nous avons utilisés pour la formalisation.

Le premier axiome représente le fait que nous avons une collection de points. Nous supposons que nous avons un corps de caractéristique différente de deux. Nous omettons les axiomes des corps qui sont standards. Nous supposons que nous avons deux fonctions, l'une d'arité deux ($\overline{AB}$) et l'autre d'arité trois ($\mathcal{S}_{ABC}$) des points dans le corps (F). La première représente la distance orientée entre A et B, la seconde représente l'aire orientée du triangle A,B,C.
Les axiomes de dimension expriment le fait que tous les points sont dans le même plan et qu'il ne sont pas tous colinéaires.
L'axiome de construction exprime le fait que l'on peut construire un unique point à une certaine distance sur une droite définie par deux points distincts.
L'axiome des proportions est très important, il fournit une relation entre le ratio des distances orientées et l'aire orientée.

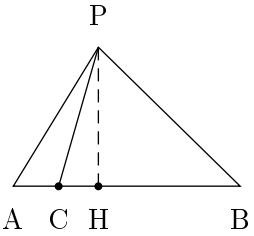
Notre axiomatique diffère légèrement de celle de Chou, Gao et Zhang.
Premièrement, nous supposons que la caractéristique du corps est différente de deux. Cela est utilisé d'une part pour simplifier l'axiomatique et d'autre part pour permettre la construction du milieu d'un segment sans avoir à faire l'hypothèse explicite que deux est différent de zéro. En effet si $2 \neq 0$ on a :

$$\forall ABC\ \mathcal{S}_{ABC} = -\mathcal{S}_{BAC} \Rightarrow \forall AC\ \mathcal{S}_{AAC} = 0$$

Avec l'axiome de Chasles on peut aussi déduire que $\overline{AB} = -\overline{BA}$.
Nous accordons de l'importance au milieu d'un segment car il est utile à la preuve de la validité de la construction d'un point sur une droite parallèle à une autre droite.
Deuxièmement nous n'admettons pas l'existence d'une notion de collinéarité, cette notion est définie en utilisant l'aire orientée (trois points sont colinéaires

TAB. 4.1 – L'axiomatique formalisée

Points	Point : Set
Corps	F est un corps $2 \neq 0$
Distance orientée	$\overline{\cdot} : \text{Point} \to \text{Point} \to F$ $\overline{AB} = 0 \iff A = B$
Aire orientée	$\mathcal{S} : \text{Point} \to \text{Point} \to \text{Point} \to F$ $\mathcal{S}_{ABC} = \mathcal{S}_{CAB}$ $\mathcal{S}_{ABC} = -\mathcal{S}_{BAC}$
Axiome de Chasles	$\mathcal{S}_{ABC} = 0 \to \overline{AB} + \overline{BC} = \overline{AC}$
Dimension	$\exists\, A, B, C : \text{Point}, \mathcal{S}_{ABC} \neq 0$ $\mathcal{S}_{ABC} = \mathcal{S}_{DBC} + \mathcal{S}_{ADC} + \mathcal{S}_{ABD}$
Construction	$\forall r : F\ \exists P : Point, \mathcal{S}_{ABP} = 0 \wedge \overline{AP} = r\overline{AB}$ $A \neq B \wedge \mathcal{S}_{ABP} = 0 \wedge \overline{AP} = r\overline{AB} \wedge \mathcal{S}_{ABP'} = 0 \wedge \overline{AP'} = r\overline{AB} \to P = P'$
Parallélogramme	$\mathcal{S}_{PCQ} + \mathcal{S}_{PQD} = 0 \Rightarrow C \neq D \Rightarrow \mathcal{S}_{CDQ} \neq 0 \Rightarrow$ $\frac{\overline{PQ}}{\overline{CD}} = 1 \Rightarrow \frac{\mathcal{S}_{PDQ}}{\mathcal{S}_{CDQ}} = 1$
Proportions	$\mathcal{S}_{PAC} \neq 0 \to \mathcal{S}_{ABC} = 0 \to \frac{\overline{AB}}{\overline{AC}} = \frac{\mathcal{S}_{PAB}}{\mathcal{S}_{PAC}}$



si l'aire du triangle qu'ils forment est nulle). Dans [CGZ94] la notion de colinéarité de trois points A,B et C est utilisée pour exprimer certains axiomes puis est prouvée équivalente à $\mathcal{S}_{ABC} = 0$.
Troisièmement, nous ajoutons un axiome concernant les ratios de distances orientées portées par des droites parallèles. L'axiome énonce le fait que si la droite (AB) est parallèle à la droite (CD) et $\overline{AB} = \overline{CD}$ alors la droite (BD) est parallèle à la droite (AC). Cet axiome est un cas particulier de la généralisation aux droites parallèles de l'axiome des proportions :

$$PQ \parallel AB \Rightarrow \mathcal{S}_{AQP} \neq 0 \Rightarrow \frac{\overline{AB}}{\overline{PQ}} = \frac{\mathcal{S}_{PAB}}{\mathcal{S}_{AQP}}$$

Nous pourrions aussi utiliser ce dernier.
Quatrièmement, nous pouvons diviser des distances arbitraires, tandis que l'axiomatique de Chou, Gao et Zhang se restreint aux ratios de distance orientée $\frac{\overline{AB}}{\overline{CD}}$ quand les droites (AB) et (CD) sont parallèles. La cohérence est préservée car la distance orientée peut être interprétée dans le modèle analytique standard. Le fait que nous puissions diviser des distances arbitraires signifie que pour interpréter la fonction distance nous devons orienter les droites du plan.
Mais la procédure de décision utilise explicitement le fait que pour chacun des ratios de distances orientées $\frac{\overline{AB}}{\overline{CD}}$, (AB) est parallèle à (CD). Nos lemmes utilisés dans la procédure contiennent l'hypothèse explicite que les droites sont parallèles. Cela signifie que dans notre formalisation on peut écrire un ratio de distances portées par des droites non parallèles mais la procédure de décision ne peut pas traiter ce cas. Cette formalisation est plus commode parce qu'elle est plus générale et permet d'utiliser l'opération de division du corps considéré, et donc de réutiliser les tactiques standards sur les corps fournies par le système Coq. Sinon nous aurions dû donner des axiomes concernant les ratios (voir page 24) et prouver des propriétés sur le lien entre le ratio de deux distances orientées et le produit, la somme, *etc.*

Ce choix de formalisation permet aussi de manipuler des ratios de distances supportées par des droites parallèles sans pour autant faire l'hypothèse *explicite* que ces droites sont parallèles. Ceci est parfois utile, par exemple, si on considère la construction utilisée pour le théorème de la droite des milieux, et que nous voulons prouver que $\frac{\overline{A'B'}}{\overline{AB}} = \frac{1}{2}$ nous ne voulons pas ajouter l'hypothèse que $A'B' \parallel AB$ parce que c'est une *conséquence* des autres hypothèses. Il résulte de ce choix que deux invariants doivent être maintenus tout au long de la preuve :

1. pour chaque dénominateur la preuve qu'il est différent de zéro doit apparaître dans le contexte
2. pour chaque ratio de distances orientées $\frac{\overline{AB}}{\overline{CD}}$ la preuve que la droite (AB) est parallèle à la droite (CD) doit apparaître dans le contexte

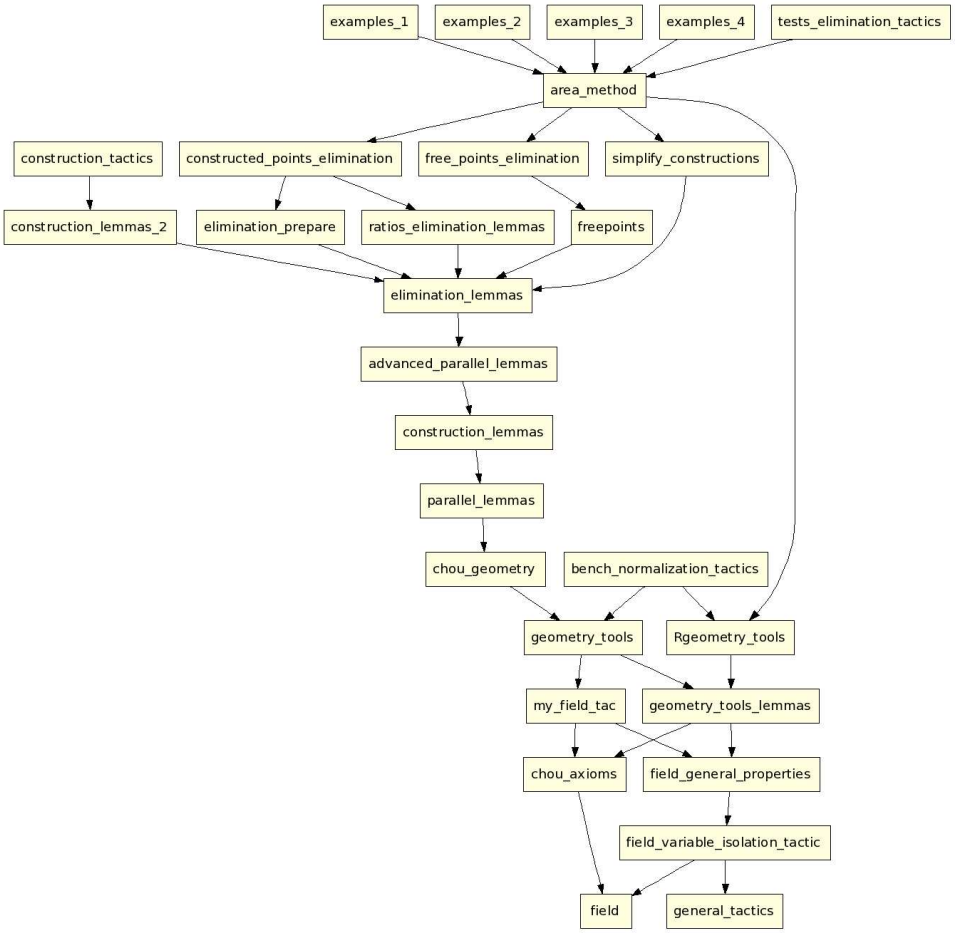


FIG. 4.2 – Graphe de dépendances entre les modules

A propos de la complétude La procédure de Chou, Gao et Zhang est complète vis à vis des énoncés constructifs dont le but est exprimé dans leur axiomatique. Pour les raisons pratiques que nous venons de donner, nous avons étendu cette axiomatique. Ce choix de formalisation implique que la procédure de décision n'est pas complète vis à vis de l'axiomatique que nous avons définie (les buts qui ne sont pas dans le langage de la tactique seront donc rejetés).

4.4 Propositions nécessaires à la tactique

Dans ce paragraphe, nous donnons un rapide aperçu des propositions qui ont été prouvées en Coq pour le développement de cette tactique :

Propositions de base

Les proposition de bases sont utilisées pour réécrire les quantités géométriques. Ce sont les propositions les plus basiques qui sont utilisées par les tactiques de simplification et d'unification. Ces propositions apparaissent tôt dans le développement afin de pouvoir utiliser les tactiques de simplification et d'unification aussi tôt que possible.

Lemmes de construction

Les lemmes de construction prouvent que les différentes constructions possibles sont correctes.

Lemmes d'élimination

Les lemmes d'élimination servent à réécrire une quantité géométrique en une expression qui ne contient pas le point que l'on doit éliminer. Les lemmes d'élimination sont associés à des lemmes pour maintenir les invariants de la tactique.

Lemmes pour le traitement des points libres

Les lemmes pour le traitement des point libres permettent d'exprimer des quantités géométriques par des expressions ne comportant que des quantités géométriques indépendantes.

4.5 La tactique proprement dite

Nous fournissons dans cette partie une description détaillée des sous-tactiques que nous utilisons.

4.5.1 Tactique d'initialisation

1. La tactique d'initialisation (appelée `geoinit`) vérifie que le but est compatible avec la procédure de décision. Ceci inclut la vérification que les invariants sont vrais avant le début de la procédure.
2. Elle déplie les définitions qui ne sont pas traitées directement par la procédure. Par exemple, le prédicat `midpoint` est exprimé comme un ratio de distances et un énoncé exprimant la collinéarité.
3. Elle introduit toutes les hypothèses dans le contexte.
4. Elle décompose certaines parties du but. Elle sépare les membres d'une conjonction et décompose certaines constructions.

Exemple. *Le théorème de la droite des milieux est énoncé en utilisant notre langage et la syntaxe Coq Version 8.0 ou supérieure de la façon suivante :*

```
Theorem midpoint_A :
 forall A B C A' B' : Point, midpoint A' B C -> midpoint B' A C ->
 parallel A' B' A B.
```

Voici le produit de la tactique `geoInit` *:*

```
1 subgoal
  A : Point
  B : Point
  C : Point
  A' : Point
  B' : Point
  H : on_line_d A' B C (1 / 2)
  H0 : on_line_d B' A C (1 / 2)
  ============================
   S A' A B' + S A' B' B = 0
```

`on_line_d A' B C (1/2)` signifie que A' appartient à la droite (BC) et que $\frac{\overline{BA'}}{\overline{BC}} = \frac{1}{2}$.

4.5.2 Tactique de simplification.

La tactique de simplification (`basic_simpl`) réalise des simplifications simples des hypothèses et du but. Notons que pour préserver nos invariants nous devons réaliser exactement les mêmes simplifications dans le but et dans les hypothèses. Par exemple, si le dénominateur d'une fraction est simplifié, la même simplification doit être réalisée dans l'hypothèse qui correspond à la preuve que le dénominateur est différent de zéro. Sinon nous perdons l'invariant qui consiste à avoir en permanence en hypothèse la non-nullité de toutes les expressions qui apparaissent *syntaxiquement* au dénominateur d'une des fractions.

Le processus de simplification consiste en l'application des règles de réécriture suivantes :

1. $\mathcal{S}_{AAB} \longrightarrow 0$
2. $\mathcal{S}_{ABB} \longrightarrow 0$
3. $\mathcal{S}_{BAB} \longrightarrow 0$
4. $\overline{AA} \longrightarrow 0$
5. $-(-x) \longrightarrow x$
6. $-0 \longrightarrow 0$
7. $0 * x \longrightarrow 0$
8. $x * 0 \longrightarrow 0$
9. $1 * x \longrightarrow x$
10. $x * 1 \longrightarrow x$

11. $x + 0 \longrightarrow x$
12. $0 + x \longrightarrow x$

Cette tactique est nécessaire afin de préserver la taille du but. Ne pas simplifier le but mène à des termes d'une très grande taille. Les exemples montrent que sans simplifications, les problèmes deviennent rapidement impraticables.

4.5.3 Tactiques d'unification.

Il y a autant de tactiques d'unification que de quantités géométriques. Les tactiques d'unification (`unify_signed_areas`,`unify_signed_distances`) changent le but et les hypothèses dans le but d'unifier les quantités géométriques : deux expressions syntaxiquement différentes qui dénotent la même quantité géométrique sont transformées en une seule expression. Par exemple si à la fois $\overline{AB}$ et $\overline{BA}$ apparaissent dans un but, $\overline{AB}$ est transformé en $-\overline{BA}$[4].
Cette tactique est utile pour deux raisons :

1. Elle peut accélérer certaines étapes, car toute réécriture d'une des quantités géométriques équivalentes entraîne la réécriture de l'autre puisque après unification elles sont identiques syntaxiquement.
2. Il est nécessaire que les quantités géométriques qui sont équivalentes aient la même forme. En effet, lors de la dernière étape de la procédure nous supposons que les quantités géométriques sont indépendantes, car nous utilisons la tactique standard de Coq pour prouver une égalité sur un corps : `field`, et cette tactique considérerait que $\overline{AB}$ et $\overline{BA}$ sont des variables différentes puisqu'elle ne sait rien à propos de la distance orientée.

Exemple. *Dans ce contexte :*

```
H9 : S C A B <> 0
H8 : S B A C <> 0
H1 : S A B C <> 0
============================
 S P B C / S A B C + S P A C / S B A C + S P A B / S C A B = 1
```

la tactique `unify_signed_areas` *transforme le but en :*

```
H8 : - S A B C <> 0
H1 : S A B C <> 0
============================
 S P B C / S A B C + S P A C / - S A B C + S P A B / S A B C = 1
```

[4]Le choix de réécrire $\overline{AB}$ ou $\overline{BA}$ est réalisé de façon arbitraire par la tactique.

4.5.4 Tactique d'élimination.

La tactique d'élimination (appelée `eliminate_all`) cherche dans le contexte un point qui n'est pas utilisé pour construire un autre point, en d'autres termes une feuille dans le graphe de dépendances. Ensuite pour chaque occurrence du point dans le but, la tactique applique le lemme correspondant pris dans le tableau 3.3 62. Afin de choisir le bon lemme, elle cherche dans le contexte comment le point a été construit et dans quelle quantité géométrique il apparaît. Enfin, quand toutes les occurrences du point ont été éliminées, elle efface du contexte l'hypothèse concernant la construction du point éliminé.
Notons que certains lemmes admettent une condition de bord, dans ce cas un appel recursif sur la tactique est réalisé. Si la condition est réalisée, le lemme est appliqué, sinon on effectue une étape de raisonnement classique par cas sur la condition de bord. La formalisation en Coq met en valeur cette étape de raisonnement classique. Comme nous avons pu le remarquer dans le chapitre précédent, les lemmes d'élimination donnés dans le tableau 3.3 page 62, éliminent vraiment l'occurrence du point Y uniquement si Y apparaît une seule fois dans la quantité géométrique (A,B,C et D doivent être différents de Y). Si Y apparaît deux fois dans $\mathcal{S}$, cela ne pose pas de problème car dans ce cas la quantité géométrique est dégénérée et donc éliminée par l'étape de simplification. Mais si Y apparait deux fois dans un ratio (comme dans l'expression $\frac{\overline{AY}}{\overline{BY}}$ par exemple) ceci constitue un cas particulier qui est traité à part dans l'implantation.

Exemple. *Dans le contexte suivant :*

```
1 subgoal
A : Point
B : Point
C : Point
A' : Point
B' : Point
H : on_line_d A' B C (1 / 2)
H0 : on_line_d B' A C (1 / 2)
  ============================
S A' A B' + S B A' B' = 0
```

le tactique `eliminate B'` *transforme le but en :*

```
1 subgoal
A : Point
B : Point
C : Point
A' : Point
B' : Point
```

```
H : on_line_d A' B C (1 / 2)
  ============================
1 / 2 * S A' A C + (1 - 1 / 2) * S A' A A +
(1 / 2 * S B A' C + (1 - 1 / 2) * S B A' A) = 0
```

4.5.5 Tactique d'élimination des points libres.

Cette tactique suppose que le but est une expression comportant des quantités géométriques définies uniquement avec des points libres (tous les points construits ont été éliminés par la tactique d'élimination). Le rôle de cette tactique est de transformer le but en une expression comportant uniquement trois variables indépendantes. En effet les quantités géométriques comportant des points libres ne sont pas nécessairement indépendantes, elles sont liées par les relations suivantes :

$$\mathcal{S}_{ABC} = \mathcal{S}_{DBC} + \mathcal{S}_{ADC} + \mathcal{S}_{ABD}$$

Mais les quantités géométriques ne contenant que des points libres peuvent être transformées en des variables indépendantes en les exprimant au moyen d'une base. Pour cela nous choisissons trois points non colinéaires O,U et V et nous utilisons le lemme suivant pour réécrire les quantités géométriques qui contiennent plus d'un point qui n'est pas l'un des points de la base (O,U et V) :

$$\mathcal{S}_{OUV} \neq 0 \rightarrow \mathcal{S}_{ABC} = \begin{vmatrix} \mathcal{S}_{OUA} & \mathcal{S}_{OVA} & 1 \\ \mathcal{S}_{OUB} & \mathcal{S}_{OVB} & 1 \\ \mathcal{S}_{OUC} & \mathcal{S}_{OVC} & 1 \end{vmatrix}$$

Si il y a trois points dans le contexte qui sont connus comme n'étant pas colinéaires, nous les utilisons comme base O, U, V. Sinon, nous construisons trois points non collineaires grâce à l'axiome de dimension.

4.5.6 Tactique conclusion.

Quand cette tactique est appelée, le but est une expression comportant uniquement des variables indépendantes. Si l'égalité est universellement vraie alors le théorème est vrai. Sinon il existe une fonction des variables vers le corps qui rend l'égalité fausse, et ceci fournit un contre exemple au théorème. Pour vérifier si l'égalité est universellement vrai ou non, nous utilisons la tactique Coq `field`. Cette tactique génère des sous-buts qui correspondent à la preuve de non nullité des dénominateurs. Ceux-ci sont prouvés en utilisant les hypothèses ou en utilisant le fait que $x \neq 0 \wedge y \neq 0 \rightarrow x * y \neq 0$. Cette tactique est assez courte pour être expliquée en détail :

```
Ltac field_and_conclude :=
    abstract(field; repeat (assumption || apply nonzeromult); geometry).
```

Cette tactique réalise un appel à la tactique `field`, et tente d'appliquer l'une des hypothèses. Si ce processus échoue, elle décompose le produit et

résout les sous-buts en utilisant la tactique `geometry`. Cette dernière tactique est capable de résoudre des buts comme $\overline{AB} \neq 0$ quand $A \neq B$ figure en hypothése. La tactique `abstract` est utilisée pour des raisons techniques : elle accélère le processus de typage en créant un lemme fantôme.

4.6 Un exemple détaillé

Dans cette partie nous donnons une description détaillée du fonctionnement de la tactique sur le premier exemple en la décomposant en petites étapes. Ces étapes ne sont pas nécessairement exactement les mêmes que celles executées par notre procédure automatique (il se peut que la procédure automatique traite les points dans un ordre différent et réalise plus d'étapes de simplification ou d'unification).

Exemple.

```
forall A B C A' B' : Point,
   midpoint A' B C ->
   midpoint B' A C ->
   parallel A' B' A B.
```

Ici il serait suffisant de taper `autogeom` *pour résoudre le but en utilisant notre procédure de décision, mais pour la clarté de la présentation nous mimons le comportement de la tactique en utilisant les sous-tactiques qui ont été décrites dans les parties précédentes. Nous donnons le nom de la sous-tactique utilisée suivi du résultat renvoyé par Coq :*

```
geoInit.

  H : on_line_d A' B C (1 / 2)
  H0 : on_line_d B' A C (1 / 2)
  ============================
   S A' A B' + S A' B' B = 0

eliminate B'.

  H : on_line_d A' B C (1 / 2)
  ============================
   1 / 2 * S A' A C + (1 - 1 / 2) * S A' A A +
   (1 / 2 * S B A' C + (1 - 1 / 2) * S B A' A) = 0

basic_simpl.

  H : on_line_d A' B C (1 / 2)
  ============================
   1 / 2 * S A' A C + (1 / 2 * S B A' C + 1 / 2 * S B A' A) = 0
```

```
eliminate A'.

  ============================
   1 / 2 * (1 / 2 * S A C C + (1 - 1 / 2) * S A C B) +
   (1 / 2 * (1 / 2 * S C B C + (1 - 1 / 2) * S C B B) +
    1 / 2 * (1 / 2 * S A B C + (1 - 1 / 2) * S A B B)) = 0

basic_simpl.

  ============================
   1 / 2 * (1 / 2 * S A C B) + 1 / 2 * (1 / 2 * S A B C) = 0

unify_signed_areas.

  ============================
   1 / 2 * (1 / 2 * S A C B) + 1 / 2 * (1 / 2 * - S A C B) = 0

field_and_conclude.

Proof completed.
```

4.7 Exemples

Dans cette partie nous listons quelqu'uns des thèorèmes traités par notre tactique.

4.7.1 Ceva

Théorème 1 (Ceva). *Soit ABC un triangle et P un point du plan.*
Soit D l'intersection des droites (AP) et (BC).
Soit E l'intersection des droites (BP) et (AC).
Soit F l'intersection des droites (CP) et (AB).
On a :

$$\frac{\overline{AF}}{\overline{FB}} + \frac{\overline{BD}}{\overline{DC}} + \frac{\overline{CE}}{\overline{EA}} = 1$$

Voici le théorème exprimé dans la syntaxe de Coq :

```
Theorem Ceva :
forall A B C D E F G P : Point,
inter_ll D B C A P ->
inter_ll E A C B P ->
inter_ll F A B C P ->
F <> B ->
D <> C ->
E <> A ->
parallel A F F B ->
parallel B D D C ->
parallel C E E A ->
(A ** F / F ** B) * (B ** D / D ** C) * (C ** E / E ** A) = 1.
```

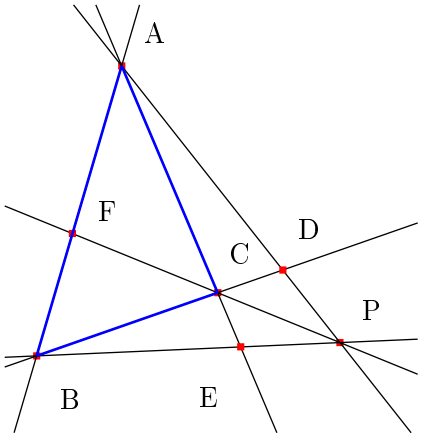


FIG. 4.3 – Le théorème de Céva

4.7.2 Menelaus

Théorème 2 (Menelaus)**.** *Soit ABC un triangle.*
Soit D sur la droite (BC).
Soit E sur la droite (AC).
Soit F l'intersection des droites (DE) et (AB).
Alors l'égalité suivante est vérifiée :

$$\frac{\overline{AF}}{\overline{FB}}.\frac{\overline{BD}}{\overline{DC}}.\frac{\overline{CE}}{\overline{EA}} = -1$$

```
Theorem Menelaus_2 :
 forall A B C X Y D E F : Point,
 inter_ll D B C X Y ->
 inter_ll E A C X Y ->
 inter_ll F A B X Y ->
 F <> B ->
 D <> C ->
 E <> A ->
 parallel A F F B ->
 parallel B D D C ->
 parallel C E E A ->
 (A**F / F**B) * (B**D / D**C) * (C**E / E**A) = - (1).
```

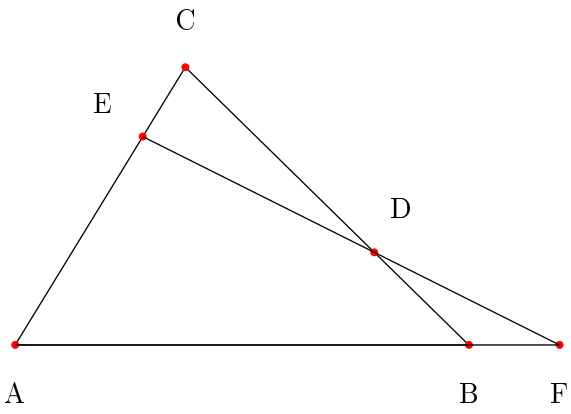


Fig. 4.4 – Le théorème de Ménélaus

4.7.3 Pascal

Théorème 3 (Pascal[5]). *Soient A, A' et B trois points.*
Soit C sur la droite (AB).
Soit B' sur la droite parallèle à (BA') passant par A.
Soit C' sur l'intersection de la droite $(A'B')$ et la droite parallèle à la droite (CA') passant par A.
Alors la droite (BC') est parallèle à la droite $(B'C)$.

```
Theorem Pascal :
 forall A B C A' B' C' : Point,
 on_line C A B ->
 on_parallel B' A B A' ->
 on_inter_line_parallel C' A A' B' C A' ->
 parallel B C' B' C.
```

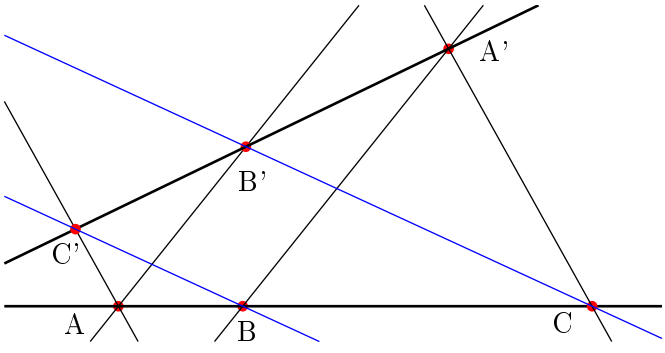


FIG. 4.5 – Le théorème de Pascal

[5]Ce théorème est parfois appelé version affine du théorème de Pappus.

4.7.4 Pappus

Théorème 4 (Pappus). *Soient ABC et $A'B'C'$ deux triplets de points alignés.*
Soit P l'intersection des droites $(A'B)$ et (AB').
Soit Q l'intersection des droites (AC') et $(A'C)$.
Soit R l'intersection des droites $(B'C)$ et (BC').
Les points P, Q et R sont alignés.

Constructivement, ce théorème peut s'exprimer de la façon suivante :

```
Theorem Pappus : forall A B C A' B' C' P Q R :Point,
  on_line C A B ->
  on_line C' A' B' ->
  inter_ll P A' B A B' ->
  inter_ll Q A C' A' C ->
  inter_ll R B' C B C' ->
  Col P Q R.
```

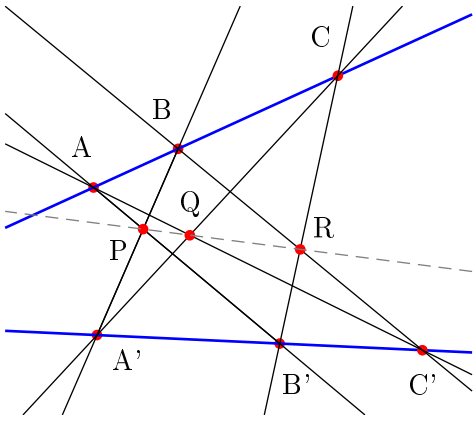


FIG. 4.6 – Le théorème de Pappus

4.7.5 Desargues

Théorème 5 (Desargues). *Soient XAA', XBB' et XCC' trois triplets de points alignés. Si $(AB) \parallel (A'B')$ et $(AC) \parallel (A'C')$ alors $(BC) \parallel (B'C')$.*

Constructivement, ce théorème peut s'exprimer de la façon suivante :

```
Theorem Desargues :
 forall A B C X A' B' C' : Point,
 on_line A' X A ->
 on_inter_line_parallel B' A' X B A B ->
 on_inter_line_parallel C' A' X C A C ->
 parallel B C B' C'.
```

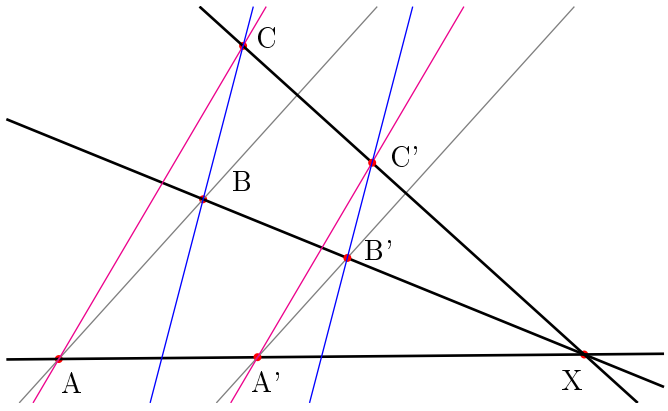


FIG. 4.7 – Le théorème de Desargues

4.7.6 Centre de gravité

Théorème 6 (Centre de gravité).
Soit ABC un triangle,
Soit F le milieu du segment $[AB]$
Soit E le milieu du segment $[BC]$
Soit O l'intersection des droites (AE) et (CF).
La distance $\overline{AO}$ vaut le double de la distance $\overline{OE}$.

```
Theorem Centroid :
 forall A B C E F O : Point,
 is_midpoint F A B ->
 is_midpoint E B C ->
 inter_ll O A E C F ->
 O <> E ->
 parallel A O O E ->
 A ** O / O ** E = 2.
```

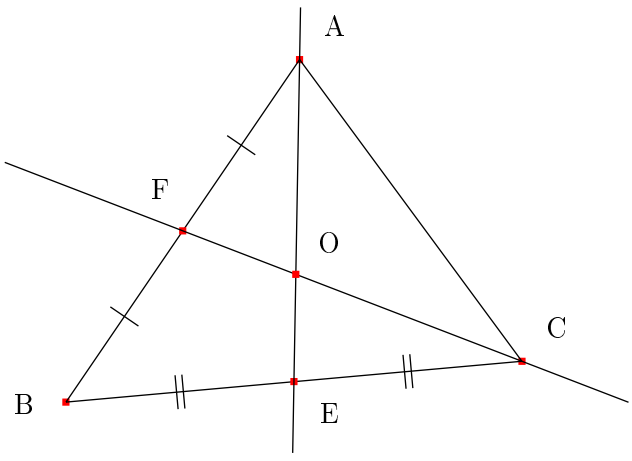


FIG. 4.8 – Le théorème du centre de gravité

4.7.7 Droite de Gauss

Théorème 7 (Droite de Gauss). *Soient A_0,A_1,A_2 et A_3 quatre points.*
Soit X l'intersection des droites (A_1A_2) et (A_0A_3).
Soit Y l'intersection des droites (A_0A_1) et (A_2A_3).
Soient M_1,M_2 et M_3 les milieux des segments $[A_1A_3]$, $[A_0A_2]$ et $[XY]$ respectivement.
Alors M_1, M_2 et M_3 sont alignés.

```
Theorem gauss_line :
 forall A0 A1 A2 A3 X Y M1 M2 M3 : Point,
 inter_ll X A0 A3 A1 A2 ->
 inter_ll Y A2 A3 A1 A0 ->
 is_midpoint M1 A1 A3 ->
 is_midpoint M2 A0 A2 ->
 is_midpoint M3 X Y ->
 S A0 A1 A2 <> 0 ->
 Col M1 M2 M3.
```

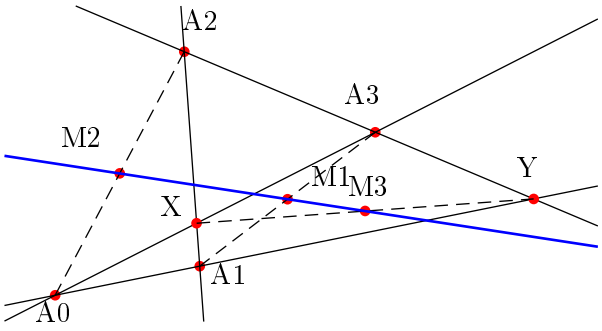


FIG. 4.9 – Le théorème de la droite de Gauss

4.8 Conclusion

Nous avons formalisé la méthode des aires pour la géométrie affine plane dans l'assistant de preuve Coq. Notre implantation utilise à la fois le langage de tactiques de Coq et Coq lui-même comme des langages de programmation. Notre travail montre qu'il est possible de marier la déduction automatique avec la preuve interactive. Nous enviseagons d'utiliser l'implantation réalisée dans le cadre de la preuve de programmes ou de la pédagogie (voir page 149). Il serait intéressant de formaliser d'autres méthodes de démonstration automatique en géométrie. La méthode des vecteurs serait une extension simple de notre implantation. Nous envisageons aussi de formaliser la méthode de Wu.

Troisième partie

Visualisation

Chapitre 5

GeoProof, géométrie dynamique et preuve formelle

Nous présentons dans ce chapitre la conception d'une interface graphique orientée vers la preuve formelle en géométrie. Le logiciel que nous avons développé combine trois éléments :

– un logiciel de géométrie dynamique pour construire des figures, les explorer, effectuer des mesures et inventer des conjectures,

– un module de démonstration automatique pour vérifier des faits et

– un système de preuve interactive (Coq) pour vérifier les preuves réalisées par l'utilisateur.

5.1 Introduction

Les logiciels de géométrie dynamique [1] et les systèmes de calcul formel comme Maple ou Mupad [2] sont les logiciels les plus utilisés dans le cadre de l'enseignement des mathématiques.

Les logiciels de géométrie dynamique permettent à l'utilisateur de créer des figures géométriques complexes pas à pas en se basant sur des objets libres (c'est à dire les objets qui peuvent être déplacés librement) et sur des constructions prédéfinies qui dépendent d'autres objets (par exemple la droite passant par deux points, le milieu d'un segment, ...).

Les objets libres peuvent être déplacés au moyen de la souris, la figure est alors mise à jour en temps réel.

Les logiciels les plus utilisés sont les précurseurs du domaine à savoir,

[1] que nous noterons D.G.S. (Dynamic Geometry Software).

[2] que nous noterons C.A.S. (Computer Algebra Software).

Geometer's sketchpad [Jac90] et Cabri Géomètre [LB98]. Ils sont apparus dans les années 90. Mais depuis de nombreux autres logiciels sont apparus, certains sont commerciaux, mais d'autres sont gratuits ou même libres. Le tableau 5.1 fournit une liste non exhaustive des logiciels de géométrie dynamique.

Le monde de l'éducation s'est intéressé à l'impact de l'utilisation de ces logiciels sur l'activité mathématique qui consiste à produire des *preuves* [Yev04, FD03].

Les logiciels de géométrie dynamique sont principalement utilisés dans le cadre de deux activités :

- pour faire faire aux étudiants des constructions géométriques ;
- pour faire faire aux étudiants des conjectures et/ou vérifier expérimentalement des faits.

Nous pensons que des logiciels peuvent aussi être utilisés dans le cadre de l'activité de preuve en elle-même et que l'utilisation d'un logiciel de preuve formelle peut faciliter l'apprentissage des mécanismes de démonstration.

Des travaux ont été menés dans cette direction, et plusieurs logiciels de géométrie dynamique avec des fonctionnalités liées à la preuve ont été produits. Ces systèmes peuvent être répartis en deux catégories :

1. les systèmes qui permettent de construire des preuves ;
2. les systèmes qui permettent de vérifier un fait en utilisant un système de démonstration automatique.

Les systèmes suivants appartiennent plutôt[3] à la première catégorie :

- *Geometry Tutor* [ABY85],
- *Mentoniezh* [Py90],
- *Defi* [AA92],
- *Chypre* [Ber93],
- *Cabri-Euclide* [Lue97],
- *Geometrix* [Gre98]
- *Baghera* [BPO$^+$02]

Ces systèmes permettent à l'élève de produire une preuve à partir d'une base de lemmes connus. Mais dans la plupart des cas, l'utilisateur ne peut pas inventer une preuve très différente de celles que connaît le programme. Par exemple il ne peut souvent pas ajouter d'objets au dessin ni même utiliser une proposition qui ne figure pas dans au moins une des preuves connues par le système. Les preuves « modèles » sont soit calculées par avance en utilisant des méthodes de démonstration automatique soit saisies par l'auteur de l'exercice. Dans le système Chypre, l'utilisateur n'est pas contraint à une démarche particulière de preuve, mais il doit tout de même utiliser une base de lemmes connus. Il semblerait que le logiciel *Cabri-Euclide* fasse exception. En effet il contient un petit système formel et donc laisse plus de

[3]Nous verrons qu'il est difficile de les classer précisement, chaque système ayant une approche différente.

TAB. 5.1 – Les principaux logiciels de géométrie dynamique

Logiciel	**Licence**
Baghera	?
Cabri	Commerciale
Cabri-Euclide	?
CaR	GPL
Chypre	
Cinderella	Commerciale
Déclic	Commerciale
Defi	
Dr. Geo	GPL
Eukleides	GPL
Gava	GPL
GeoFlash	Commerciale
GeoGebra	GPL
GeoLabo	GPL
GeoLog	Freeware
Geometria	Commerciale
Geometrix	Freeware
Geometry Explorer	Non diffusé
GeoNext	GPL
GeoPlanW	Shareware
GeoSpaceW	Shareware
GEOTHER	Gratuite pour un usage non commercial
GEUP	Commerciale
GeoView	Libre
GEX	Commerciale
GRACE	?
KGeo	GPL
KIG	GPL
Mentoniezh	
MM-Geometer	Commerciale
Sketchpad	Commerciale
XCas	GPL

liberté à l'utilisateur. Le logiciel *Baghera* fournit par ailleurs des fonctionnalités d'apprentissage en ligne, comme la gestion d'une liste d'exercices ou la communication par internet entre les professeurs et les étudiants.

Les logiciels suivants peuvent être classés dans la deuxième catégorie :
- *MMP-Geometer*[Gao00],
- *Geometry Expert* [GL02],
- *Geometry Explorer*[WF05],
- *GEOTHER*[Wan00]
- *Cinderella* [Kk99, RGKk99, Sch79, KRG04] .

Geometry Expert, MMP-Geometer, et GEOTHER sont des logiciels de géométrie dynamique qui sont utilisés comme une interface graphique pour une implantation des principales procédures de décision en géométrie.

Geometry Explorer consiste en une implantation en prolog de la méthode des angles de Chou [CGZ96]. Il fournit une représentation diagrammatique des preuves générées automatiquement par la méthode sous forme d'un arbre contenant des figures géométriques décorées par les faits connus à propos des angles à chaque étape de la démonstration.

Cinderella contient un « démonstrateur probabiliste » qui permet à l'utilisateur de vérifier des faits et au logiciel de simplifier certains calculs en interne. Cinderella permet aussi à l'utilisateur d'exporter la description de la figure vers des systèmes de calcul formel afin d'utiliser leur capacité à faire des preuves algébriques.

Les travaux les plus proches des nôtres sont ceux de Yves Bertot, Frédérique Guilhot et Loïc Pottier [BGP03] et de Judit Robu [Rob02].

Le système *GeoView* fournit un outil de visualisation pour certains énoncés géométriques exprimés en Coq. Geoview combine un logiciel existant de géométrie dynamique (GeoPlan) et l'interface PCoq pour l'assistant de preuve Coq [BT98, ABPR01]. Cette interface est conçue pour être utilisée conjointement avec la formalisation en Coq par Frédérique Guilhot de la géométrie telle qu'elle est enseignée en France au niveau lycée [Gui02, Gui04].

La thèse de Judit Robu propose une implantation de procédures de décision en géométrie au sein du système Theorema [4]. Theorema est un système de preuve implanté dans Mathematica. Son système permet d'engendrer automatiquement une figure interactive.

Nous présentons dans ce chapitre notre prototype appelé *GeoProof*. Nous avons décider de développer notre propre système afin de pouvoir l'adapter précisement à nos besoins. Il combine des méthodes de démonstration automatique, la possibilité de réaliser des preuves automatiques ou interactives

[4] `http://www.risc.uni-linz.ac.at/research/theorema/description/`

au moyen de l'assistant de preuve Coq et les fonctionnalités classiques d'un logiciel de géométrie dynamique.

Notre approche est guidée par les motivations suivantes :

– Il est très naturel en géométrie d'illustrer une preuve par une représentation diagrammatique. Dans certains cas le diagramme peut même être vu comme une représentation de haut niveau pour une preuve [WF05, Win04a, Win04b, Jam01, Mil01]. Mais parfois un diagramme peut induire en erreur. C'est pour cette raison que la vérification de la preuve dans un système de preuve formelle est cruciale. Elle permet d'atteindre un très haut niveau de confiance.

– Comparée à un système de preuve spécialisé en géométrie, l'utilisation d'un système de preuve tel que l'assistant de preuve Coq, fournit une façon de combiner des preuves géométriques avec des preuves de nature différente, utilisant potentiellement une logique plus riche. Par exemple, il est possible d'utiliser le système Coq pour prouver des faits à propos des polygones par induction sur le nombre de points que comporte le polygone, ou bien des faits à propos des isométries en utilisant les nombres complexes.

– Il y a des faits qui sont faciles à représenter par une figure et il y a des faits qui sont difficiles à représenter diagrammatiquement. Nous devons donc combiner les deux approches.

– Nous voulons avoir la possibilité de faire des preuves arbitrairement complexes, mais aussi d'utiliser une base de lemme connus suivant le niveau de l'utilisateur.

Ce chapitre est organisé de la façon suivante : nous donnons d'abord un aperçu général de notre prototype : *GeoProof*, puis nous nous concentrons sur les fonctionnalités liées à la preuve de *GeoProof* : la démonstration *automatique* et *interactive*.

5.2 Présentation de *GeoProof*

GeoProof est un logiciel à la fois libre et gratuit de géométrie dynamique. Il est distribué sous les termes de la licence GPL Version 2. L'implantation a été réalisée à partir d'un projet initié par *Nicolas François* : *DrGeoCaml*. *GeoProof* est écrit dans le langage *ocaml* en utilisant uniquement des bibliothèques portables de telle manière qu'il peut être compilé à la fois sous Linux, Windows et MacOSX. GeoProof propose les principales constructions et transformations manipulant des points, des cercles et/ou des droites. Les documents sont enregistrés en utilisant un format avec lequel il est facile d'être compatible puisqu'il est basé sur la technologie XML[5]. *GeoProof* est capable d'exporter les figures, soit sous forme « bitmap » en utilisant les formats PNG, BMP et JPEG, soit sous forme vectorielle en utilisant le format SVG. Le format SVG est le standard défini par le W3C pour les graphismes

[5]Nous utilisons notre propre DTD.

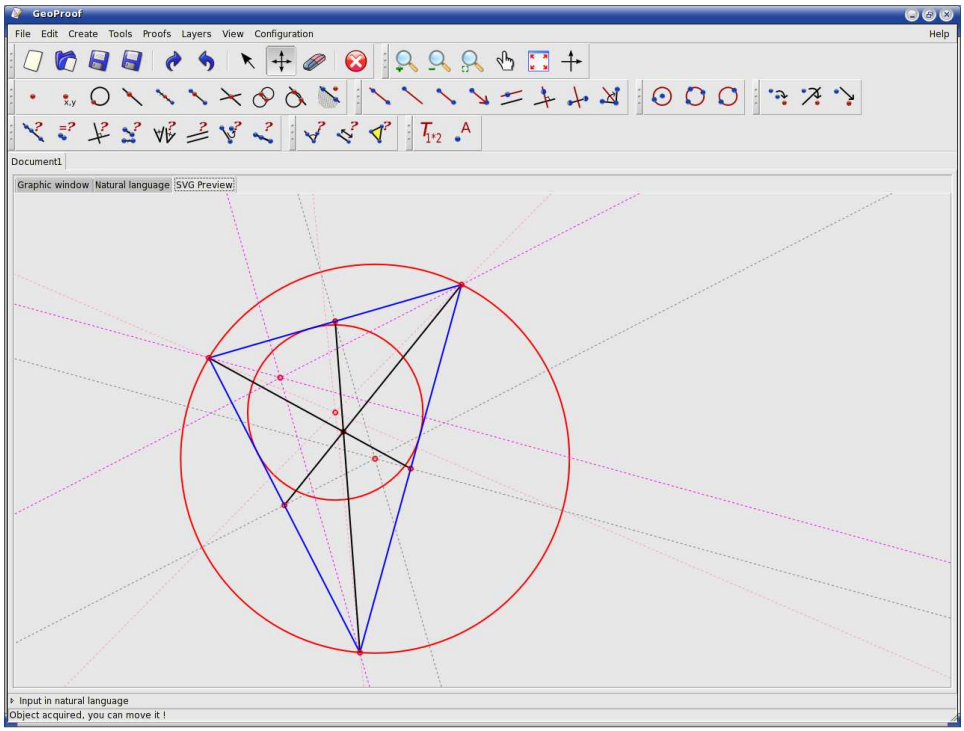


FIG. 5.1 – Une copie d'écran de *GeoProof*, l'exemple qui apparaît représente les points d'intérêt d'un triangle.

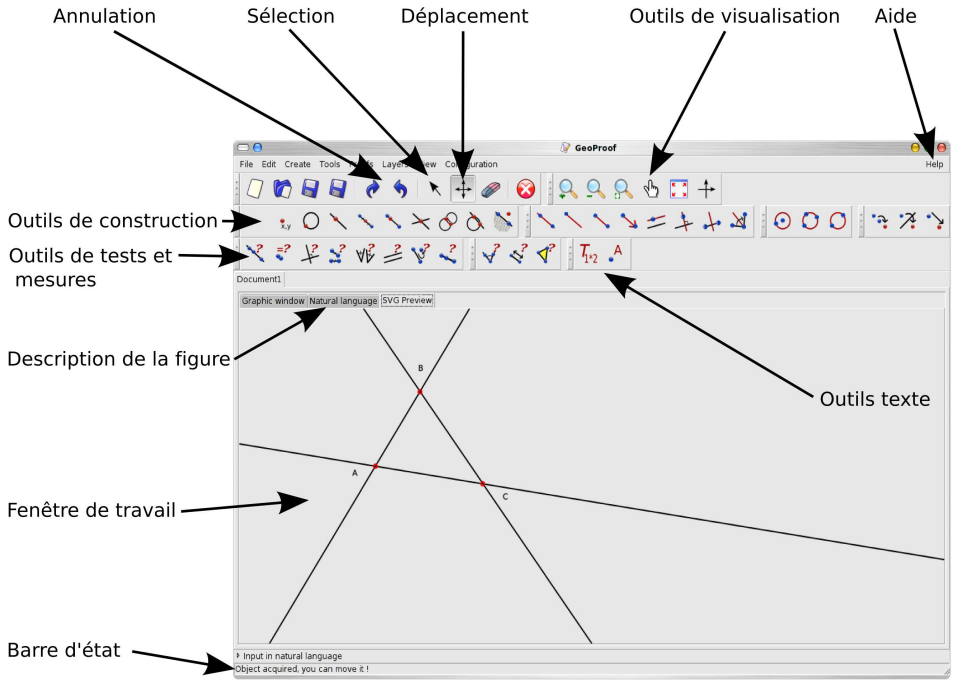
en deux dimensions sous forme vectorielle[6]. La figure peut aussi être traduite dans le langage *Eukleides*[7] pour faciliter l'insertion de figures dans un document LaTeX. Ce langage permet une description de haut niveau de la figure, seules les coordonnées des points libres apparaissent. Ainsi le script *Eukleides* généré par *GeoProof* est facilement modifiable de manière textuelle dans le code LaTeX sans avoir à ouvrir à nouveau la figure dans *GeoProof*. Les figures géométriques qui apparaissent dans cette thèse ont été générées grâce à ce procédé.

La figure 5.1 donne une idée de l'interface graphique de *GeoProof*.

Nous donnons maintenant un rapide aperçu des fonctionnalités de *GeoProof* en nous laissant guider par l'interface. Nous avons pris comme convention que, sur les icônes, l'objet créé est en rouge.

[6] Pour plus de détail voir `http://www.w3.org/Graphics/SVG/`

[7] `http://www.eukleides.org/`



GeoProof comporte les fonctionnalités suivantes :

Visualisation déplacement des objets libres, zoom avant, zoom arrière, zoom automatique, déplacement de la feuille, mode plein écran, affichage du repère, calques, ...

Construction points, droites, segments, vecteurs et cercles avec les constructions usuelles.

Transformations symétrie axiale, symétrie centrale, translation.

Tests et mesures collinéarité, congruence de segments, aire d'un triangle, longueur d'un segment, ...

Expressions expressions comportant des champs dynamiques avec expressions arithmétiques et logiques.

Attributs style de points, style de trait, épaisseur, couleur, ...

Fenêtre de travail vue normale, en SVG ou en langue naturelle.

Pour plus d'informations à propos des fonctionnalités de *GeoProof* voir la manuel de référence qui figure en annexe C.

Les fonctionnalités originales de *GeoProof* sont celles qui concernent la preuve formelle et que nous allons décrire plus en détail dans la partie suivante.

5.3 Démonstration automatique

Nous présentons dans cette partie la communication entre *GeoProof* et des méthodes de démonstration automatique. Nous avons implanté les fonctionnalités de démonstration automatique au moyen de deux systèmes différents : le premier est basé sur une implantation de la méthode des bases de Gröbner et de la méthode de Wu [Wu78, Cho88] écrite par John Harrison [Har03][8], la seconde consiste en notre implantation de la procédure de décision pour la géométrie affine que nous avons décrite dans le chapitre 4.

5.3.1 Méthode de démonstration automatique intégrée

La formalisation utilisée par John Harrison, est basée sur une théorie ne comportant que des points comme objet de base. Dans *GeoProof* trois types d'objets sont considérés comme les objets mathématiques de base :

- les points
- les droites
- les cercles

L'entrée du système de démonstration automatique est une formule en logique du premier ordre comportant les prédicats suivants :

- *collinear*, (noté *col*)
- *parallel*, (noté *par*)
- *perpendicular*, (noté *per*)
- *eq_distance* (que nous noterons $AB = CD$) et
- *eq_angle*.

Ces prédicats sont définis en utilisant les formules algébriques sur les coordonnées des points.

Soient x_P et y_P l'abscisse et l'ordonnée de P.

$$\begin{array}{lcl} col(A,B,C) & \equiv & (x_A - x_B)(y_B - y_C) - (x_B - x_C)(y_A - y_B) = 0 \\ par(A,B,C,D) & \equiv & (x_A - x_B)(y_C - y_D) - (x_C - x_D)(y_A - y_B) = 0 \\ per(A,B,C,D) & \equiv & (x_A - x_B)(x_C - x_D) - (y_A - y_B)(y_C - y_D) = 0 \end{array}$$

$eq\_distance(A,B,C,D) \equiv$

$$(x_A - x_B)^2 + (y_A - y_B)^2 - (x_C - x_D)^2 - (y_C - y_D)^2 = 0$$

$eq\_angle(A,B,C,D,E,F) \equiv$

$$\begin{array}{l} ((y_B - y_A) * (x_B - x_C) - (y_B - y_C) * (x_B - x_A)) * \\ ((x_E - x_D) * (x_E - x_F) + (y_E - y_D) * (y_E - y_F)) \\ = \\ ((y_E - y_D) * (x_E - x_F) - (y_E - y_F) * (x_E - x_D)) * \\ ((x_B - x_A) * (x_B - x_C) + (y_B - y_A) * (y_B - y_C)) \end{array}$$

[8] Attention, cette implantation a été conçue pour accompagner un livre sur la démonstration automatique, elle n'est pas censée être efficace.

Traduction d'une construction en un énoncé compréhensible par la méthode de démonstration automatique.

Nous devons traduire d'un langage à l'autre. L'idée de la traduction consiste à maintenir l'invariant que les droites et les cercles sont toujours définis par deux points. Bien sûr ce n'est pas toujours vrai dans *GeoProof*. Par exemple l'utilisateur peut construire une droite comme parallèle à une autre passant par un point. Dans ce cas nous définissons un second point pour la droite. De nouveaux points sont ainsi générés pendant la traduction. Les méthodes employées ne permettant pas de traiter l'orientation, nous ne traduisons pas les constructions impliquant l'orientation du plan ou de la droite comme β_T [9] ou $left\_turn$[10] par exemple. Pour plus d'informations à propos des méthodes de démonstration automatique en géométrie voir le chapitre 3. Nous allons définir la traduction en distinguant les cas selon la méthode qui a été utilisée pour construire l'objet.

Le tableau 5.2 donne les points qui définissent les droites et les cercles en fonction de leur mode de construction. $P1_l$,$P2_l$ et O_c sont des variables fraîches.
Les droites sont définies par deux points $\mathcal{P}_1(l)$ et $\mathcal{P}_2(l)$. Lorsque nous connaissons au moins un point sur la droite nous l'utilisons pour définir la droite au lieu de créer un nouveau point car cela simplifie les formules générées.
Les cercles sont définis par leur centre $\mathcal{O}(c)$ et un point sur le cercle $\mathcal{P}(c)$.
Le tableau 5.3 fournit la traduction de chacune des constructions de *GeoProof*[11] dans le langage accepté par le démonstrateur automatique intégré. Les prédicats qui sont précédés d'une négation correspondent aux conditions de non dégénérescence. La traduction dans son ensemble peut être vue comme la linéarisation d'une suite d'applications imbriquées de lemmes d'existence dont les hypothèses sont supposées vraies.

Les conditions de non dégénérescence sont inspirées par celle de [CG92]. Le prédicat *isotropique* est défini par :

$$isotropic(A, B) \equiv perpendicular(A, B, A, B)$$

En géométrie euclidienne c'est équivalent à $A = B$ mais ce n'est pas le cas en géométrie métrique. Par exemple, dans le plan complexe qui est un modèle de la géométrie métrique, une droite non dégénérée peut-être perpendiculaire à elle-même : toute droite de coefficient directeur i ou $-i$. Nous produisons un énoncé qui est interprété en géométrie métrique car les méthodes de Wu et Gröbner sont complètes pour la géométrie métrique. Pour plus d'information sur ce point voir [CG92, Cho88]. De plus si I_1 et I_2 sont les points d'intersection de deux cercles, ou d'une droite et d'un cercle alors nous ajoutons le fait

[9] $\beta_T\ A\,B\,C$ signifie que B appartient au segment $[AC]$, voir page 16.

[10] $left\_turn\ A\,B\,C$ signifie que C est à gauche de la droite (AB) considérée de A vers B.

[11] Pour simplifier la présentation nous ne donnons la traduction que pour les constructions principales de *GeoProof*.

TAB. 5.2 – Définition des points définition des cercles et des droites

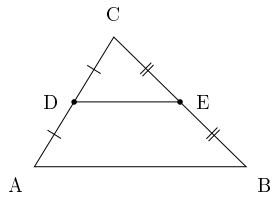
Construction de *GeoProof*	**Points de définition**
l passant par A et B	$\mathcal{P}_1(l) = A\ \mathcal{P}_2(l) = B$
l parallèle à m passant par A	$\mathcal{P}_1(l) = A\ \mathcal{P}_2(l) = P2_l$
l perpendiculaire à m passant par A	$\mathcal{P}_1(l) = A\ \mathcal{P}_2(l) = P2_l$
l la médiatrice de $[AB]$	$\mathcal{P}_1(l) = P1_l\ \mathcal{P}_2(l) = P2_l$
l la bissectrice de l'angle formé par A, B et C	$\mathcal{P}_1(l) = B\ \mathcal{P}_2(l) = P2_l$
c le cercle de centre O passant par A	$\mathcal{O}(c) = O\ \mathcal{P}(c) = A$
c le cercle passant par A,B et C	$\mathcal{O}(c) = O_c\ \mathcal{P}(c) = A$
c le cercle de diamètre A B	$\mathcal{O}(c) = O_c\ \mathcal{P}(c) = A$

que $I_1 \neq I_2$ dans les hypothèses. Notons que des constructions différentes de la même figure peuvent produire des conditions de dégénérescence différentes et donc des formules différentes.

Exemple

Nous prenons comme exemple, une fois de plus, le théorème de la droite des milieux :

Théorème 8. *Soit ABC un triangle*[12]*, et soient D et E les milieux de AC et BC respectivement. Le droite DE est parallèle à la base AB.*



La construction est traduite en l'énoncé suivant :

```
(((((is_midpoint(D,C,A) /\ is_midpoint(E,C,B))/\
~C=A) /\ ~A=B) /\ ~B=C) /\ ~D=E) /\ ~A=B
```

Le fait que $AB \parallel DE$ est vérifié par la méthode des bases de Gröbner.

Traitement des conditions de non dégénérescence.

Les conditions de non dégénérescence jouent un rôle crucial en géométrie formelle, comme nous avons pu le voir dans les chapitres précédents, et comme d'autres auteurs ont pu le remarquer [Gui04, MF03, Nar04]. Cette

[12]Notons que par le terme « triangle » nous voulons désigner uniquement un triplet de points. Ceux-ci ne sont pas nécessairement distincts et non alignés. L'hypothèse qui énonce le fait que les points A et B sont distincts vient de la construction du segment $[AB]$. A la création du segment, *Geoproof* engendre cette condition de non dégénérescence.

TAB. 5.3 – Prédicat correspondant à chaque type de construction

***GeoProof* Construction**	**Prédicat**
Point libre	$true$
Point P sur la droite l	$collinear(P, \mathcal{P}_1(l), \mathcal{P}_2(l))$
Point P sur le cercle c	$\mathcal{O}(c)\mathcal{P}(c) = P\mathcal{O}(c)$
I milieu de A B	$IA = IB \wedge collinear(I, A, B)$
I intersection de l_1 et l_2	$collinear(I, \mathcal{P}_1(l_1), \mathcal{P}_2(l_1)) \wedge$ $collinear(I, \mathcal{P}_1(l_2), \mathcal{P}_2(l_2)) \wedge$ $\neg parallel(\mathcal{P}_1(l_1), \mathcal{P}_2(l_1), \mathcal{P}_1(l_2), \mathcal{P}_2(l_2))$
I une intersection de c_1 et c_2	$I\mathcal{O}(c_1) = \mathcal{O}(c_1)\mathcal{P}(c_1) \wedge$ $I\mathcal{O}(c_2) = \mathcal{O}(c_2)\mathcal{P}(c_2) \wedge$ $\neg isotropic(\mathcal{O}(c_1), \mathcal{O}(c_2))$
I une intersection de c et l	$I\mathcal{O}(c) = \mathcal{O}(c)\mathcal{P}(c) \wedge$ $collinear(I, \mathcal{P}_1(l), \mathcal{P}_2(l)) \wedge$ $\neg isotropic(\mathcal{P}_1(l), \mathcal{P}_2(l))$
l passant par A et B	$A \neq B$
l parallèle à m passant par A	$parallel(A, \mathcal{P}_2(l), \mathcal{P}_1(m), \mathcal{P}_2(m)) \wedge$ $A \neq \mathcal{P}_2(l)$
l perpendiculaire à m passant par A	$perpendicular(A, \mathcal{P}_2(l), \mathcal{P}_1(m), \mathcal{P}_2(m)) \wedge$ $A \neq \mathcal{P}_2(l)$
l médiatrice de $[AB]$	$\mathcal{P}_1(l)A = \mathcal{P}_1(l)B \wedge \mathcal{P}_2(l)A = \mathcal{P}_2(l)B \wedge$ $\mathcal{P}_1(l) \neq \mathcal{P}_2(l) \wedge A \neq B$
l bissectrice de l'angle A,B,C	$eq\_angle(A, B, \mathcal{P}_2(l), \mathcal{P}_2(l), B, C) \wedge$ $B \neq \mathcal{P}_2(l) \wedge A \neq B \wedge B \neq C$
c cercle de centre O passant par A	$true$
c cercle passant par A,B et C	$\mathcal{O}(c)A = \mathcal{O}(c)B \wedge \mathcal{O}(c)B = \mathcal{O}(c)C$ $\neg collinear(A, B, C)$
c cercle de diamètre A B	$collinear(\mathcal{O}(c), A, B) \wedge$ $\mathcal{O}(c)A = \mathcal{O}(c)B$

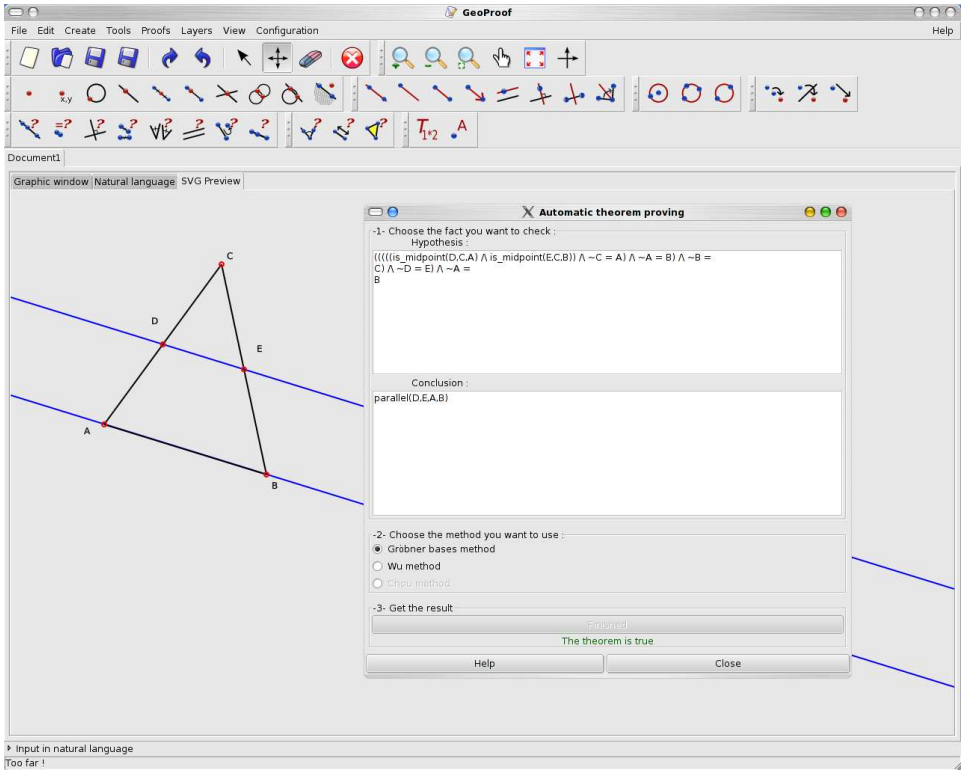


FIG. 5.2 – Vérifions le théorème de la droite des milieux en utilisant le démonstrateur automatique intégré.

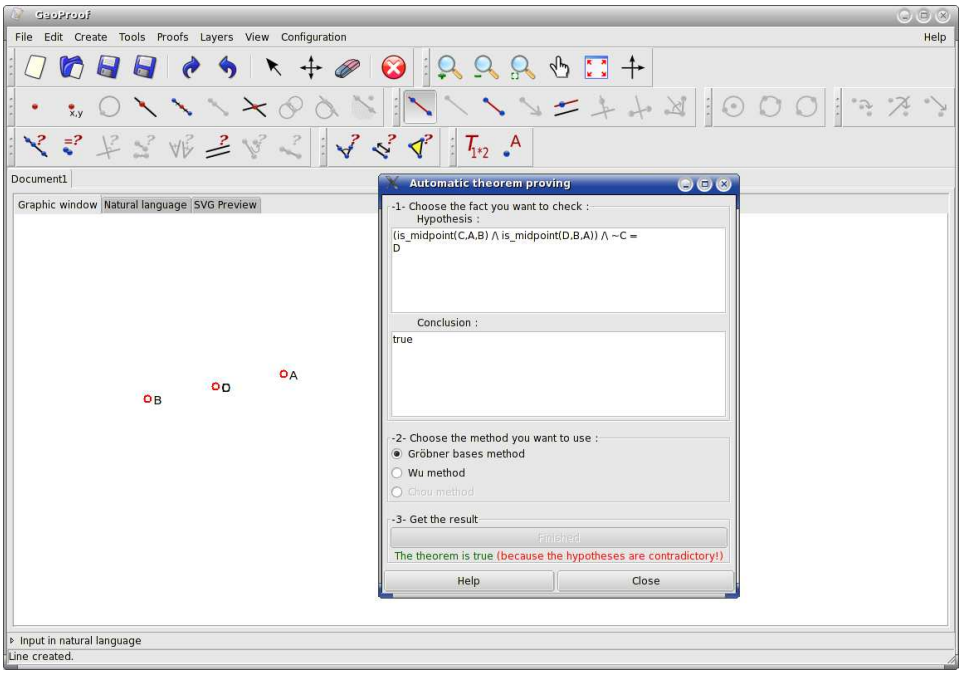


FIG. 5.3 – Preuve d'une propriété avec des hypothèses contradictoires

traduction n'est pas une exception, nous devons faire attention à la sémantique des énoncés générés. Pour cette traduction nous avons décidé de considérer *GeoProof* comme un outil qui permet de définir une formule géométrique. Il n'est pas censé construire un modèle de cette formule. L'utilisateur peut définir des figures « impossibles ». Par exemple si nous effectuons la construction suivante :
D'abord construire un point A puis tracer la droite passant par A et A. Alors si nous essayons de prouver que $A \neq A$, *GeoProof* répondra par l'affirmative, puisque les hypothèses du théorème sont contradictoires (*ex falso quod libet*). Ceci est « logique » mais va à l'encontre de l'intuition de l'utilisateur parce que les objets « impossibles » ne sont pas représentés par *GeoProof*. C'est la raison pour laquelle en fait nous vérifions d'abord si nous pouvons prouver *faux*, si c'est le cas nous avertissons l'utilisateur que sa construction est impossible comme on peut le voir sur la figure 5.3. Notons que sur l'exemple que nous montrons nous n'avons pas exactement créé la droite passant par A et A, en effet *GeoProof* interdit ce genre de constructions dégénérées en particulier. Nous avons créé deux points égaux (C et D) en utilisant l'outil milieu appliqué deux fois au même segment. Empêcher l'utilisateur de produire des constructions impossibles nécessiterait d'utiliser le démonstrateur automatique, c'est ce qui est réalisé dans Cinderella, voir [Kk99, RGKk99, Sch79, KRG04].

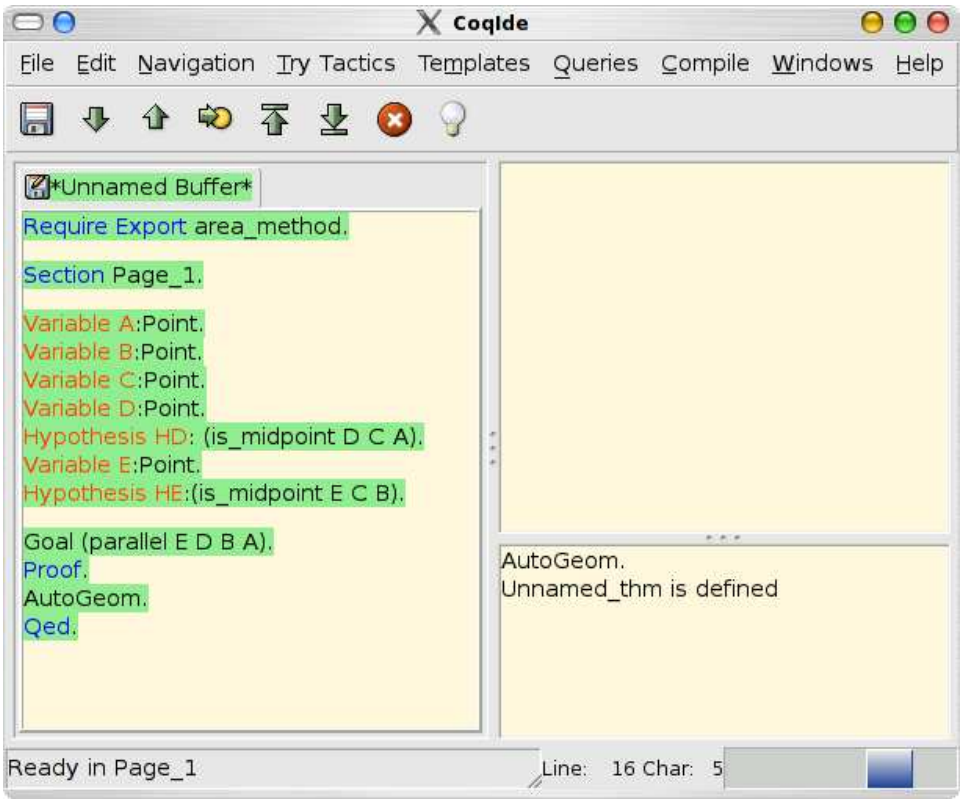


FIG. 5.4 – Le théorème de la droite des milieux, exprimé en Coq dans le langage adapté à la procédure de décision.

5.3.2 Avec Coq

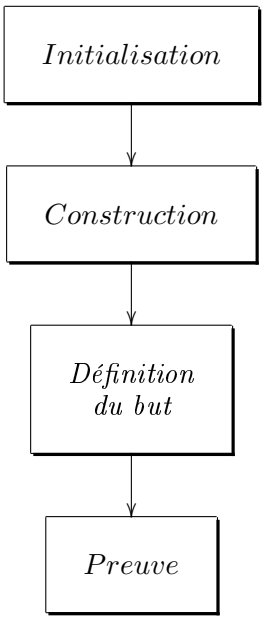
Dans le chapitre 4 nous avons décrit l'implantation de la procédure de décision de Chou, Gao et Zhang dans l'assistant de preuve Coq. Nous nous intéressons maintenant à l'exportation d'une construction réalisée au moyen de *GeoProof* dans le langage de notre développement Coq. Notre implantation de la procédure de décision de Chou, Gao et Zhang est restreinte à la géométrie affine plane, c'est pourquoi les constructions de *GeoProof* qui ne possède pas de concept correspondant dans l'implantation Coq sont grisés. Comme nous avons déjà pu le remarquer, l'axiomatique sur laquelle notre développement Coq (voir page 69), se fonde est une axiomatique qui comporte uniquement des points comme objets de base. Ainsi nous devons réaliser une traduction similaire à celle décrite dans le paragraphe précédent.

5.4 Preuve interactive

Dans cette partie nous décrivons le fonctionnement de *GeoProof* en mode de preuve interactive. Via le menu de configuration, l'utilisateur peut choisir entre deux modes de preuve interactive, le premier utilise le langage décrit dans la partie 5.3.2 et le deuxième utilise le langage du développement Coq pour la géométrie de Frédérique Guilhot [Gui04]. Dans le premier mode l'utilisateur peut traiter la géométrie affine et dans le second la géométrie euclidienne plane[13]. L'interaction entre l'utilisateur et Coq est réalisée à travers l'interface CoqIDE. *GeoProof* communique avec CoqIDE[14] grâce à un canal privé.

Notre implantation consiste principalement en une traduction d'une construction *GeoProof* en un énoncé Coq. Nous réalisons la même traduction que celle décrite dans [BGP03] mais dans la direction inverse (ici nous traduisons *vers* Coq)[15].

Le mode interactif de *GeoProof* se décompose en quatre phases :



Dans la phase d'initialisation, la communication entre *CoqIDE* et *GeoProof* est lancée. En fonction du langage utilisé, les outils de construction qui ne peuvent pas être exportés en Coq sont grisés dans *GeoProof*. Les définitions Coq correspondant au langage utilisé sont chargées en utilisant la

[13] La formalisation de Frédérique Guilhot inclut aussi la géométrie spatiale mais *GeoProof* est limité à la dimension deux.

[14] Cette fonctionnalité nécessite CoqIDE version 8.1 ou supérieure.

[15] À l'avenir, nous devrions fusionner nos développements pour permettre la communication dans les deux directions, ceci nécessite un mécanisme de communication plus complexe comme nous l'expliquons dans la partie « perspectives ».

commande Coq `Require`. Une nouvelle section Coq est ouverte. Dans le cas où l'utilisateur a déjà construit des objets avant d'avoir lancé le mode de preuve interactive, ces objets sont exportés en Coq. Les objets qui n'ont pas de signification dans le langage sélectionné sont ignorés.

Dans la phase de construction les objets créés par l'utilisateur sont ajoutés au contexte Coq avec les prédicats qui leur sont associés. Sur l'exemple qui apparaît sur la figure 5.6 cela correspond aux lignes qui commencent par les commandes `Variable` et `Hypothesis`.

Dans la phase « Définition du but » l'utilisateur doit préciser ce qu'il désire prouver. Dans le contexte de l'enseignement des mathématiques cette étape peut être présentée comme un exercice qui consiste à trouver des conjectures à propos de la figure proposée par le professeur. C'est à cette fin que *GeoProof* fournit des labels qui peuvent contenir des parties dynamiques (voir page 94 et pour plus de détails voir le manuel de référence en annexe aux pages 177 et 178). Si l'utilisateur veut prouver un fait qui fait l'objet d'un test dans un label, il peut cliquer sur le fait avec le bouton droit et choisir l'entrée de menu correspondant, le prédicat est alors traduit dans Coq.

Dans la phase « Preuve » l'utilisateur prouve son énoncé au sein de *CoqIDE*. Si durant la preuve il est nécessaire de créer un nouvel objet, il peut le faire en utilisant *GeoProof*. En effet lorsque un nouvel objet est créé dans *GeoProof* la tactique Coq correspondant à la preuve de l'existence de cet objet est insérée dans *CoqIDE*. Cette tactique applique le théorème qui prouve l'existence de l'objet qui vient d'être créé et introduit dans le contexte ce que l'on sait de ce nouvel objet. Dans certains cas, cela génère des conditions de non dégénérescence qui doivent être prouvées par l'utilisateur (ou ajoutées comme hypothèses au théorème). La figure 5.5 montre la tactique (définie en $\mathcal{L}_{tac}$: le langage de tactique de Coq) qui est utilisée pour créer le point d'intersection de deux droites.
Si l'utilisateur efface un objet dans *GeoProof* il est effacé du contexte Coq au moyen de la commande `clear` de Coq. Cela permet de simplifier la figure quand l'état de la preuve ne nécessite plus de visualiser tous les objets. Si l'utilisateur désire effacer un objet dans *GeoProof* sans l'effacer du contexte Coq, il peut le cacher au moyen du menu contextuel propre à l'objet dans *GeoProof*.

5.5 Perspectives

La version actuelle de *GeoProof* utilise un presse papier privé comme tuyau de communication entre *GeoProof* et *CoqIDE*. Cette approche a l'avantage d'être à la fois facile à implanter et facile à utiliser. L'utilisateur peut commencer une interaction sans avoir à configurer quoi que ce soit. Il doit seulement lancer *GeoProof* et *CoqIDE* sur le même ordinateur. Mais cette

FIG. 5.5 – La tactique qui permet d'introduire le point d'intersection de deux droites.

```
Ltac DecompEx H P := elim H;intro P;intro;clear H.

Ltac let_intersection I A B C D :=
let id1 := fresh in ((assert (id1:exists I,
I =  pt_intersection (line A B) (line C D));
[apply (existence_pt_intersection)|DecompEx id1 I])).
```

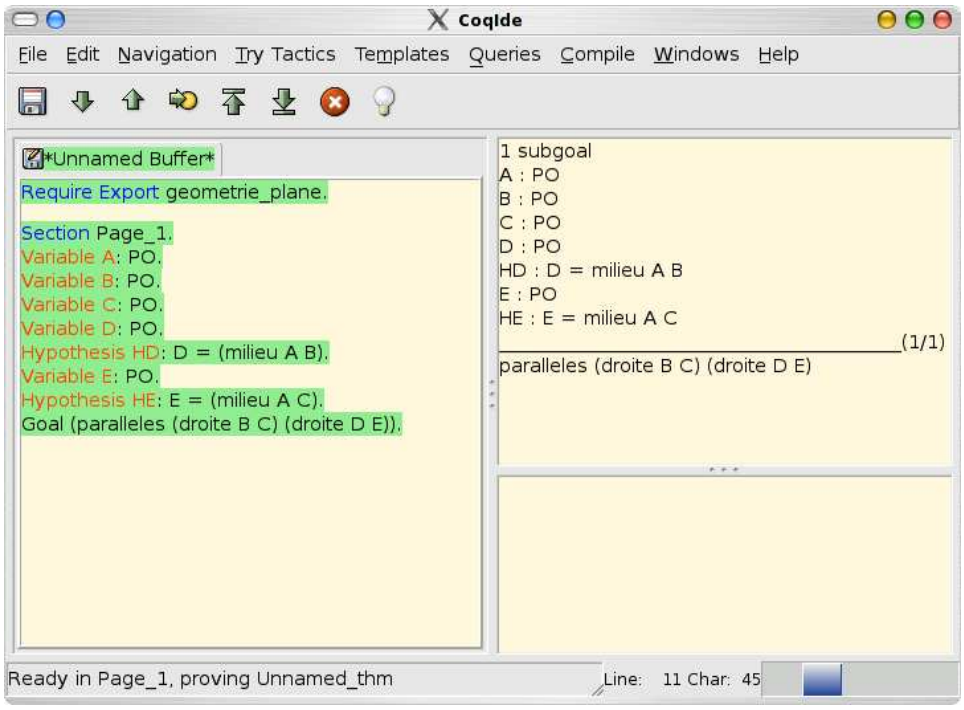
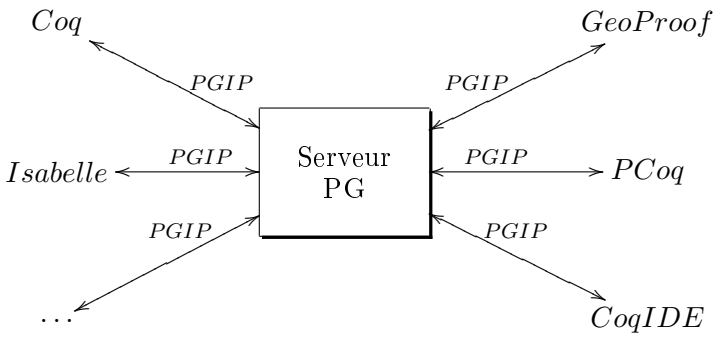


FIG. 5.6 – Le théorème de la droite des milieux dans le langage du développement de Frédérique Guilhot.

FIG. 5.7 – Intégration de *GeoProof* dans l'architecture de *Proof General*



architecture possède des limites. Premièrement, la communication avec Coq est réalisée en utilisant la syntaxe de Coq, elle est relativement simple à produire mais il est difficile d'en réaliser l'analyse syntaxique (à cause des mécanismes de notations notamment). Deuxièmement, la synchronisation entre ce qui est saisi par l'utilisateur dans CoqIDE et l'état de *GeoProof* n'est pas garantie. Il pourrait donc être intéressant de fonder le mécanisme de communication entre *Coq* et *GeoProof* sur l'architecture « Proof General Interaction Protocol (PGIP) » [WAL04, ALW04]. Ce cadre de travail est basé sur XML et permet d'avoir plusieurs interfaces qui interagissent avec le même assistant de preuve. C'est exactement ce dont nous avons besoin car, comme nous l'avons mentionné plus haut, certaines preuves sont plus faciles à comprendre diagrammatiquement et d'autres sont plus faciles à représenter de manière textuelle (on peut penser aux preuves utilisant les nombres complexes par exemple). Dans ce cadre, *GeoProof* et *CoqIDE* interagiraient avec l'assistant de preuve Coq. Cette approche pourrait de plus être généralisée à d'autres assistants de preuve et interfaces graphiques tels que Isabelle, Eclipse/Proof General et PCoq par exemple. La figure 5.7 montre la structure de cette architecture. Cette approche nécessite l'implantation de PGIP dans Coq, CoqIDE et GeoProof.

De plus, les fonctionnalités de preuve interactive de *GeoProof* se doivent d'être étendues. Nous travaillons sur les extensions suivantes :

- avoir la possibilité d'appliquer un théorème diagrammatiquement par glisser/déposer,
- avoir la possibilité de marquer des faits sur le diagramme pour produire de nouvelles propositions dans Coq,
- pouvoir considérer une macro comme une preuve de l'existence d'une construction à la règle et au compas (cela sous entend qu'il faut ajouter la possibilité de réaliser des macros à *GeoProof*).

Une autre extension prévue de *GeoProof* est de l'adapter afin de pouvoir réaliser des raisonnements « diagrammatiques ». En effet certains raisonne-

ments géométriques peuvent facilement être représentés de manière diagrammatique. Afin d'aborder cette thématique, nous nous sommes intéressés à un domaine où les raisonnements géométriques sont fréquents et permettent de traiter une large classe de formules : la réécriture abstraite. Ceci fait l'objet du chapitre suivant.

5.6 Conclusion

La démonstration est au centre des mathématiques et doit donc également avoir un rôle central dans l'enseignement des mathématiques. Les logiciels le plus utilisés pour l'enseignement des mathématiques le sont principalement pour explorer, visualiser, calculer, trouver des contre-exemples, des conjectures, ou vérifier des faits, mais la plupart ne permettent pas de réaliser des preuves. Nous pensons que les assistants de preuve sont maintenant assez mûrs pour être adaptés au monde de l'éducation. Nous avons présenté dans ce chapitre un prototype qui tente d'intégrer la géométrie dynamique, la preuve automatique et la preuve formelle. C'est un premier pas vers l'utilisation d'un assistant de preuve en classe. Nous espérons pouvoir dans l'avenir poursuivre ce travail en étudiant les problématiques de la preuve formelle qui sont spécifiques à l'enseignement (respect du programme, *etc.*).

Chapitre 6

Un système de preuve diagrammatique pour la réécriture abstraite

Les diagrammes sont utilisés régulièrement dans la communauté qui s'intéresse à la réécriture à la fois pour représenter certaines propriétés mais aussi certaines preuves. Dans ce chapitre nous formalisons une certaine classe de diagrammes qui figure dans la littérature à propos de la réécriture abstraite. Nous donnons une définition formelle des digrammes utilisés pour énoncer des propriétés. Nous proposons des règles d'inférence pour formaliser certaines preuves diagrammatiques comme par exemple certaines propriétés de transitivité de systèmes de réécriture abstraite et le lemme de Newman. Nous montrons la correction et la complétude du système par rapport au calcul des séquents.

6.1 Introduction

Certains diagrammes peuvent être vus comme des descriptions de haut niveau pour des preuves dans le sens où ils convainquent le lecteur qu'une propriété est vraie. Ce genre de diagrammes apparaît dans différents domaines des mathématiques et de l'informatique, nous pouvons citer entre autres la géométrie euclidienne, la théorie des nombres, l'analyse réelle, la théorie des ensembles, la théorie des catégories, la réécriture...

Dans [Jam01] Jamnik utilise des diagrammes comme des aides pour un démonstrateur automatique dans le domaine de la théorie des nombres. Miller a proposé un système formel pour faire des preuves diagrammatiques formelles en géométrie euclidienne [Mil01]. Winterstein a proposé un autre système formel qui lui s'intéresse à la preuve diagrammatique pour analyse

réelle [Win04b].

Dans ce chapitre nous nous intéressons aux diagrammes que l'on peut trouver dans la littérature liée à la théorie de la réécriture. Dave Barker-Plummer et Sidney C. Bailin se sont interessés au raisonnement diagrammatique en réécriture [BPB91], mais dans leur approche le digramme était considéré seulement comme une *aide* pour un démonstrateur automatique.

Notre but ici est de donner aux diagrammes le statut d'objet de preuve afin de les utiliser comme langage d'entrée de haut niveau pour l'assistant de preuve Coq [Coq04, HKPM04]. Cette approche nous amène à donner une définition formelle de ce qu'est un diagramme, sa sémantique et la correction d'une preuve diagrammatique.

Nous nous intéressons à la réécriture car les diagrammes sont utilisés fréquemment dans la littérature sur ce domaine. Par exemple dans [BN98] des diagrammes apparaissent tout au long de la présentation. Leur sémantique est même définie assez précisément. Néanmoins nous verrons qu'il est nécessaire de donner une définition plus formelle puisque la signification de certains diagrammes dans [BN98] varie suivant le contexte. En effet dans cette présentation les variables sont parfois implicitement quantifiées universellement et d'autres fois elles ne le sont pas.

Dans ce chapitre nous donnons une présentation des principales propriétés de la réécriture abstraite similaire à [BN98] sauf que notre but est de considérer les diagrammes non pas comme des illustrations pour les preuves mais comme des objets de preuve en eux-mêmes.

Nous rappelons d'abord la définition d'un système de réécriture abstraite, et donnons une définition formelle d'un diagramme de réécriture. Ensuite nous définissons quelques propriétés représentables diagrammatiquement et donnons un système formel en nous appuyant sur un exemple simple. Enfin nous ajoutons des règles d'inférence pour formaliser les preuves par induction en nous appuyant sur l'exemple du lemme de Newman [New42]. Enfin nous décrivons l'implantation du système qui a été réalisée au sein de l'assistant de preuve Coq et permet de prouver les propriétés principales des systèmes de réécriture abstraite.

6.2 Représentation diagrammatique en réécriture abstraite

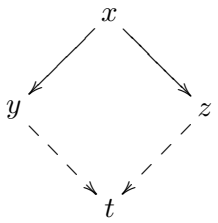
Dans cette partie nous rappelons la définition d'un système de réécriture abstraite et donnons des définitions pour les diagrammes qui représentent des propriétés classiques.

Un *système de réécriture abstraite* est une paire $(A, \rightarrow)$ où la *réduction* $\rightarrow$ est une relation binaire sur l'ensemble A, i.e. $\rightarrow \subseteq A \times A$.

Notre but ici est de formaliser les diagrammes tels qu'on peut les trouver dans la littérature. Nous ne voulons pas définir une nouvelle sorte de diagrammes comme cela est fait dans [BvOK98], nous nous efforçons de définir un langage qui sera utilisé en entrée de Coq. Celui-ci doit être aussi proche que possible de l'usage habituel. Ainsi, le fait que (x, y) appartient à $\rightarrow$ sera représenté par une flèche en position infixe : $x \longrightarrow y$.

Informellement, nous utiliserons la convention habituelle qui consiste à représenter les hypothèses par des flèches pleines et la conclusion par des flèches en pointillé. Les sommets qui sont reliés uniquement à des flèches en pointillé sont supposés, par défaut, être quantifiés existentiellement.
Les sommets qui sont reliés uniquement à des flèches pleines sont toujours quantifiés universellement.

Avant de donner une définition formelle, examinons un premier exemple. Une propriété très connue d'un système de réécriture abstraite est la propriété du diamant. Elle est habituellement représentée par le diagramme suivant :



La *signification* de ce diagramme est la suivante :

$$\forall xyz,\ x \longrightarrow y \wedge x \longrightarrow z \Rightarrow \exists t,\ y \longrightarrow t \wedge z \longrightarrow t$$

Ici notre but est de traiter les diagrammes comme des citoyens de première classe, c'est à dire non pas comme des notations pour des objets mathématiques mais comme des objets mathématiques, pour cela nous avons besoin d'un définition formelle de ce qu'est un diagramme.

La définition d'un diagramme étant basée sur un graphe, nous définissons d'abord les graphes dont on a besoin ainsi que les notations associées.

Définition 17 (Multi-graphe orienté)**.** *Un multi graphe orienté est un quadruplet* (V, A, s, d) *où*

- V *est l'ensemble des sommets*
- A *est l'ensemble des flèches*
- $s : A \to V$ *est la fonction qui à chaque flèche associe son sommet source*
- $d : A \to V$ *est la fonction qui à chaque flèche associe son sommet destination*

Notons qu'une flèche peut avoir une source et une destination identiques.

Définition 18 (Diagramme)**.** *Un diagramme de réécriture* D *est un multigraphe orienté fini dont les flèches sont étiquetées par une relation et un statut (soit* conclusion *soit* hypothèse*) et les sommets sont étiquetés par un nom et un statut (soit* universel, existentiel *ou* libre*) vérifiant les conditions suivantes :*

- *Si un sommet est en contact avec au moins une flèche hypothèse alors sont statut n'est pas existentiel.*
- *Le diagramme comporte au moins une flèche marquée conclusion.*
- *Le diagramme ne comporte pas de sommet de degré zéro.*

Formellement c'est un 10-uplet $(\Sigma_V, \Sigma_A, V, A, s, d, l_A, l_V, s_A, s_V)$ *où :*

- Σ_V *est un ensemble de symboles de sommets*
- Σ_A *est un ensemble de symboles de relations*
- V *est l'ensemble des sommets*
- A *est l'ensemble des flèches*
- $s : A \to V$ *est la fonction qui à chaque flèche associe son sommet source*
- $d : A \to V$ *est la fonction qui à chaque flèche associe son sommet destination*
- $l_A : A \to \Sigma_A$ *est une fonction qui associe à chaque flèche un symbole de relation*
- $l_V : V \to \Sigma_V$ *est une fonction injective qui associe à chaque sommet un symbole de sommet*
- $s_A : A \to \{\mathcal{H}, \mathcal{C}\}$ *est une fonction qui associe à chaque flèche un statut de flèche*
- $s_V : V \to \{\forall, \exists, \mathcal{F}\}$ *est une fonction qui associe à chaque sommet un statut de sommet*

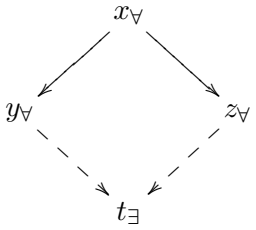
vérifiant que :

- $\forall v \in V, (\exists a \in A,\ (s(a) = v \vee d(a) = v) \wedge s_A(a) = \mathcal{H}) \Rightarrow s_V(v) \neq \exists$
- $\exists a \in A, s_A(a) = \mathcal{C}$
- $\forall v \in V, \exists a \in A, s(a) = v \vee d(a) = v$

Premières notations (N1) :

Si toutes les flèches sont étiquetées par la même relation nous omettons ce label.
Les flèches qui ont le statut conclusion sont représentées par une flèche en pointillé.
Les flèches qui ont le statut hypothèse sont représentées par une flèche pleine.
Les sommets qui ont le statut universel sont marqués par le symbole $\forall$.
Les sommets qui ont le statut existentiel sont marqués par le symbole $\exists$.
Les sommets libres sont soulignés.

En utilisant ces notations la propriété du diamant peut être représentée de la façon suivante :



Nous disons qu'un terme $x \xrightarrow{R} y$ est représenté par une flèche si le diagramme contient une flèche f étiquetée par R tel que $s(f) = x$ et $d(f) = y$.

Maintenant nous définissons une sémantique pour nos diagrammes. Notons que cette définition n'est pas nécessaire à la construction d'un système formel pour faire des preuves en réécriture. En effet on pourrait considérer que la sémantique des diagrammes est implicitement définie par les règles d'inférence qui les manipulent. Nous donnons ici la sémantique d'une part pour clarifier la présentation et d'autre part parce qu'elle est nécessaire pour énoncer les propriétés de correction et de complétude vis à vis du calcul des séquents que nous prouverons dans la partie 6.4.

Définition 19 (sémantique)**.**
La sémantique d'une flèche $x \xrightarrow{R} y$ *est* $R(x, y)$*.*
Soit $\overline{e}$ *l'ensemble des étiquettes des sommets marqués existentiellement et* $\overline{u}$ *l'ensemble des étiquettes des sommets marqués universellement.*
Soit C *la conjonction des termes représentés par une flèche conclusion.*
Soit H *la conjonction des termes représentés par une flèche hypothèse ou vrai si la conjonction est vide.*

Par définition la sémantique d'un diagramme D *notée* $[\![D]\!]$ *est :*

$$[\![D]\!] := \forall \overline{u}, H \Rightarrow \exists \overline{e}, C$$

Notons que grâce à la première condition dans la définition des diagrammes, la conjonction C ne peut pas être vide et grâce à la deuxième condition H ne peut pas contenir d'occurrence d'une variable qui est contenue dans $\overline{e}$
Notons aussi que nous ne précisons pas l'ordre des variables dans $\overline{e}$ et $\overline{u}$ ainsi que l'ordre des termes dans C et H mais ceci n'introduit pas fondamentalement d'ambiguïté car les formules obtenues par permutation sont équivalentes.

Avec cette définition, il est clair que toutes les formules de la logique du premier ordre ne sont pas la sémantique d'un diagramme. Nous pouvons uniquement décrire des formules de la forme $\forall \overline{u} \bigwedge_i H_i \Rightarrow \exists \overline{e} \bigwedge_i C_i$ où les H_i et C_i sont des prédicats d'arité deux. Quand nous aurons besoin d'une formule qui n'est pas dans ce fragment nous nous servirons de la syntaxe habituelle utilisant les connecteurs logiques.

Remarque 3. *Si un diagramme contient plusieurs composantes connexes (au sens de la théorie des graphes), sa sémantique est équivalente à la conjonction des sémantiques de chacune de ses composantes connexes.*

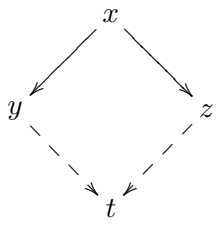
Preuve: Par injectivité de la fonction l_V*.* □

Secondes notations (N2) :

Comme notre but est de donner une définition des diagrammes aussi proche que possible de l'usage habituel dans la communauté, nous introduisons deux autres notations :

1. Dans la représentation d'un diagramme si nous omettons le statut d'un sommet, alors il a le statut implicite suivant :
   Si un sommet est en contact avec seulement des flèches conclusion alors son statut est existentiel sinon son statut est universel.

Ainsi on retrouve la notation habituelle pour la propriété du diamant :



2. *Si pour représenter un diagramme, on dessine* seulement *des flèches pleines et que l'on omet le statut des sommets, nous considérons ceci comme une notation pour représenter le diagramme avec le même graphe sous jacent mais composé uniquement de flèches en pointillé et de sommets libres.*
   Exemple : $x \longrightarrow y$ est une notation pour $\underline{x} - - \succ \underline{y}$
   Notons que cette notation ne crée pas d'ambiguïté car tout diagramme comporte au moins une flèche conclusion.
   Notons aussi que si nous échangions les rôles des flèches en pointillé ou non dans la définition de la sémantique d'un diagramme nous pourrions simplifier cette notation. Nous gardons cette convention pour rester le plus proche possible de l'usage dans la communauté.

Avant d'aller plus loin, et pour clarifier ces définitions, voici quelques exemples de diagrammes avec leur sémantique associée :

Formule	Diagramme
$x \longrightarrow x$	$\underline{x}$ noté aussi[a] x
$\forall x,\ x \longrightarrow x$	$x_\forall$
$\exists x,\ x \longrightarrow x$	x
$\exists xy,\ x \longrightarrow y$	$x - - \succ y$
$\forall x \exists y,\ x \longrightarrow y$	$x_\forall - - \succ y$
$\forall xy,\ x \longrightarrow y$	$x_\forall - - \succ y_\forall$
$x \longrightarrow y$	$\underline{x} - - \succ \underline{y}$ noté aussi $x \longrightarrow y$

[a] en l'absence d'autres flèches dans le diagramme

6.2.1 Extension aux disjonctions.

Habituellement, dans la littérature sur la réécriture, la disjonction n'est pas représentée par un diagramme. Mais afin de pouvoir définir la clôture transitive d'une relation par exemple, nous sommes amenés à définir des

diagrammes pour la disjonction. En effet nous devons exprimer le fait que[1] :

$$\forall xy, x \xrightarrow{+} y \Rightarrow (x \longrightarrow y \vee \exists y', x \longrightarrow y' \xrightarrow{+} y)$$

Pour cela nous devons étendre notre définition de diagramme à une notion de diagramme disjonctif :

Définition 20 (diagramme disjonctif)**.** *Un diagramme disjonctif est un ensemble fini de diagrammes (dans le sens de la définition 18) dont les sous-diagrammes restreints aux flèches pleines et sommets universels sont identiques.*

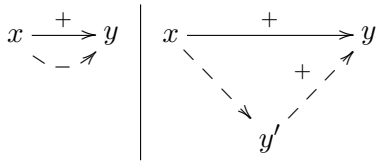
Notation : Nous séparons les sous-diagrammes formant la disjonction par une barre verticale |.

La sémantique est la suivante :

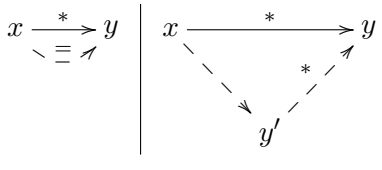
Définition 21 (sémantique des diagrammes disjonctifs)**.**
Soit $D = \{D_1 \dots D_n\}$ un diagramme disjonctif. Comme les diagrammes D_i partagent les mêmes flèches pleines nous savons qu'ils ont une sémantique de la forme $\forall \overrightarrow{u}, H \Rightarrow \exists \overrightarrow{e_i}, C_i$.
La sémantique de D est par définition :

$$[\![D]\!] := \forall \overrightarrow{u}, H \Rightarrow \bigvee_{i \in 1 \dots n} \exists \overrightarrow{e_i}, C_i$$

Par exemple, voici les diagrammes qui expriment les deux cas possibles de construction pour les réductions $\xrightarrow{+}$ et $\xrightarrow{*}$:



$$\forall xy, x \xrightarrow{+} y \Rightarrow (x \longrightarrow y \vee \exists y', x \longrightarrow y' \xrightarrow{+} y)$$



$$\forall xy, x \xrightarrow{*} y \Rightarrow (x \xrightarrow{=} y \vee \exists y', x \longrightarrow y' \xrightarrow{*} y)$$

[1] Les relations $\xrightarrow{+}$ et $\xrightarrow{*}$ seront définies dans la partie 6.2.4.

6.2.2 Langage des formules représentées

Après l'extension aux diagrammes disjonctifs, les formules qui peuvent être représentées par un diagramme sont exactement celles de la forme :

$$\forall \overline{u} \bigwedge_i H_i \Rightarrow \bigvee_i \exists \overline{e_i} \bigwedge_j C_{i_j}$$

Où les H_i et C_{i_j} sont des prédicats d'arité deux.

Ces énoncés forment un sous-langage de ce que Marc Bezem et Thierry Coquand appellent la *logique cohérente* (coherent logic). Pour plus d'information à propos de cette logique voir [BC05, BC04b].

Dans la suite nous appellerons $\mathcal{D}$ cette classe de formules.

6.2.3 A propos de la négation

La classe $\mathcal{D}$ de formules que nous venons de définir ne contient pas de négations. Cela représente une réelle limitation puisque par exemple, nous ne pouvons donc pas définir la notion de forme normale en réécriture. Mais cette propriété est très importante car les diagrammes que nous utilisons sont basés sur la représentation d'un exemple qui a valeur générale. Il est difficile de dénoter diagrammatiquement, par un exemple, le fait que quelquechose n'est pas. De même en géométrie, les figures impossibles ne sont pas représentables par un exemple. Toutefois, dans certains domaines, la négation (appliquée uniquement sur des atomes) peut être représentée diagrammatiquement. Elle est parfois symbolisée par la notion de «complémentarité». Par exemple, la non appartenance à un ensemble peut-être représentée dans le cadre des diagrammes d'Euler. Il faut noter que dans ce contexte la négation n'a alors pas la même signification, puisque cette notation implique implicitement que l'on travaille dans une logique classique : si l'élément n'est pas dans A alors il est dans son complémentaire $\neg A$.

6.2.4 Définitions et propriétés usuelles

Nous donnons maintenant quelques définitions en utilisant les diagrammes que nous avons définis. Celles-ci seront utiles aux exemples qui illustrent la partie suivante.

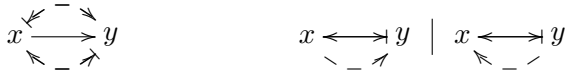
A chaque relation on associe quatre relations :

- la clôture réflexive ($\xrightarrow{=?}$),
- la clôture transitive ($\xrightarrow{+}$),
- la clôture réflexive et transitive ($\xrightarrow{*}$),
- la clôture symétrique ($\leftrightarrow\!\!\!|$).

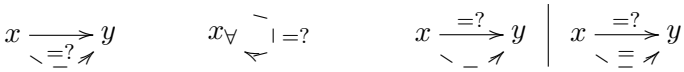
Les définitions des trois premières sont usuelles. Mais pour la définition de la clôture symétrique nous n'utilisons pas le symbole habituel ($\leftrightarrow$). En effet le symbole a la propriété qu'il dénote : il est symétrique ! C'est une

des raisons qui font que cette représentation est vraiment diagrammatique. Mais en toute rigueur, nous devons garder une notation asymétrique pour la définir. Nous verrons que dans les preuves diagrammatiques, la notation symétrique cache en réalité une étape de raisonnement implicite, nous verrons comment les traiter dans la partie 6.6.2.

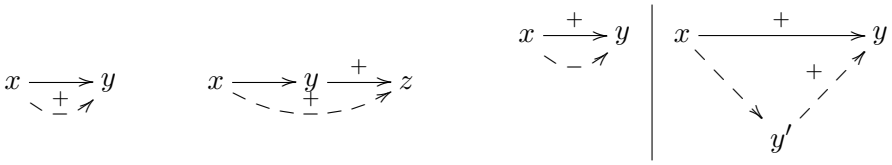
Définition 22 (Clôture symétrique). *La clôture symétrique d'une relation est définie par les deux diagrammes suivants :*



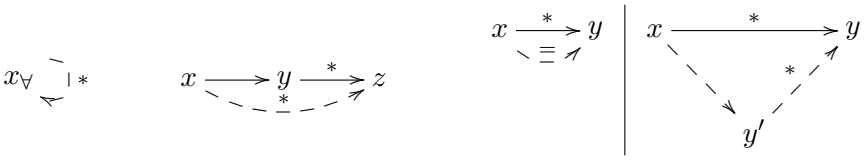
Définition 23 (Clôture réflexive). *La clôture réflexive d'une relation est définie par les trois diagrammes suivants :*



Définition 24 (Clôture transitive). *La clôture transitive d'une relation est définie*[2] *par les trois diagrammes suivants :*



Définition 25 (Clôture transitive et réflexive). *La clôture transitive et réflexive d'une relation est définie par les trois diagrammes suivants :*



Définition 26 (Vocabulaire).

On dit que x est réductible si :

$$\underline{x} - - \succ y$$

On dit que y est le successeur direct de x si :

$$\underline{x} - - \succ \underline{y} \; noté \; aussi \; x \longrightarrow y$$

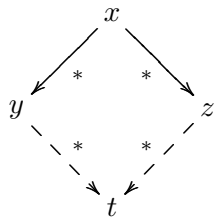
[2] Les clôtures transitive et transitive-réflexive n'étant pas définissables au premier ordre, cette définition n'est pas complète. Elle le sera uniquement quand nous aurons ajouté les principes d'induction associés à ces définitions dans la partie 6.5.

On dit que y est un successeur de x si :

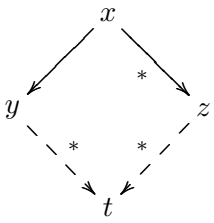
$$\underline{x} - \overset{+}{-} \succ \underline{y} \text{ noté aussi } x \xrightarrow{+} y$$

On dit que x et y sont joignables si :

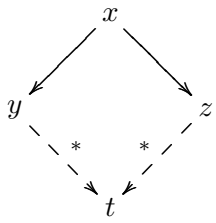
Définition 27 (Propriétés de confluence).



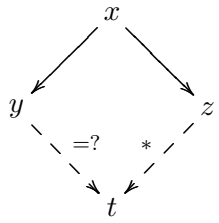
Confluence



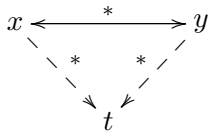
Semi-confluence



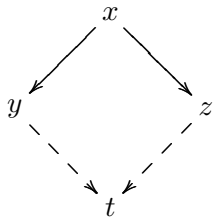
Confluence locale



Confluence forte

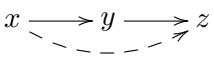


Church-Rosser



Propriété du diamant

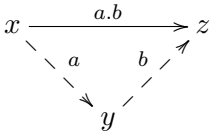
Définition 28 (Transitivité).
Une relation $\longrightarrow$ est transitive si elle vérifie le diagramme suivant :

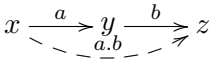


Définition 29 (Réflexivité).
Une relation $\longrightarrow$ est réflexive si elle vérifie le diagramme suivant :

Définition 30 (Composition).

La composition de deux relations $\overset{a}{\longrightarrow}$ et $\overset{b}{\longrightarrow}$ est définie par les diagrammes suivants :

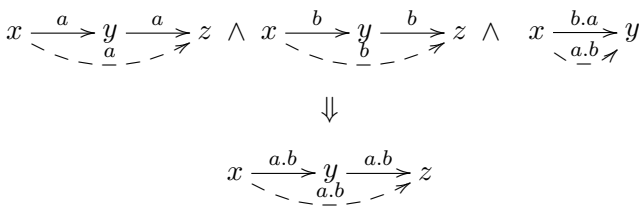




6.3 Preuves diagrammatiques

Précédemment nous avons défini formellement des diagrammes qui représentent des formules en réécriture. Mais des diagrammes du même genre sont aussi utilisés pour représenter des preuves. Avant de donner une définition formelle nous allons étudier une preuve très simple présentée en utilisant un diagramme (pour l'instant informel).

Exemple. *Si* $\xrightarrow{a}$ *et* $\xrightarrow{b}$ *sont deux relations qui sont transitives et* $\xrightarrow{b.a} \subseteq \xrightarrow{a.b}$ *alors* $\xrightarrow{a.b}$ *est transitive.*



Preuve traditionnelle

Soient x, y et z tels que $x \xrightarrow{a.b} y$ et $y \xrightarrow{a.b} z$.

Nous devons montrer que $x \xrightarrow{a.b} z$.

Par définition de $\xrightarrow{a.b}$ il existe u et v tels que $x \xrightarrow{a} u \xrightarrow{b} y$ et $y \xrightarrow{a} v \xrightarrow{b} z$.

Par définition de $\xrightarrow{b.a}$, on a $u \xrightarrow{b.a} v$.

Comme $\xrightarrow{b.a} \subseteq \xrightarrow{a.b}$, on a $u \xrightarrow{a.b} v$.

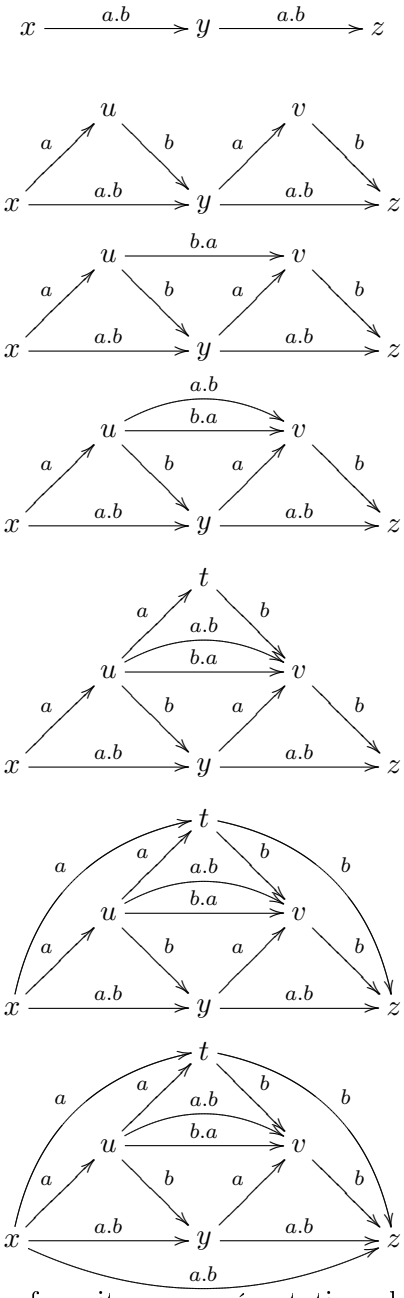
Par définition de $\xrightarrow{a.b}$, il existe t tel que $u \xrightarrow{a} t$ et $t \xrightarrow{b} v$.

Comme $\xrightarrow{a}$ et $\xrightarrow{b}$ sont transitives nous savons que $x \xrightarrow{a} t$ et $t \xrightarrow{b} z$.

Nous pouvons en conclure que :

$$x \xrightarrow{a.b} z$$

Preuve diagrammatique



Le diagramme qui figure sur la droite fournit une représentation claire de la preuve. Notons qu'il est nécessaire de fournir une «animation» de la manière dont le diagramme a été construit, une preuve consiste à mettre en évidence le fait qu'un diagramme peut être construit en utilisant certaines

règles. Ce diagramme représente ce que l'on sait tout au long de la preuve.

Notre but ici est de formaliser ce genre de preuve diagrammatique. Nous allons définir quelques règles qui formeront un petit système formel pour faire des preuves en utilisant des diagrammes. Nous voulons définir des règles qui imitent précisément les mêmes étapes de raisonnement que celles que l'on réalise en construisant des diagrammes comme celui donné en exemple. C'est pour cela que les règles que l'on va définir ne sont pas atomiques, elles pourraient être décomposées en des règles logiques plus simples.

Nous choisissons de définir un système formel dans le style du raisonnement vers l'avant. Cela signifie que les théorèmes seront prouvés pas à pas en partant des hypothèses (et non pas en transformant la conclusion). Nous verrons que ce choix permet d'envisager une implantation claire sous forme d'une interface graphique. En effet l'absence de règles de raisonnement qui transforment la conclusion permet d'envisager une implantation où seules les hypothèses sont représentées, la conclusion pouvant rester implicite pendant le processus de preuve.

Le raisonnement est formalisé de manière classique. Nous admettons que nous avons un ensemble d'hypothèses et un but. Les hypothèses et le but sont ici des diagrammes. De plus nous distinguons une hypothèse des autres, cette hypothèse sera appelée *factuelle*, les autres seront appelées *universelles*. L'hypothèse factuelle représente ce que l'on sait pendant la preuve, et les hypothèses universelles forment la «boîte à outils» pour prouver le théorème.

Définition 31 (hypothèse factuelle)**.** *Nous appelons* hypothèse factuelle*, un diagramme qui contient uniquement des sommets libres et des flèches conclusion.*

Remarque 4. *Notons que grâce aux notations que nous avons définies, les diagrammes factuels peuvent en fait être représentés uniquement avec des flèches pleines.*

Définition 32 (hypothèse universelle)**.** *Nous appelons* hypothèse universelle*, un diagramme qui n'est pas factuel.*

Cela signifie que nous avons des « séquents » de la forme suivante :

$$U_1, U_2, \ldots U_n, F \vdash D$$

où $U_1, \ldots U_n$ sont des diagrammes universels et F est un diagramme factuel. Les règles d'inférence transforment un séquent en un autre.

Pour décrire ces règles nous avons besoin de définir quelques transformations qui opèrent sur les diagrammes.

Définition 33 (inversion)**.** *Étant donné un diagramme D l'inversion de D est par définition D où chaque flèche hypothèse a été transformée en une*

flèche conclusion.
Formellement si $D = (\Sigma_V, \Sigma_A, V, A, f, l_A, l_V, s_A, s_V)$ *alors*

$$\mathcal{I}(D) = (\Sigma_V, \Sigma_A, V, A, f, l_A, l_V, s'_A, s_V)$$

où $s'(a) = \begin{cases} \mathcal{C} & si\ s_A(a) = \mathcal{H}, \\ s_A(a) & sinon \end{cases}$

Définition 34 (union)**.** *Nous ne définissons l'union que pour les diagrammes factuels.*
On dit que D *est l'union de deux diagrammes factuels* D_1 *et* D_2*, noté* $D_1 \cup D_2$*, ssi le graphe sous-jacent à* D *est l'union des deux graphes sous-jacents à* D_1 *et* D_2 *et tous les sommets sont libres et les flèches sont des flèches conclusion.*

Définition 35 (sous-diagramme)**.**
On dit qu'un diagramme

$$D_1 = (\Sigma_{V_1}, \Sigma_{A_1}, V_1, A_1, s_1, d_1, l_{A_1}, l_{V_1}, s_{A_1}, s_{V_1})$$

est un sous-diagramme de

$$D_2 = (\Sigma_{V_2}, \Sigma_{A_2}, V_2, A_2, s_2, d_2, l_{A_2}, l_{V_2}, s_{A_2}, s_{V_2})$$

noté $D_1 \subseteq D_2$ *ssi :*
- $V_1 \subseteq V_2$
- $A_1 \subseteq A_2$
- *les fonctions* $s_1, d_1, l_{A_1}, l_{V_1}, s_{A_1}, s_{V_1}$ *et* $s_2, d_2, l_{A_2}, l_{V_2}, s_{A_2}, s_{V_2}$ *coïncident deux à deux sur les points où elles sont toutes les deux définies.*

Notations : Nous notons D_H (resp. D_C) le sous-diagramme de D qui ne contient que les flèches hypothèses (resp. conclusions).

6.3.1 Règles d'inférence

Notre système comprend six règles d'inférence :

`intros` permet d'introduire les hypothèses dans le contexte,

`apply` permet d'utiliser l'information contenue dans un diagramme universel pour enrichir le diagramme factuel,

`conclusion` est une règle axiome, elle permet de conclure lorsque le diagramme factuel contient assez d'informations,

`substitute` et `reflexivity` permettent de traiter l'égalité,

`cut` permet de réutiliser des résultats prouvés précédement.

Notons que nous choisissons de définir l'égalité comme une notion primitive. Nous aurions pu la définir au moyen de diagrammes. Mais cette approche aurait augmenté la taille des diagrammes en interdisant de simplifier les diagrammes où plusieurs sommets sont égaux.

`intros`

La première règle est la règle `intros`, elle a été omise dans l'exemple informel que nous avons donné.
Soit $\overline{f}$ l'ensemble des étiquettes des sommets libres de $H_1, \ldots, H_n, G$.
Soit $G_{hyp} = \sigma(\mathcal{I}(G_H))$ et $G_{concl} = \sigma(G_C)$, avec σ une substitution d'une partie des sommets universels de G en des sommets libres étiquettés par des labels frais.

$$\texttt{intros}\ \frac{H_1, \ldots, H_n, G_{hyp} \vdash G_{concl}}{H_1, \ldots, H_n \vdash G}$$

Notons qu'en utilisant la deuxième notation (N2), cela signifie que graphiquement G_{hyp} est représenté par le sous-diagramme de G restreint aux flèches pleines.

Exemple.

$$\textit{intros}\ \frac{x \xrightarrow{a.b} y \xrightarrow{a.b} z \ \vdash\ x \xrightarrow{a.b} z}{\vdash\ \begin{tikzcd} x \arrow[r, "a.b"] \arrow[rr, bend right, dashed, "a.b"'] & y \arrow[r, "a.b"] & z \end{tikzcd}}$$

`apply`

La deuxième règle est la règle `apply`. C'est la règle qui est utilisée à chaque étape du premier exemple. Elle consiste à appliquer un diagramme universel D à un sous-diagramme du diagramme factuel F. Si D est un diagramme disjonctif cette règle introduit alors une distinction de cas.

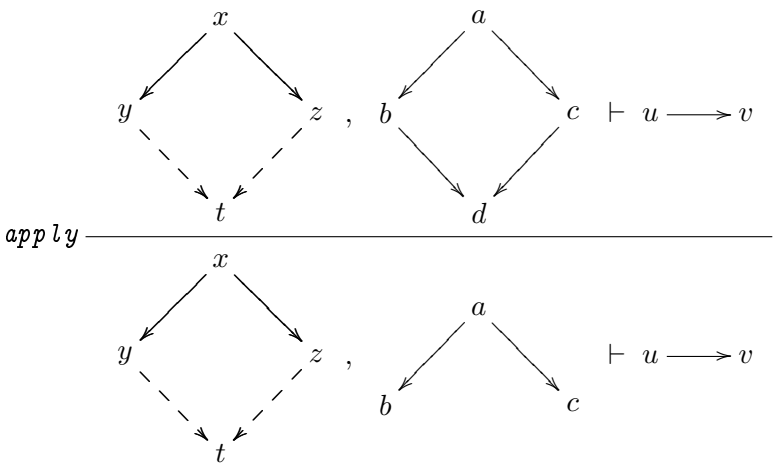
Étant donné un diagramme universel D dans l'ensemble des hypothèses et une substitution σ qui substitue les sommets universels de telle manière que les hypothèses forment un sous-diagramme du diagramme factuel, pour chaque diagramme D_j formant le diagramme disjonctif, la règle `apply` impose de prouver le théorème avec le diagramme factuel enrichi par la conclusion de D_i, les sommets existentiels étant instanciés par des variables fraîches.

Formellement :

$$\texttt{apply}\ \frac{D_1, \ldots, D_n, F \cup \delta_1(F_1) \vdash G \qquad \ldots \qquad D_1, \ldots, D_n, F \cup \delta_m(F_m) \vdash G}{D_1, \ldots, D_n, F \vdash G}$$

si $\exists i, \sigma, \mathcal{I}(\sigma(D_i)_H) \subseteq F$
et $(\sigma(D_i))_C = (F_1 | \ldots | F_m)$
et $\delta_1, \ldots, \delta_m$ associent aux sommets existentiels de $F_1, \ldots, F_m$ des variables fraîches.

Exemple.



`substitute`

Si le diagramme factuel contient un sous-diagramme de la forme $\underline{x} \xrightarrow{=} \underline{y}$ la règle `substitute` permet de remplacer certaines occurrences de $\underline{x}$ par $\underline{y}$ et/ou de fusionner les sommets $\underline{x}$ et $\underline{y}$ dans tous les diagrammes.

Exemple.

$$\textit{substitute}\ \frac{a \longrightarrow x \vdash \quad \circlearrowright x \longrightarrow z}{a \longrightarrow x \xrightarrow{=} y \vdash\ x \longrightarrow y \longrightarrow z}$$

`reflexivity`

La règle de réflexivité de l'égalité est la suivante :

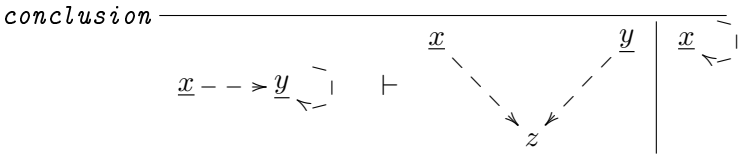
$$\texttt{reflexivity}\ \frac{}{\Gamma \vdash x = x}$$

`conclusion`

La règle conclusion est utilisée pour finir la preuve. Si le but est un diagramme $G = G_1 | \ldots | G_m$ sans flèche hypothèse ni sommet universel (avec $m = 1$ si G n'est pas disjonctif), la règle `conclusion` prouve le théorème s'il existe un diagramme G_i et une substitution σ des sommets existentiels de G_i tels que $\sigma(G_i)$ est un sous-diagramme de l'hypothèse factuelle F.

$$\texttt{conclusion}\ \frac{}{D_1, \ldots, D_n, F \vdash G_1 | \ldots | G_m}\ \text{si}\ \exists i \sigma, \sigma(G_i) \subseteq F$$

Exemple.



`cut`

La règle `cut` est la règle de coupure usuelle.

$$\texttt{cut}\ \frac{D_1, \ldots, D_n, F \vdash G \qquad D_1, \ldots, D_n, G, F \vdash J}{D_1, \ldots, D_n, F \vdash J}$$

6.4 Correction et complétude du système

Dans cette partie, nous prouvons la correction et la complétude du système formel proposé vis à vis d'un calcul des séquents enrichi d'une notion d'égalité.

6.4.1 Logique intuitionniste vs classique

Avant de prouver la correction et la complétude de notre système, il faut choisir un système de référence. La question se pose alors de choisir entre un système en logique intuitionniste ou classique. Mais il se trouve en fait que pour la classe de formules que nous considérons si une formule est prouvable dans le calcul des séquents classique alors elle est prouvable dans le calcul des séquents intuitionniste. Ce résultat a été prouvé plusieurs fois [BC04b, Neg03], nous montrons ici que l'on peut se ramener facilement à un résultat de Gopalan Nadathur [Nad00] au moyen du lemme de permutation des règles de Kleene [Kle52].

Dans cette partie nous notons $\vdash_{LJ}$ la prouvabilité dans la calcul des séquents intuitionniste et $\vdash_{LK}$ dans le calcul des séquents classique[3]. Puisque ces deux notions coïncident, pour la classe de formules considérée, nous omettrons de distinguer les deux dans les parties suivantes.

Lemme 1 (Kleene). *Si $\Gamma \vdash_{LK} A, \Delta$ alors on peut construire des preuves des séquents suivants :*

- *si A est de la forme $P \Rightarrow Q$ alors $\Gamma, P \vdash_{LK} Q, \Delta$*
- *si A est de la forme $\forall x\, P$ alors $\Gamma \vdash_{LK} [c/x]P, \Delta$ avec c une variable fraîche.*

Preuve: La preuve de ce lemme figure dans [Kle52] et sous une forme généralisée à la déduction modulo dans [Her05]. □

Théorème 9 (Nadathur).
Considérons les classes suivantes de H et G-formules, en supposant que A représente des formules atomiques.

$$G ::= \top | \bot | A | G \wedge G | G \vee G | \exists x G$$

$$H ::= \top | \bot | A | G \Rightarrow H | H \wedge H | H \vee H | \exists x H | \forall x H$$

Si Γ est un multi-ensemble composé de H-formules, et F est une G-formule alors

$$\Gamma \vdash_{LK} F \iff \Gamma \vdash_{LJ} F$$

Preuve: Voir [Nad00], Théorème 6. □

[3] Notons que en réalité nous adoptons une présentation du type G_3, le traitement des règles structurelles n'étant pas notre propos.

Théorème 10. *Si $D_1, \ldots, D_n$ et G sont dans $\mathcal{D}$ alors*

$$D_1, \ldots, D_n \vdash_{LK} G \iff D_1, \ldots, D_n \vdash_{LJ} G$$

Preuve:
L'implication de droite à gauche est toujours vraie.
Il suffit donc de prouver que $D_1, \ldots, D_n \vdash_{LK} G \Rightarrow D_1, \ldots, D_n \vdash_{LJ} G$.
Supposons que $D_1, \ldots, D_n \vdash_{LK} G$.
Comme $G \in \mathcal{D}$, G est de la forme :

$$\forall\, \overline{u} \bigwedge_i H_i \Rightarrow \bigvee_i \exists\, \overline{e_i} \bigwedge_j C_{i_j}$$

Où les H_i et C_{i_j} sont des prédicats d'arité deux.
D'après le lemme de Kleene appliqué à $\forall$ et $\Rightarrow$, on peut construire une preuve de :

$$D_1, \ldots, D_n, [\overline{c}/\overline{u}] \bigwedge_i H_i \vdash_{LK} [\overline{c}/\overline{u}] \bigvee_i \exists\, \overline{e_i} \bigwedge_j C_{i_j}$$

où $\overline{c}$ sont des variables fraîches.
D'après le théorème de Nadathur, on a aussi :

$$D_1, \ldots, D_n, [\overline{c}/\overline{u}] \bigwedge_i H_i \vdash_{LJ} [\overline{c}/\overline{u}] \bigvee_i \exists\, \overline{e_i} \bigwedge_j C_{i_j}$$

Et par applications successives des règles $\forall_{\mathcal{R}}$ et $\Rightarrow_{\mathcal{R}}$, on a $D_1, \ldots, D_n, \vdash_{LJ} G$.
□

6.4.2 Calcul considéré

Nous définissons ici le système formel qui nous sert de référence pour les preuves de correction et de complétude en se basant sur le calcul des séquents. La classe de formules considérée $\mathcal{D}$ ne contient pas de négation, nous omettons donc les règles associées. De plus notre système permet de manipuler l'égalité de manière primitive, nous rajoutons donc aussi les règles analogues dans le calcul des séquents. Le système obtenu est donné sur le tableau 6.1. Nous notons $\vdash_=$ la prouvabilité dans ce calcul, $\vdash$ la prouvabilité dans ce calcul sans utiliser les règles E_1, E_2, $=_{\mathcal{R}}$. Enfin nous notons $\vdash_{\mathcal{D}}$ la prouvabilité dans le calcul diagrammatique défini en 6.3.1.

6.4.3 Correction

Dans cette partie nous prouvons la correction du système proposé. La correction ne pose pas de problème puisque chacune des règles d'inférence diagrammatiques correspondent en fait à une suite de règles dans le calcul des séquents. Seule la règle de `substitute` fait exception. Elle nécessite le lemme suivant :

TAB. 6.1 – Calcul des séquents classique sans négation

$$\text{axiome}\ \frac{}{\Gamma, A \vdash \Delta, A}$$

$$\Rightarrow_{\mathcal{L}} \frac{\Gamma \vdash A, \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \Rightarrow B \vdash \Delta} \qquad \Rightarrow_{\mathcal{R}} \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \Rightarrow B, \Delta}$$

$$\wedge_{\mathcal{L}} \frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \qquad \wedge_{\mathcal{R}} \frac{\Gamma \vdash \Delta, A \qquad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B}$$

$$\vee_{\mathcal{L}} \frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}$$

$$\vee_{\mathcal{R}} \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}$$

$$\forall_{\mathcal{L}} \frac{\Gamma, \forall x B, B[x \leftarrow t] \vdash \Delta}{\Gamma, \forall x B \vdash \Delta} \qquad \forall_{\mathcal{R}} \frac{\Gamma \vdash B[x \leftarrow c], \Delta}{\Gamma \vdash \forall x B, \Delta}$$

$$\exists_{\mathcal{L}} \frac{\Gamma, B[x \leftarrow c] \vdash \Delta}{\Gamma, \exists x B \vdash \Delta} \qquad \exists_{\mathcal{R}} \frac{\Gamma \vdash \exists x B, B[x \leftarrow t], \Delta}{\Gamma \vdash \exists x B, \Delta}$$

$$=_{\mathcal{R}} \frac{}{\Gamma \vdash s = s, \Delta}$$

$$E_1 \frac{\Gamma, s = t \vdash [s/x]\Delta}{\Gamma, s = t \vdash [t/x]\Delta} \qquad E_2 \frac{\Gamma, s = t \vdash [t/x]\Delta}{\Gamma, s = t \vdash [s/x]\Delta}$$

dans $\exists_{\mathcal{L}}$, c n'apparaît pas libre dans $\exists x\, B, \Gamma, \Delta$
dans $\forall_{\mathcal{R}}$, c n'apparaît pas libre dans $\forall x\, B, \Gamma, \Delta$

Lemme 2. *Les règles de substitution généralisée*

$$GE_1\ \frac{[s/x]\Gamma, s = t \vdash [s/x]\Delta}{[t/x]\Gamma, s = t \vdash [t/x]\Delta} \qquad GE_2\ \frac{[t/x]\Gamma, s = t \vdash [t/x]\Delta}{[s/x]\Gamma, s = t \vdash [s/x]\Delta}$$

sont admissibles.

Preuve: Par induction sur la structure des dérivations. □

Théorème 11 (Correction)**.**
Si $D_1, \ldots, D_n, F \vdash_{\mathcal{D}} G$ *alors* $[\![D_1]\!], \ldots, [\![D_n]\!], [\![F]\!] \vdash_{=} G$.

Preuve: Par induction sur la structure de la preuve et par cas sur la règle utilisée :

`intros` *Par application des règles* $\forall_{\mathcal{R}}$, $\Rightarrow_{\mathcal{R}}$, $\wedge_{\mathcal{L}}$.

`apply` *Par application des règles* $\forall_{\mathcal{L}}$, $\Rightarrow_{\mathcal{L}}$ *puis* $\wedge_{\mathcal{R}}$,$\wedge_{\mathcal{L}}$, *axiome d'une part et* $\vee_{\mathcal{L}}$, $\exists_{\mathcal{L}}$, $\wedge_{\mathcal{L}}$, *axiome d'autre part.*

`conclusion` *Par application des règles* $\vee_{\mathcal{R}}$, $\exists_{\mathcal{R}}$, $\wedge_{\mathcal{R}}$, $\wedge_{\mathcal{L}}$, *axiome.*

`substitute` *Par application de* GE_1 *et* GE_2.

`reflexivity` *Par application de* $=_{\mathcal{R}}$.

`cut` *La règle de coupure du calcul des séquents étant admissible, elle a été omis de notre présentation, nous l'utilisons ici.*

□

6.4.4 Complétude

Il est possible de séparer complètement le raisonnement à propos de l'égalité du reste de la preuve. Grâce à cette possibilité nous pouvons réutiliser des résultats connus à propos du raisonnement sans égalité. Pour la complétude du système sans égalité nous nous ramenons à un résultat de Marc Bezem et Thierry Coquand. Bien que développées séparement et dans un but différent[4], nos règles d'inférence correspondent précisement à la définition 6.1 de [BC04b]. Notons que le calcul des séquents que nous utilisons ici n'est pas défini de la même manière que dans [BC04b] (par exemple la règle du $\vee$ multiplicative), mais puisqu'ils sont équivalents, nous ne les distinguerons pas. Nous étendons ensuite le résultat pour obtenir la complétude du système complet.

Système sans égalité

Théorème 12 (Complétude partielle)**.**
Si $D_1, \ldots, D_n, F$ *et* G *sont dans* $\mathcal{D}$ *et* $D_1, \ldots, D_n, F \vdash G$ *alors il existe des diagrammes* $D'_1, \ldots, D'_n$, F' *et* G' *tels que :*

[4] Marc Bezem et Thierry Coquand s'intéressent à l'*automatisation* de la géométrie cohérente.

$$[\![D'_1]\!] = D_1, \ldots, [\![D'_n]\!] = D_n,\ [\![F']\!] = F \text{ et } [\![G']\!] = G \text{ et}$$
$$D'_1, \ldots, D'_n, F' \vdash_{\mathcal{D}} G'$$

Preuve: Comme G est dans $\mathcal{D}$, G est de la forme $\forall \overline{u}, C \Rightarrow D$. Donc d'après le lemme de Kleene, on peut construire une preuve de

$$D_1, \ldots, D_n, F, [\overline{c}/\overline{u}]C \vdash D.$$

D'après le théorème 6.2 de [BC04b] avec pour tout X, X' étant n'importe quel diagramme tel que $[\![X']\!] = X$, on a

$$D'_1, \ldots, D'_n, F', [\overline{c}/\overline{u}]C' \vdash_{\mathcal{D}} D'.$$

(Le cas de base de la définition 6.1 de [BC04b] correspond à notre règle `conclusion` *et le cas d'induction correspond à notre règle* `apply`*.)*
Grâce à la règle `intros` *nous pouvons en conclure que :*

$$D'_1, \ldots, D'_n, F' \vdash_{\mathcal{D}} G'$$

□

Traitement de l'égalité

Dans cette partie nous montrons la complétude du système avec égalité. Afin d'utiliser le résultat de complétude partielle du système sans égalité nous utilisons le fait que le raisonnement lié à l'égalité peut-être repoussé sur les feuilles de l'arbre de dérivation. En d'autres termes si $\Gamma \vdash_= \Delta$ alors $\Gamma \vdash_{|=} \Delta$, le système $\vdash_{|=}$ étant défini sur le tableau 6.2. Le système $\vdash_{|=}$ correspond au système $\vdash_=$ où l'on a supprimé les règles concernant l'égalité, et remplacé la règle axiome par un mini système avec les règles concernant l'égalité.

Lemme 3. $\Gamma \vdash_= \Delta \iff \Gamma \vdash_{|=} \Delta$

Preuve: Voir [Pfe04]. □

Lemme 4. *Si $\Gamma \vdash_{|=} \Delta$ alors il existe Γ' un multi ensemble de formules qui appartiennent à la logique cohérente tel que $\Gamma', \Gamma \vdash \Delta$ et pour tout X dans Γ' il existe X' tel que $[\![X']\!] = X$ et $\vdash_{\mathcal{D}} X'$.*

Preuve: Soient Γ_i et Δ_i respectivement les hypothèses et conclusions des prémisses des règles eq-axiome.
On définit Γ' comme l'union des :

$$\Gamma'_i \Rightarrow \Delta'_i$$

où Γ'_i est la conjonction des atomes de Γ_i et Δ'_i la disjonction des formules de Δ_i. Notons que comme la règle axiome$_=$ est restreinte aux atomes, les

$$axiome_{=}\ \frac{}{\Gamma, A \vdash_{Ax_=} A} \qquad =_{\mathcal{R}}\ \frac{}{\Gamma \vdash_{Ax_=} x = x}$$

$$E_1\ \frac{\Gamma, s = t \vdash_{Ax_=} [s/x]\Delta}{\Gamma, s = t \vdash_{Ax_=} [t/x]\Delta} \qquad E_2\ \frac{\Gamma, s = t \vdash_{Ax_=} [t/x]\Delta}{\Gamma, s = t \vdash_{Ax_=} [s/x]\Delta}$$

$$\text{eq-axiome}\ \frac{\Gamma \vdash_{Ax_=} \Delta}{\Gamma \vdash_{|=} \Delta}$$

$$\Rightarrow_{\mathcal{L}}\ \frac{\Gamma \vdash_{|=} A, \Delta \qquad \Gamma, B \vdash_{|=} \Delta}{\Gamma, A \Rightarrow B \vdash_{|=} \Delta} \qquad \Rightarrow_{\mathcal{R}}\ \frac{\Gamma, A \vdash_{|=} B, \Delta}{\Gamma \vdash_{|=} A \Rightarrow B, \Delta}$$

$$\wedge_{\mathcal{L}}\ \frac{\Gamma, A, B \vdash_{|=} \Delta}{\Gamma, A \wedge B \vdash_{|=} \Delta} \qquad \wedge_{\mathcal{R}}\ \frac{\Gamma \vdash_{|=} \Delta, A \qquad \Gamma \vdash_{|=} \Delta, B}{\Gamma \vdash_{|=} \Delta, A \wedge B}$$

$$\vee_{\mathcal{L}}\ \frac{\Gamma, A \vdash_{|=} \Delta \qquad \Gamma, B \vdash_{|=} \Delta}{\Gamma, A \vee B \vdash_{|=} \Delta}$$

$$\vee_{\mathcal{R}}\ \frac{\Gamma \vdash_{|=} A, B, \Delta}{\Gamma \vdash_{|=} A \vee B, \Delta}$$

$$\forall_{\mathcal{L}}\ \frac{\Gamma, \forall x B, B[x \leftarrow t] \vdash_{|=} \Delta}{\Gamma, \forall x B \vdash_{|=} \Delta} \qquad \forall_{\mathcal{R}}\ \frac{\Gamma \vdash_{|=} B[x \leftarrow c], \Delta}{\Gamma \vdash_{|=} \forall x B, \Delta}$$

$$\exists_{\mathcal{L}}\ \frac{\Gamma, B[x \leftarrow c] \vdash_{|=} \Delta}{\Gamma, \exists x B \vdash_{|=} \Delta} \qquad \exists_{\mathcal{R}}\ \frac{\Gamma \vdash_{|=} \exists x B, B[x \leftarrow t], \Delta}{\Gamma \vdash_{|=} \exists x B, \Delta}$$

dans $\exists_{\mathcal{L}}$, c n'apparaît pas libre dans $\exists x\, B, \Gamma, \Delta$
dans $\forall_{\mathcal{R}}$, c n'apparaît pas libre dans $\forall x\, B, \Gamma, \Delta$
dans $axiome_{=}$, A est un atome

TAB. 6.2 – Le système $\vdash_{|=}$.

éléments de Δ_i sont des atomes. Les éléments de Γ' appartiennent donc à l'ensemble des formules représentables par un diagramme.
On obtient le résultat pour la règle axiome$_=$ grâce aux règles `intros`, `apply` *et* `conclusion`. *Pour les autres règles (E_1,E_2 et $=_{\mathcal{R}}$) on utilise* `substitute` *et* `reflexivity`. □

Théorème 13 (Complétude)**.**
Si $D_1, \ldots, D_n, F \vdash_= G$ alors il existe des diagrammes D'_1,...,D'_n, F' et G' tels que :

$$[\![D'_1]\!] = D_1, \ldots, [\![D'_n]\!] = D_n,\ [\![F']\!] = F \text{ et } [\![G']\!] = G \text{ et}$$

$$D'_1, \ldots, D'_n, F' \vdash_{\mathcal{D}} G'$$

Preuve: Supposons que $D_1, \ldots, D_n, F \vdash_= G$ alors d'après le lemme 3 on sait que $D_1, \ldots, D_n, F \vdash_{|=} G$.
D'après le lemme 4 il existe Γ tel que $\Gamma, D_1, \ldots, D_n, F \vdash G$ et pour tout X dans Γ il existe un diagramme X' tel que $[\![X']\!] = X$ et $\vdash_{\mathcal{D}} X'$.
D'après la complétude du système sans égalité, on peut en déduire qu'il existe Γ', D'_1, ..., D'_n et G' tels que

$$\Gamma', D'_1, \ldots, D'_n, F' \vdash_{\mathcal{D}} G'$$

comme les diagrammes de Γ' sont prouvables dans le contexte vide, en utilisant la règle `cut` *on a*

$$D'_1, \ldots, D'_n, F' \vdash_{\mathcal{D}} G'$$

□

6.5 Extension aux preuves par induction

Dans cette partie nous étendons notre calcul avec des règles permettant de faire des preuves par induction. Nous considérons l'induction sur la longueur des dérivations et l'induction bien fondée.

6.5.1 L'induction classique

La règle d'induction sur la longueur d'une dérivation $\xrightarrow{*}$ est la suivante : Si on veut prouver $\forall xy, P(x,y)$ avec $x \xrightarrow{*} y$ il suffit de montrer $P(x,x)$ et $P(x,y)$ sous l'hypothèse qu'il existe y' tel que $x \longrightarrow y' \xrightarrow{*} y$ et $P(y',y)$ est vrai. La règle traditionnelle sur laquelle on s'appuie est la suivante :

$$ind_* \frac{\forall xy\ x = y \Rightarrow P(x,y) \qquad \forall xy'y\ x \longrightarrow y' \xrightarrow{*} y \wedge P(y',y) \Rightarrow P(x,y)}{\forall xy\ x \xrightarrow{*} y \Rightarrow P(x,y)}$$

Diagrammatiquement, on utilise la règle suivante :
Soit G un diagramme avec deux sommets universels x et y tels que $x \xrightarrow{*} y$.
Soit $G_=$ le même diagramme où d'une part les sommets x et y ont été remplacés par des sommets libres étiquettés par des variables fraîches et d'autre part l'arête $x \xrightarrow{*} y$ a été remplacée par $x = y$.
Soit G_{ind} le diagramme G où d'une part le sommet étiquetté par x est maintenant étiquetté par y' et d'autre part y' et y sont libres.
Soit G_H, le diagramme factuel $x \longrightarrow y' \xrightarrow{*} y$.
Soit G_+, le diagramme G où x et y sont libres.
Alors on a :

$$ind_* \frac{\Gamma \vdash G_= \qquad \Gamma, G_{ind}, G_H \vdash G_+}{\Gamma \vdash G}$$

Exemple. $\xrightarrow{*}$ *est transitive.*

Preuve:

$$\vdash\ x \xrightarrow{*} y \xrightarrow{*} z \quad (x \xrightarrow{*} z)$$

Cas 1 :

$$\vdash\ \underline{x} \xrightarrow{=} \underline{y} \xrightarrow{*} z \quad (\underline{x} \xrightarrow{*} z)$$

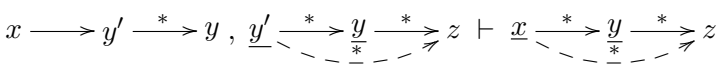
par la règle `intros`

$$x \xrightarrow{=} y \xrightarrow{*} z\ \vdash\ x \xrightarrow{*} z$$

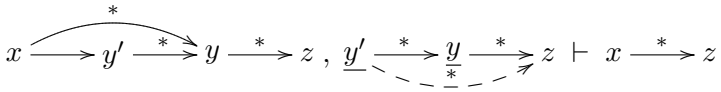
par `substitute`

$$x \xrightarrow{*} z\ \vdash\ x \xrightarrow{*} z$$

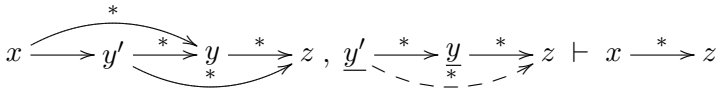
La règle `conclusion` *permet de conclure ce cas.*
Cas 2 :



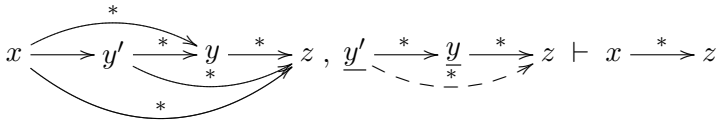
par `intros`



par `apply`



par `apply` *utilisée avec la définition de* $\xrightarrow{*}$



La règle `conclusion` *permet de conclure ce cas.* □

6.5.2 L'induction bien fondée

Dans cette partie nous ajoutons une règle pour pouvoir réaliser l'induction bien fondée. La règle d'induction dit que si une relation $\longrightarrow$ termine alors pour prouver que $\forall x\, P(x)$ il est suffisant de montrer que $P(x)$ est vrai en admettant que $P(y)$ est vrai pour n'importe quel y tel que $x \xrightarrow{+} y$. Formellement :

$$\frac{\forall x\ (\forall y\, x \xrightarrow{+} y \Rightarrow P(y)) \Rightarrow P(x)}{\forall x\, P(x)} \text{ si } \longrightarrow \text{ termine}$$

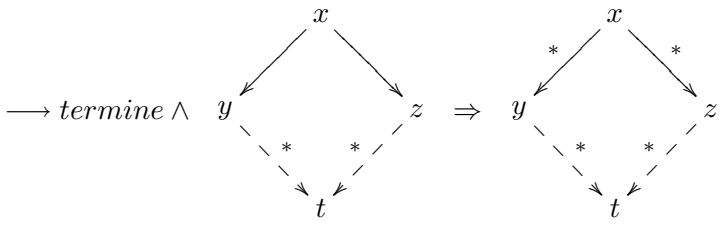
Nous pouvons formaliser cette règle d'inférence diagrammatiquement de la manière suivante :
Considérons un diagramme G, s'il contient au moins un sommet quantifié universellement et que nous savons qu'une réduction $\longrightarrow$ termine alors nous pouvons utiliser la règle d'induction bien fondée. La règle d'induction comporte deux arguments : le premier est la relation qui termine et le second est l'un des sommets universellement quantifiés du but (appelons le x). L'effet de la règle d'induction est de rajouter un diagramme correspondant à l'hypothèse d'induction H_i dans les hypothèses et de changer le but en un diagramme G'. Le diagramme d'hypothèse d'induction H_i est composé de G où x a été renommé par un symbole frais y et enrichi d'une nouvelle flèche :

$\underline{x} \xrightarrow{+} y$.
Le diagramme G' est exactement G excepté que le statut de x est maintenant $\mathcal{F}$.

$$\texttt{wf\_induction}\,\frac{D_1, \ldots, D_n, H_i \vdash G'}{D_1, \ldots, D_n, F \vdash G}$$

Nous étendons les hypothèses possibles avec un nouveau type d'hypothèse spécial qui exprime le fait qu'une relation termine.

Exemple (Lemme de Newman)**.**
Une relation qui termine est confluente si elle est localement confluente.

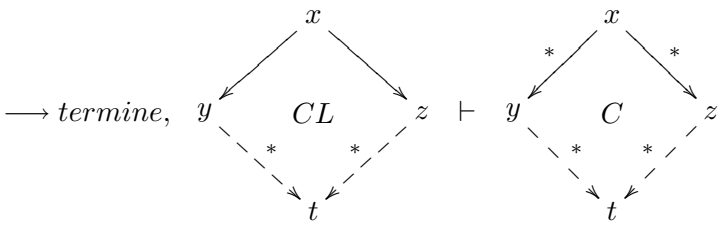


Preuve traditionnelle *(Gérard Huet [Hue80])*
Nous devons montrer que $\forall xyz, x \xrightarrow{*} y \wedge x \xrightarrow{*} z \Rightarrow \exists t, y \xrightarrow{*} t \wedge z \xrightarrow{*} t$.
Prouvons le théorème par induction bien fondée en utilisant le fait que $\longrightarrow$ *termine et le prédicat* $P(x) = \forall yz, x \xrightarrow{*} y \wedge x \xrightarrow{*} z \Rightarrow \exists t, y \xrightarrow{*} t \wedge z \xrightarrow{*} t$.
Si $x = y$ *le théorème est vérifié parce que* $x \xrightarrow{*} z$ *et* $z \xrightarrow{*} z$.
Si $x = z$ *le théorème est vérifié parce que* $x \xrightarrow{*} y$ *et* $y \xrightarrow{*} y$.
Sinon $x \neq y$ *et* $x \neq z$ *alors il existe* y' *et* z' *tels que* $x \longrightarrow y' \xrightarrow{*} y$ *et* $x \longrightarrow z' \xrightarrow{*} z$.
Par confluence locale nous savons qu'il existe t *tel que* $y' \xrightarrow{*} t$ *et* $z' \xrightarrow{*} t$.
D'après l'hypothèse d'induction et le fait que $x \xrightarrow{+} y'$ *nous savons qu'il existe* u *tel que* $y \xrightarrow{*} u$ *et* $t \xrightarrow{*} u$.
D'après l'hypothèse d'induction et le fait que $x \xrightarrow{+} z'$ *nous savons qu'il existe* v *tel que* $u \xrightarrow{*} v$ *et* $z \xrightarrow{*} v$.
De $y \xrightarrow{*} u$ *et* $u \xrightarrow{*} v$ *nous déduisons que* $y \xrightarrow{*} v$.

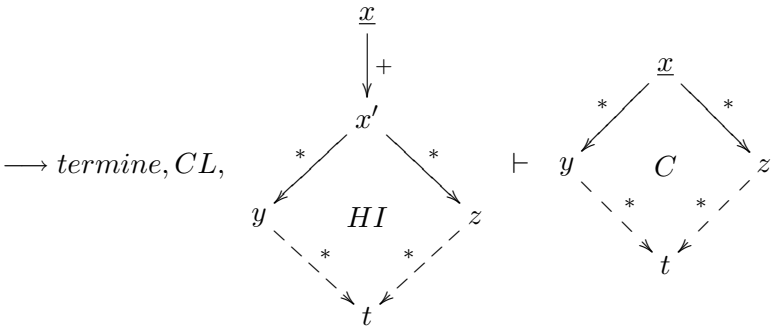
Preuve diagrammatique

Pour la clarté de la présentation nous omettons les diagrammes concernant les définitions de $\xrightarrow{+}$ *et* $\xrightarrow{*}$. *Nous admettons que le contexte contient aussi les diagrammes concernant la transitivité de* $\xrightarrow{*}$.

L'énoncé est le suivant :



par induction sur $\longrightarrow$



en utilisant la règle `intros` *:*

$$\longrightarrow termine, CL, HI,\ y \overset{*}{\longleftarrow} x \overset{*}{\longrightarrow} z \ \vdash\ \underline{y} - \overset{*}{-} \succ t \prec \overset{*}{-} - \underline{z}$$

par distinction de cas sur $x \xrightarrow{*} y$

Cas 1

$$\longrightarrow termine, CL, HI,\ y \overset{=}{\longleftarrow} x \overset{*}{\longrightarrow} z \ \vdash\ \underline{y} - \overset{*}{-} \succ t \prec \overset{*}{-} - \underline{z}$$

par `substitute`

$$\longrightarrow termine, CL, HI,\ x \overset{*}{\longrightarrow} z \ \vdash\ \underline{x} - \overset{*}{-} \succ t \prec \overset{*}{-} - \underline{z}$$

par `apply` *en utilisant la définition de* $\xrightarrow{*}$

$$\longrightarrow termine, CL, HI,\ x \overset{*}{\longrightarrow} z\ ,\ z \overset{*}{\longrightarrow} z \ \vdash\ x \overset{*}{\longrightarrow} z \overset{*}{\longleftarrow} z$$

La règle `conclusion` *permet de conclure.*

Cas 2

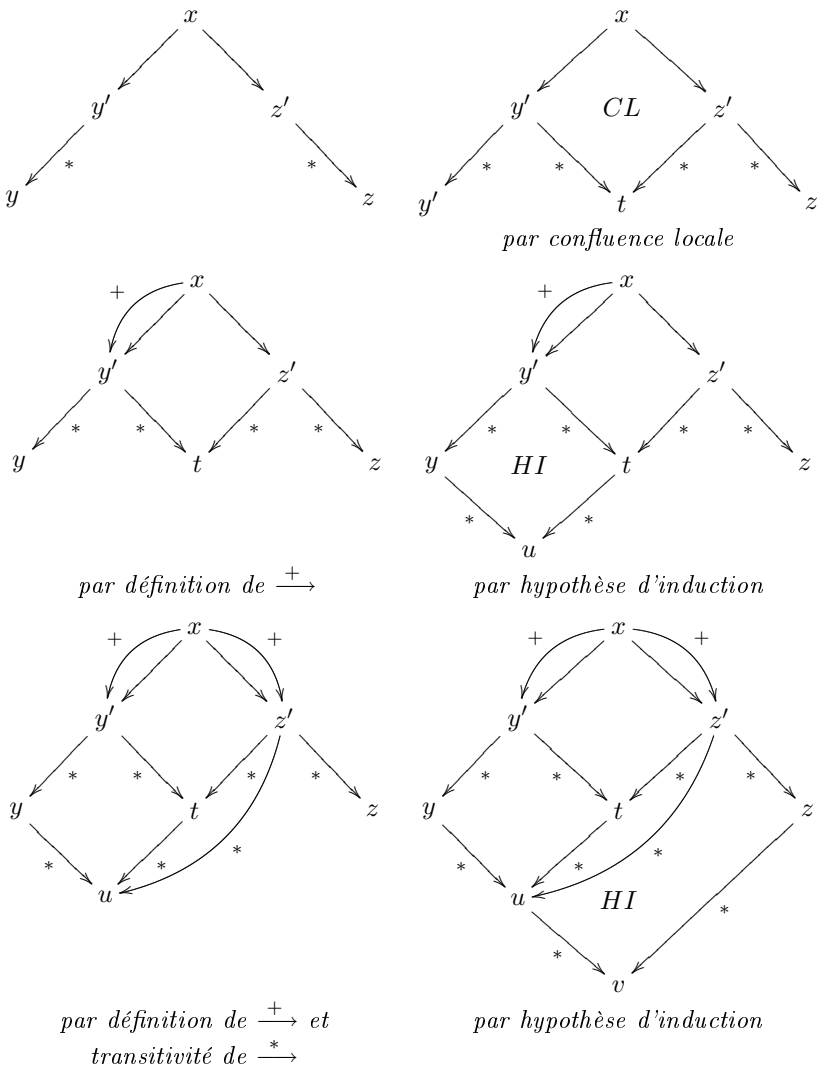
$$\longrightarrow termine, CL, HI,\ y \overset{*}{\longleftarrow} y' \longleftarrow x \longrightarrow z \ \vdash\ \underline{y} - \overset{*}{-} \succ t \prec \overset{*}{-} - \underline{z}$$

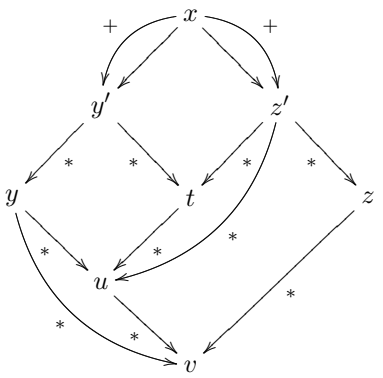
Par distinction de cas sur $x \xrightarrow{*} z$

Cas 2.1 *est similaire au cas 1*

Cas 2.2

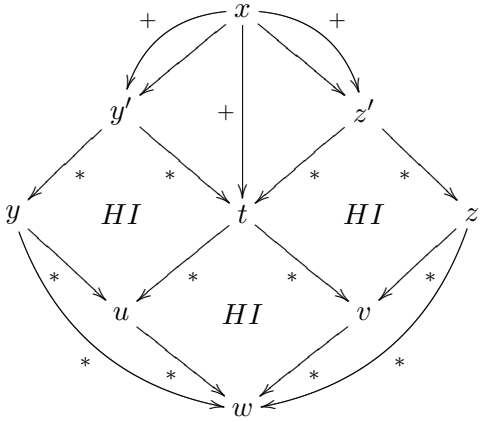
Pour la fin de la preuve nous représentons seulement l'hypothèse factuelle :





par transitivité de $\xrightarrow{*}$

Notons qu'il y a une preuve dont le diagramme final est symétrique. Mais cette preuve utilise trois fois l'hypothèse d'induction (notée HI sur le diagramme).



6.6 Implantation en Coq

Le système formel que nous avons présenté peut être implanté et utilisé effectivement pour faire des démonstrations au moyen d'un assistant de preuve. Dans un premier temps nous décrivons l'implantation que nous avons réalisée en Coq au moyen du langage de tactiques appelé $\mathcal{L}_{tac}$ [Del01, Del00, Coq04]. Nous verrons que le système proposé permet d'obtenir des preuves formelles concises car les preuves obtenues suivent précisément la construction du diagramme.

6.6.1 Règles d'inférence

Nous ne détaillons que l'implantation de la règle `apply`, les autres règles peuvent être traduites directement en Coq[5].

Pour construire la tactique correspondant à la règle `apply`, nous définissons d'abord une tactique qui est capable de trouver la conclusion d'une hypothèse[6] :

```
Ltac conclusion_aux t :=
  match t with
    | ?P1 -> ?P2 => conclusion_aux P2
    | _ => t
end.
```

Enfin nous exprimons la règle `apply` de la manière suivante, nous prouvons que la conclusion du diagramme universel est vraie grâce à la tactique `auto` de `apply_decompose`. Puis on décompose cette nouvelle connaissance grâce aux règles $\vee$,$\wedge$ et $\exists$ gauches en utilisant la tactique `decompose`.

```
Ltac decompose_and_clear id :=
 progress (decompose [or and ex] id);clear id.

Ltac apply_decompose H :=
  let t := type of H in
  let conc := conclusion_aux t in
  let id:= fresh in
  (assert (id:conc);[auto|try decompose_and_clear id]).

Ltac apply_diagram H :=
  let id:=fresh in
  (assert (id:=H);apply_decompose id;clear id);
  unfold_all.
```

[5] Attention, dans notre implantation les tactiques sont souvent capables de faire plus que les règles d'inférence qu'elles implantent. Nous supposons donc implicitement que les tactiques sont utilisées à bon escient.

[6] Nous supposons que les hypothèses sont curryfiées

Exemple

Nous reprenons l'exemple du lemme de Newman mais cette fois ci en Coq.

```
Theorem newman :
 local_confluence S R -> noetherian S R -> confluence S R.
Proof.
intros.
(* induction *)
assert (ind:=H0 (confluence_in S R));clear H0.
unfold confluence.
apply ind;clear ind.
unfold confluence_in.

start.
rename y into x.
rename y0 into y.

(* First degenerated case *)
apply_diagram (Rstar_cases x y).
substitute y.
apply_diagram (Rstar_cont_eq S R z).
conclusion.

(* Second degenerated case *)
apply_diagram (Rstar_cases x z).
substitute z.
apply_diagram (Rstar_cont_eq S R y).
conclusion.

(* General case *)
start.
apply_diagram (H x x0 x1).
apply_diagram (H0 x0);apply_diagram (H4 y x2).
apply_diagram (Rstar_transitivity x1 x2 x3).
apply_diagram (H0 x1);apply_diagram (H12 x3 z).
apply_diagram (Rstar_transitivity y x3 x4).
conclusion.
Qed.
```

6.6.2 Règles implicites

Comme le lecteur l'a peut-être déjà remarqué les preuves diagrammatiques dans notre système formel ressemblent fort à la preuve informelle

mais elles contiennent certaines étapes de preuve qui ne figurent pas dans la preuve informelle. Dans la preuve informelle certaines propriétés restent en effet implicites, par exemple la transitivité de l'égalité ou le fait qu'une relation est contenue dans sa clôture transitive.

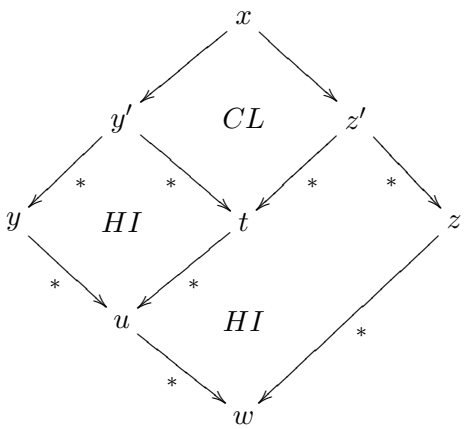
Nous sortons un instant du cadre formel que nous avons proposé et présentons comment rendre ces propriétés implicites aussi dans l'implantation Coq.

Les propriétés que nous choisissons comme implicites sont les suivantes :
- $\overset{*}{\longrightarrow}$ est transitive,
- $\overset{+}{\longrightarrow}$ est transitive,
- $\overset{*}{\longrightarrow}$ est réflexive,
- $\overset{*}{\longrightarrow}$ contient $\longrightarrow$,
- $\overset{+}{\longrightarrow}$ contient $\longrightarrow$.

Nous ajoutons ces propriétés à ce qui est appelé en Coq une base de «Hints». Puis nous définissons des variantes des tactiques décrites précédemment. Ces nouvelles tactiques permettent de travailler sans préciser les étapes que l'on a définies comme implicites.

```
Ltac Rconclusion :=
   eauto with Rules.
Ltac Rapply_diagram H :=
   apply_diagram H;[idtac|eauto with Rules].
```

L'utilisation de ces tactiques permet par exemple d'automatiser trois étapes dans la preuve que nous avons présentée ci-dessus. On obtient alors une preuve qui correspond au diagramme habituel pour la preuve du lemme de Newman, à savoir :



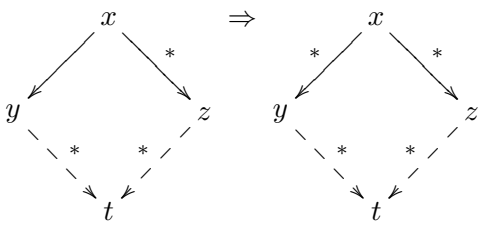
L'ensemble du développement est disponible en ligne à l'adresse suivante : `http://www.lix.polytechnique.fr/Labo/Julien.Narboux/Rewriting/rewriting.html`.

6.7 Quelques preuves diagrammatiques.

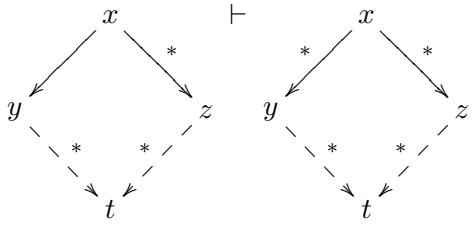
Dans cette partie nous donnons à titre d'exemple quelques preuves diagrammatiques de propriétés usuelles.

6.7.1 Propriétés de confluence

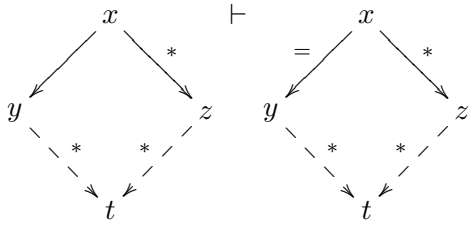
Lemme 5. *La semi-confluence implique la confluence.*



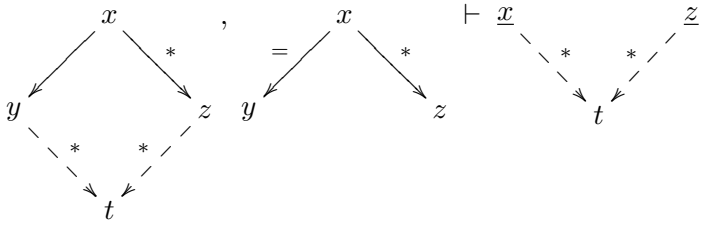
Preuve:



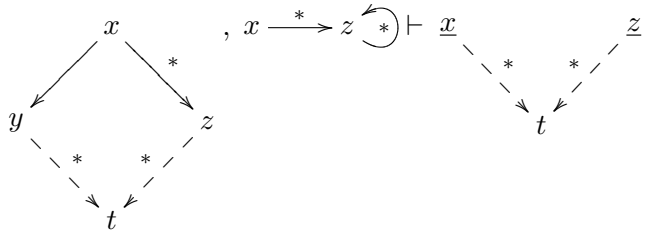
Par la règle ind_*
Cas 1 :



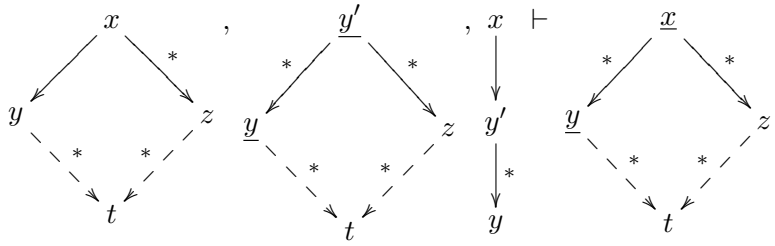
par la règle ***intros***



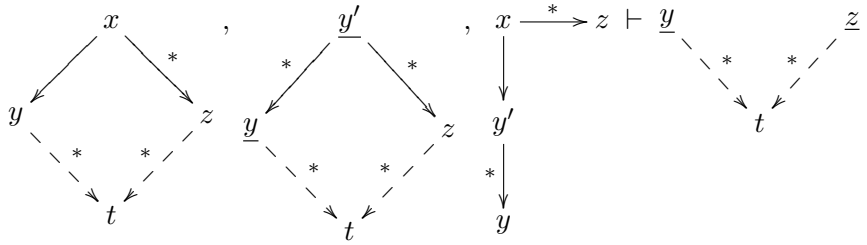
par la règle **apply** *appliquée à la définition de* $\xrightarrow{*}$



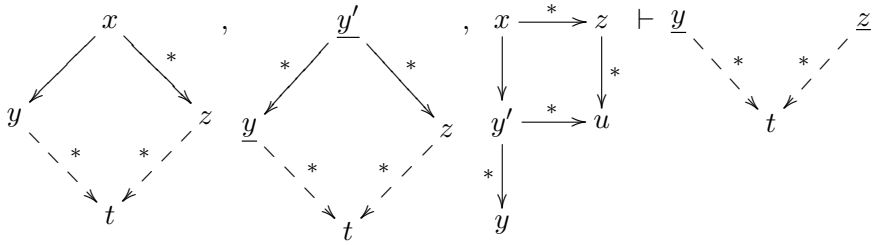
Cas 2 :



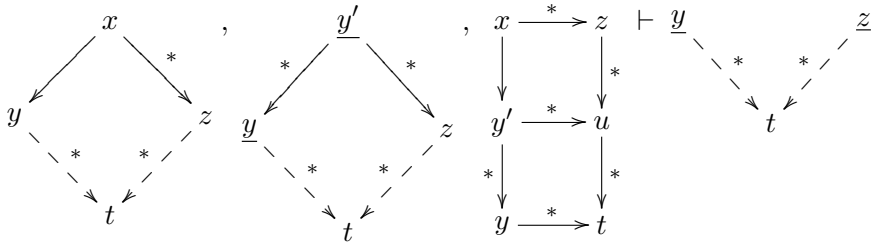
par la règle **intros**



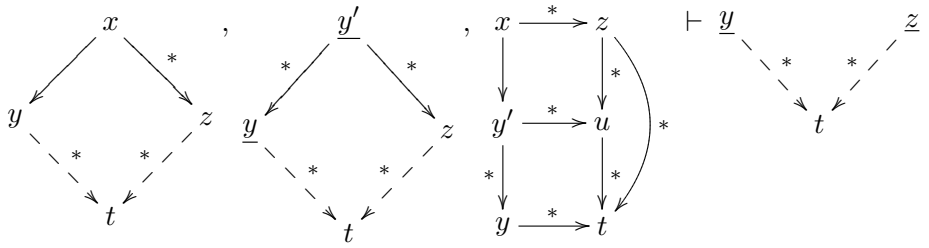
par la règle **apply**



par la règle **apply**

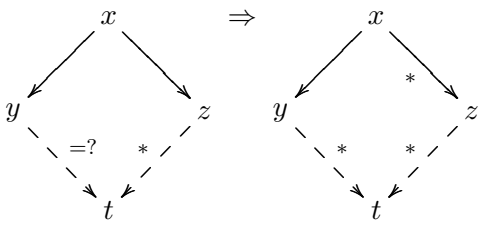


par la règle `apply` *appliquée aves la transitivité de* $\xrightarrow{*}$

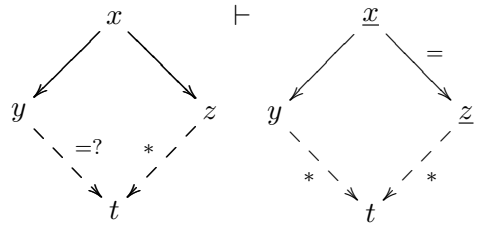


□

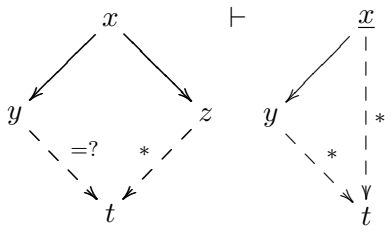
Lemme 6. *La confluence forte implique la semi confluence.*



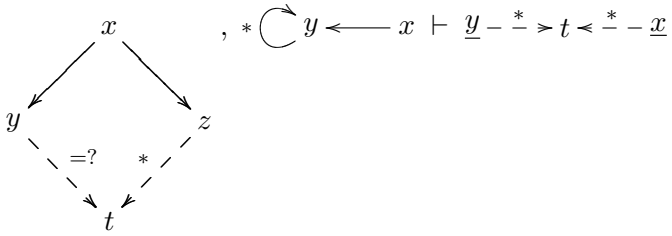
Preuve:
par la règle ind_*
Cas 1 :



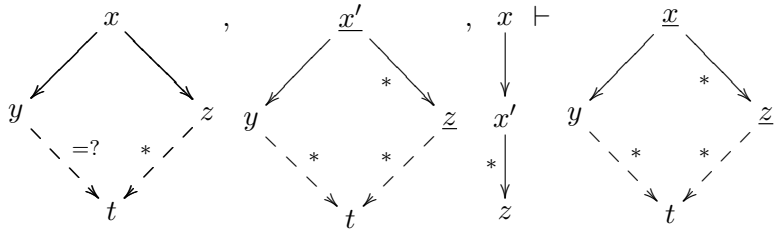
par la règle `substitute`



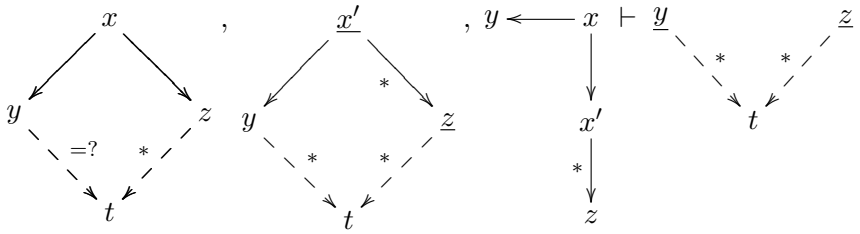
par les règles `intros` *et* `apply`



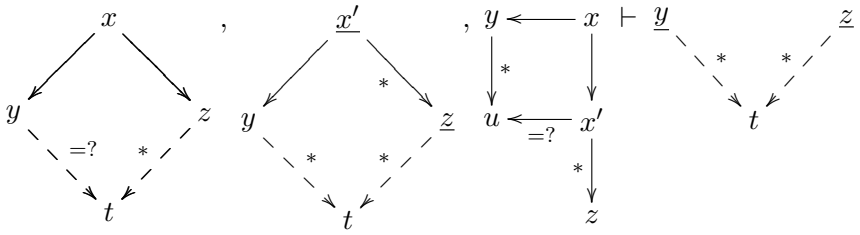
Cas 2 :



par la règle `intros`

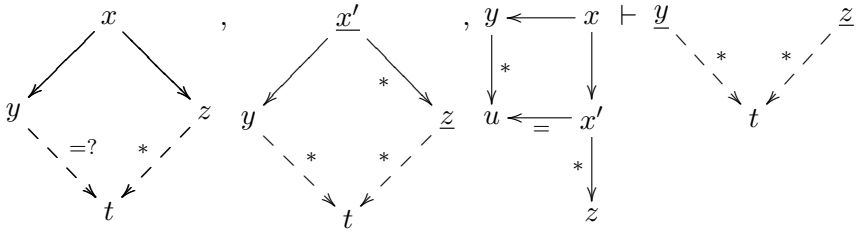


par la règle `apply`

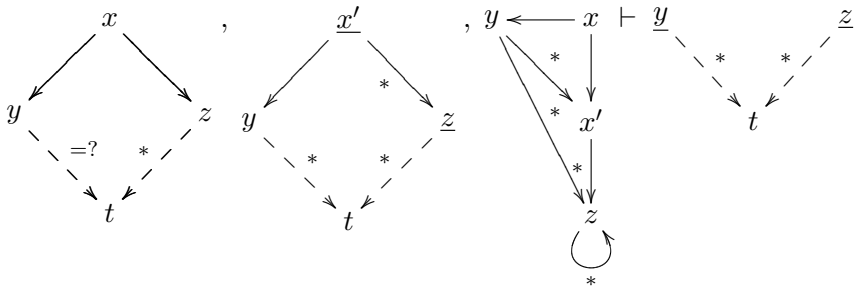


par la règle `apply` *appliquée à la définition de* $\xrightarrow{=?}$

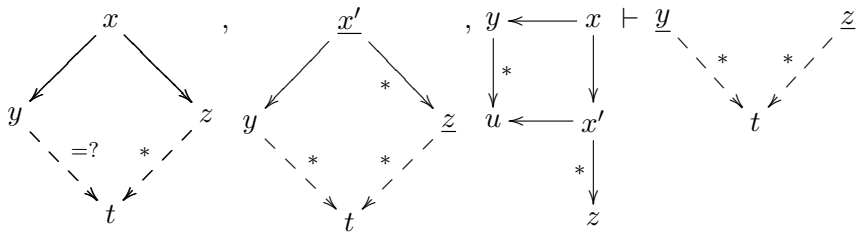
Cas 2.1 :



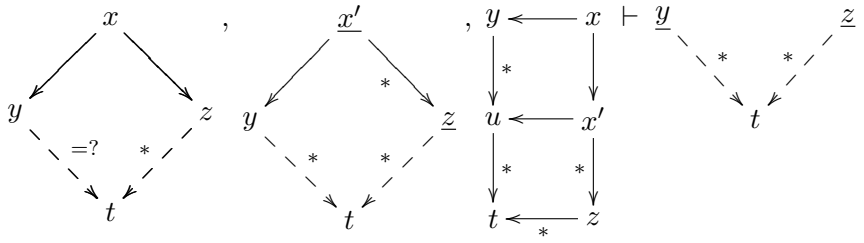
par la règle `substitute` *et la règle* `apply` *deux fois*



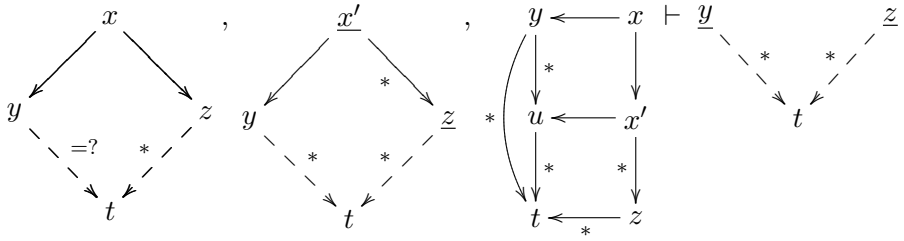
Cas 2.2 :



par la règle `apply`

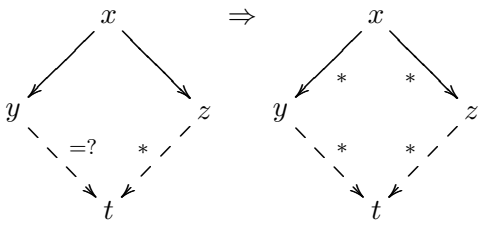


par la règle `apply` *appliquée avec la transitivité de* $\xrightarrow{*}$



□

Théorème 14. *La confluence forte implique la confluence.*



Preuve: Par application des lemmes précédents. □

6.8 Conclusion et perspectives

Nous avons proposé un système formel *diagrammatique* correct et complet pour la logique cohérente restreinte aux prédicats d'arité deux mais étendu avec des définitions inductives. Celui-ci permet de justifier formellement le genre de raisonnement diagrammatique utilisé dans le cadre de la réécriture abstraite.
Nous envisageons plusieurs extensions. Certaines théories géométriques peuvent être axiomatisées par des formules de la logique cohérente (la géométrie projective par exemple). Il serait interressant d'étudier les représentations diagrammatiques possibles des preuves dans ces théories. Nous envisageons aussi de nous intéresser à la théorie des catégories, dont la littérature comporte de nombreux diagrammes. Ces multiples extensions possibles suggèrent que la géométrie cohérente est bien adaptée au raisonnement diagrammatique. Il serait donc intéressant de prendre le point de vue inverse et de s'intéresser à la plus grande classe de formules pour laquelle il existe un système complet permettant le raisonnement diagrammatique. Nous discutons de ce problème plus en détails dans le chapitre « Perspectives ».

Perspectives

Nous évoquons dans cette partie trois directions dans lesquelles notre travail peut être étendu. Nous discutons d'abord de deux applications envisagées puis de la généralisation possible de nos résultats à propos du raisonnement diagrammatique.

Preuve de programmes en algorithmique géométrique

L'une des applications possibles des développements réalisés dans cette thèse réside dans la preuve de programmes en algorithmique géométrique. Comme la plupart des algorithmes géométriques résolvent des problèmes énoncés dans une géométrie *ordonnée* la méthode des aires ne peut pas être utilisée telle qu'elle. Afin de pouvoir résoudre des problèmes dans une géométrie ordonnée, nous envisageons de suivre l'idée de Judit Robu qui consiste à utiliser le processus d'élimination des points de la méthode des aires pour simplifier le but avant de le résoudre avec la méthode de décomposition algébrique cylindrique réalisée par Assia Mahboubi. L'implantation de cette méthode dans l'assistant de preuve Coq nécessite un travail conséquent afin de mettre en place les mécanismes qui permettront de passer facilement d'une représentation algébrique à une représentation géométrique et vice versa. En effet, quand c'est possible, nous souhaitons garder une représentation géométrique puisqu'elle permet de réaliser des développements modulaires. Par exemple les résultats que nous avons prouvés qui n'utilisent pas l'axiome d'Euclide pourront être réutilisés dans un futur développement à propos des géométries non euclidiennes.

Coq et l'enseignement

Notre expérience personnelle nous a montré que l'utilisation d'un assistant de preuve permet de mieux comprendre ce qu'est une preuve. Au lycée, la géométrie est le domaine qui permet aux éléves de découvrir la notion de *démonstration*. Notre but est donc de fournir un environnement qui permette l'utilisation d'un assistant de preuve dans une classe. Nous pensons que *GeoProof* représente une avancée dans cette direction. Mais il reste du chemin à parcourir avant d'obtenir un système qui répond aux contraintes liées à une utilisation dans un cadre pédagogique.
Voici quelques unes des améliorations nécessaires :

- réduire au maximum le temps d'apprentissage du logiciel en en améliorant l'ergonomie.
- automatiser la preuve de certains sous buts, en particulier les cas dégénérés.
- pouvoir faire varier la puissance du système et les théorèmes disponibles en fonction du programme du niveau considéré.

Raisonnement diagrammatique

La formalisation que nous avons réalisée du raisonnement diagrammatique en réécriture abstraite nous a permis de dégager deux composantes essentielles d'un raisonnement diagrammatique.
La première réside dans le fait que l'on doit pouvoir visualiser facilement les formules manipulées par *une syntaxe qui mime la sémantique*. Par exemple la notation pour la clôture symétrique est symétrique.
La seconde composante essentielle réside dans le fait que pour la classe de formules manipulée, il faut que les preuves puissent être réalisées selon un schéma bien particulier[7] : on part des hypothèses et on complète le diagramme au fur et à mesure de la preuve jusqu'à obtenir une instance du but désiré. Une propriété importante de ce schéma réside dans le fait que le but ne change pas pendant la preuve et peut donc rester *implicite* dans la représentation graphique. Nous pensons que ce schéma de raisonnement est bien adapté au raisonnement diagrammatique, et qu'il serait intéressant de rechercher quelle est la plus grande classe de formules pour laquelle il existe un système de preuve complet suivant ce schéma.

Nous avons réalisé une formalisation du raisonnement diagrammatique dans le cadre de la réécriture abstraite. Il est assez naturel de vouloir étendre ce travail aux autres preuves diagrammatiques qui apparaissent dans la littérature, par exemple celles à propos de la théorie des catégories. Dans ce cadre, la sémantique d'un diagramme consisterait en la conjonction de chacune des propriétés de commutation des cellules qui forment le diagramme.

[7] Nous nous plaçons ici dans le cas d'une preuve sans induction.

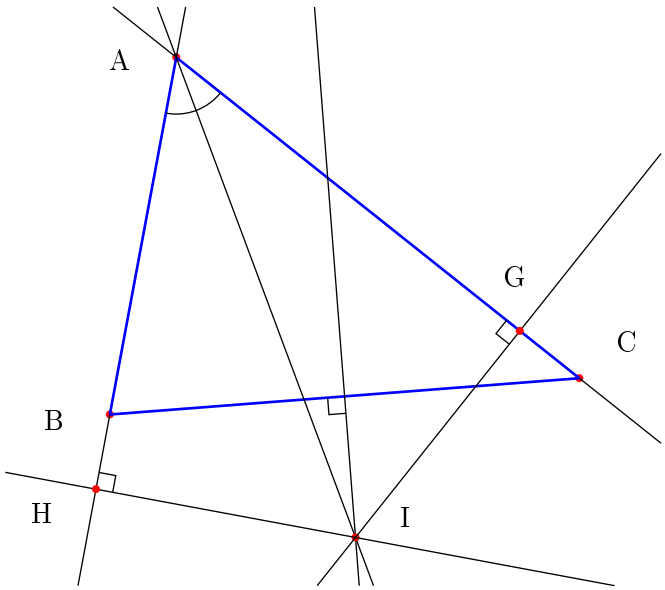
Il serait aussi intéressant d'étudier la conception d'un système de raisonnement diagrammatique dans le cadre de la géométrie. Dans un premier temps on pourra s'intéresser à la géométrie projective, car comme ses axiomes appartiennent à la géométrie cohérente, les règles d'inférence pourront être réutilisées. En revanche, il faudra s'intéresser au problème de la représentation graphique des énoncés manipulés. La géométrie euclidienne quant à elle représente un plus grand challenge, car il faut s'attaquer au problème de la représentation de la négation puisque celle-ci est présente dans les conditions de non-dégénérescence.

Annexe A

Tous les triangles sont-ils isocèles ?

Cette page explique la faille qui figure dans la « preuve » donnée en introduction (voir page 3) que tous les triangles sont isocèles. Cet exemple très connu de raisonnement non valide est dû à W. W. Rouse Ball.

L'explication est en fait très simple, les deux figures fournies sont fausses et induisent le lecteur en erreur ! La droite (AI) n'est pas exactement la bissectrice de l'angle $\angle BAC$, voici une figure correcte. L'argument d'addition des longueurs des segments n'est pas valide car il faudrait que G et H soient tous les deux à l'extérieur ou tous les deux à l'intérieur du triangle ABC.



Annexe B

La géométrie de Tarski

Cette annexe contient le détail des énoncés prouvés concernant la formalisation des huit premiers chapitres de [SST83]. Pour de plus amples informations voir le chapitre 2 page 29. Ces pages sont générées automatiquement à partir du développement Coq, les commentaires sont donc en anglais.

B.1 Axiomes

We use the axioms used in the book of Schwabäuser, Szmielew and Tarski.

We assume that we have a set of points. This axiomatic is based only on points, lines are implicit.

Parameter $Point : Set.$

Tarski's axiomatic is based on two predicates over points, one ternary and one quaternary.

Cong A B C D informally means that the distance AB is equal to the distance CD.

Parameter $Cong : Point \to Point \to Point \to Point \to Prop.$

Bet A B C, informally means that : A, B and C are on the same line B is between A and C. Note that they are not necessarily distinct. This is the difference between the Between predicate of Tarski and the BetweenH prdicate of Hilbert

Parameter $Bet : Point \to Point \to Point \to Prop.$

Definition $Col\ (A\ B\ C : Point) : Prop :=$
$Bet\ A\ B\ C \vee Bet\ B\ C\ A \vee Bet\ C\ A\ B.$

Identity axiom for betweenness.

Axiom *between_identity* : $\forall\ A\ B,\ Bet\ A\ B\ A \rightarrow A{=}B.$

Reflexivity axiom for equidistance.

Axiom *cong_pseudo_reflexivity* : $\forall\ A\ B : Point,\ Cong\ A\ B\ B\ A.$

Identity axiom for equidistance.

Axiom *cong_identity* : $\forall\ A\ B\ C : Point,\ Cong\ A\ B\ C\ C \rightarrow A = B.$

Transitivity axiom for equidistance.

Axiom *cong_inner_transitivity* : $\forall\ A\ B\ C\ D\ E\ F : Point,$
$Cong\ A\ B\ C\ D \rightarrow Cong\ A\ B\ E\ F \rightarrow Cong\ C\ D\ E\ F.$

Pasch Axiom (Inner form).

Axiom *inner_pasch* : $\forall\ A\ B\ C\ P\ Q : Point,$
$Bet\ A\ P\ C \rightarrow Bet\ B\ Q\ C \rightarrow$
$\exists\ x,\ Bet\ P\ x\ B \wedge Bet\ Q\ x\ A.$

Euclid Axiom.

Axiom *euclid* : $\forall\ A\ B\ C\ D\ T : Point,$
$Bet\ A\ D\ T \rightarrow Bet\ B\ D\ C \rightarrow A{\neq}D \rightarrow$
$\exists\ x,\ \exists\ y,$
$Bet\ A\ B\ x \wedge Bet\ A\ C\ y \wedge Bet\ x\ T\ y.$

Five segments axiom.

Axiom *five_segments* : $\forall\ A\ A'\ B\ B'\ C\ C'\ D\ D' : Point,$
$Cong\ A\ B\ A'\ B' \rightarrow$
$Cong\ B\ C\ B'\ C' \rightarrow$
$Cong\ A\ D\ A'\ D' \rightarrow$
$Cong\ B\ D\ B'\ D' \rightarrow$
$Bet\ A\ B\ C \rightarrow Bet\ A'\ B'\ C' \rightarrow A \neq B \rightarrow Cong\ C\ D\ C'\ D'.$

Axiom of segment construction.

Axiom *segment_construction* : $\forall\ A\ B\ C\ D : Point,$
$\exists\ E : Point,\ Bet\ A\ B\ E \wedge Cong\ B\ E\ C\ D.$

Lower dimension axiom (2).

Axiom *lower_dim* :
$\exists\ A,\ \exists\ B,\ \exists\ C,\ \neg\ Col\ A\ B\ C.$

Upper dimension axiom (2).

Axiom *upper_dim* : $\forall\ A\ B\ C\ P\ Q : Point,$
$P \neq Q \rightarrow Cong\ A\ P\ A\ Q \rightarrow Cong\ B\ P\ B\ Q \rightarrow Cong\ C\ P\ C\ Q \rightarrow$
$Col\ A\ B\ C.$

Continuity axioms.

Axiom *continuity* :
$\forall\ X\ Y : Point \rightarrow Prop,$
$(\exists\ A : Point, (\forall\ x\ y : Point, X\ x \rightarrow Y\ y \rightarrow Bet\ A\ x\ y)) \rightarrow$
$\exists\ B : Point, (\forall\ x\ y : Point, X\ x \rightarrow Y\ y \rightarrow Bet\ x\ B\ y).$

We use classical logic, to emphasize the fact that it is mainly for the decidability of point equality we use the following lemma

Lemma *eq_dec_points* : $\forall\ A\ B : Point, A{=}B \vee \neg\ A{=}B.$

B.2 Propriétés de `Cong`

Some basic properties about Cong.

Lemma *cong_reflexivity* : $\forall\ A\ B : Point, Cong\ A\ B\ A\ B.$

Lemma *cong_symmetry* : $\forall\ A\ B\ C\ D : Point, Cong\ A\ B\ C\ D \rightarrow Cong\ C\ D\ A\ B.$

Lemma *cong_transitivity* :
$\forall\ A\ B\ C\ D\ E\ F : Point, Cong\ A\ B\ C\ D \rightarrow Cong\ C\ D\ E\ F \rightarrow Cong\ A\ B\ E\ F.$

Lemma *cong_left_commutativity* : $\forall\ A\ B\ C\ D,$
$Cong\ A\ B\ C\ D \rightarrow Cong\ B\ A\ C\ D.$

Lemma *cong_right_commutativity* : $\forall\ A\ B\ C\ D,$
$Cong\ A\ B\ C\ D \rightarrow Cong\ A\ B\ D\ C.$

Lemma *cong_trivial_identity* : $\forall\ A\ B : Point, Cong\ A\ A\ B\ B.$

Lemma *cong_reverse_identity* : $\forall\ A\ C\ D, Cong\ A\ A\ C\ D \rightarrow C{=}D.$

Lemma *cong_commutativity* : $\forall\ A\ B\ C\ D, Cong\ A\ B\ C\ D \rightarrow Cong\ B\ A\ D\ C.$

(Outer) Five segments configuration.

Definition *OFSC* := *fun A B C D A' B' C' D'* $\Rightarrow$
$Bet\ A\ B\ C \wedge Bet\ A'\ B'\ C' \wedge$
$Cong\ A\ B\ A'\ B' \wedge Cong\ B\ C\ B'\ C' \wedge$
$Cong\ A\ D\ A'\ D' \wedge Cong\ B\ D\ B'\ D'.$

Lemma *five_segments_with_def* : $\forall\ A\ B\ C\ D\ A'\ B'\ C'\ D',$
$OFSC\ A\ B\ C\ D\ A'\ B'\ C'\ D' \rightarrow A{\neq}B \rightarrow Cong\ C\ D\ C'\ D'.$

Lemma *l2_11* :
$\forall\ A\ B\ C\ A'\ B'\ C'$,
$Bet\ A\ B\ C \rightarrow Bet\ A'\ B'\ C' \rightarrow$
$Cong\ A\ B\ A'\ B' \rightarrow Cong\ B\ C\ B'\ C' \rightarrow Cong\ A\ C\ A'\ C'$.

Unicity of segment construction.

Lemma *construction_unicity* : $\forall\ Q\ A\ B\ C\ x\ y$,
$\tilde{}(Q{=}A) \rightarrow$
$Bet\ Q\ A\ x \rightarrow Cong\ A\ x\ B\ C \rightarrow$
$Bet\ Q\ A\ y \rightarrow Cong\ A\ y\ B\ C \rightarrow$
$x{=}y$.

B.3 Propriétés de `Bet`

Axiom 12 of Givant.

Lemma *beetween_trivial* : $\forall\ A\ B : Point, Bet\ A\ B\ B$.

Axiom 14 of Givant.

Lemma *between_symmetry* : $\forall\ A\ B\ C : Point$,
$Bet\ A\ B\ C \rightarrow Bet\ C\ B\ A$.

Lemma *beetween_trivial2* : $\forall\ A\ B : Point, Bet\ A\ A\ B$.

Lemma *between_egality* : $\forall\ A\ B\ C : Point, Bet\ A\ B\ C \rightarrow Bet\ B\ A\ C \rightarrow A = B$.

Inner transitivity 'axiom' for betweenness.

Lemma *between_inner_transitivity* : $\forall\ A\ B\ C\ D$,
$Bet\ A\ B\ D \rightarrow Bet\ B\ C\ D \rightarrow Bet\ A\ B\ C$.

Lemma *between_exchange3* : $\forall\ A\ B\ C\ D$,
$Bet\ A\ B\ C \rightarrow Bet\ A\ C\ D \rightarrow Bet\ B\ C\ D$.

Lemma *outer_transitivity_between2* : $\forall\ A\ B\ C\ D$,
$Bet\ A\ B\ C \rightarrow Bet\ B\ C\ D \rightarrow B{\neq}C \rightarrow Bet\ A\ C\ D$.

Lemma *between_exchange2* : $\forall\ A\ B\ C\ D$,
$Bet\ A\ B\ D \rightarrow Bet\ B\ C\ D \rightarrow Bet\ A\ C\ D$.

Axiom 16 of Givant.

Lemma *outer_transitivity_between* : $\forall\ A\ B\ C\ D$,
$Bet\ A\ B\ C \rightarrow Bet\ B\ C\ D \rightarrow B{\neq}C \rightarrow Bet\ A\ B\ D$.

Lemma *between_exchange4* : $\forall\ A\ B\ C\ D$,
$Bet\ A\ B\ C \rightarrow Bet\ A\ C\ D \rightarrow Bet\ A\ B\ D$.

We formalize Bet_n only for n=4.

Definition *Bet_4* := *fun A1 A2 A3 A4* ⇒
Bet A1 A2 A3 ∧ *Bet A2 A3 A4* ∧ *Bet A1 A3 A4* ∧ *Bet A1 A2 A4*.

Lemma *l_3_9_4* : ∀ *A1 A2 A3 A4*,
Bet_4 A1 A2 A3 A4 → *Bet_4 A4 A3 A2 A1*.

There are two distinct points !
This is the first use of the lower dimension axiom

Lemma *two_distinct_points* : ∃ *x*, ∃ *y* : *Point*, *x* ≠ *y*.

Lemma *point_construction_different* : ∀ *A B*,
∃ *C*, *Bet A B C* ∧ *B* ≠ *C*.

Given a point, we can build another one which is different from the first one.

Lemma *another_point* : ∀ *A* :*Point*, ∃ *B*, *A*≠*B*.

From the last two lemmas we can conclude that models of the first seven axioms are infinite

Lemma *l3_17* : ∀ *A B C A' B' P*,
Bet A B C → *Bet A' B' C* → *Bet A P A'* →
∃ *Q*, *Bet P Q C* ∧ *Bet B Q B'*.

Inner five segment configuration.

Definition *IFSC* := *fun A B C D A' B' C' D'* ⇒
Bet A B C ∧ *Bet A' B' C'* ∧
Cong A C A' C' ∧ *Cong B C B' C'* ∧
Cong A D A' D' ∧ *Cong C D C' D'*.

This tactic asserts Col X Y Z, for each Bet X Y Z in the context.

This tactic looks for an equality, perform the substitution and then attempts to prove other equalities and apply reccursively. It cleans the hypothesis which become trivial or duplicated

B.4 Propriétés de `Cong` et `Bet`

Lemma *l4_2* : ∀ *A B C D A' B' C' D'*,
IFSC A B C D A' B' C' D' → *Cong B D B' D'*.

Lemma *l4_3* : ∀ *A B C A' B' C'*,

Bet A B C → Bet A' B' C' →
Cong A C A' C' → Cong B C B' C' → Cong A B A' B'.

A generalization of the Cong predicate for three points.

Definition *Cong_3 := fun A1 A2 A3 B1 B2 B3 ⇒*
Cong A1 A2 B1 B2 ∧ Cong A1 A3 B1 B3 ∧ Cong A2 A3 B2 B3.

Lemma *l4_5* : ∀ *A B C A' C'*,
Bet A B C → Cong A C A' C' →
∃ *B'*, *Bet A' B' C' ∧ Cong_3 A B C A' B' C'.*

Lemma *l4_6* : ∀ *A B C A' B' C'*,
Bet A B C → Cong_3 A B C A' B' C' → Bet A' B' C'.

Permutation lemmas for Col

Lemma *col_permutation_1* : ∀ *A B C*,
Col A B C → Col B C A.

Lemma *col_permutation_2* : ∀ *A B C*,
Col A B C → Col C A B.

Lemma *col_permutation_3* : ∀ *A B C*,
Col A B C → Col C B A.

Lemma *col_permutation_4* : ∀ *A B C*,
Col A B C → Col B A C.

Lemma *col_permutation_5* : ∀ *A B C*,
Col A B C → Col A C B.

Trivial lemmas for Col.

Lemma *col_trivial_1* : ∀ *A B*, *Col A A B.*

Lemma *col_trivial_2* : ∀ *A B*, *Col A B B.*

Lemma *col_trivial_3* : ∀ *A B*, *Col A B A.*

Lemma *l4_13* : ∀ *A B C A' B' C'*,
Col A B C → Cong_3 A B C A' B' C' → Col A' B' C'.

Lemma *l4_14* : ∀ *A B C A' B'*,
Col A B C → Cong A B A' B' →
∃ *C'*, *Cong_3 A B C A' B' C'.*

Five segments configuration.

Definition $FSC := fun\ A\ B\ C\ D\ A'\ B'\ C'\ D' \Rightarrow$
$Col\ A\ B\ C \wedge Cong\_3\ A\ B\ C\ A'\ B'\ C' \wedge Cong\ A\ D\ A'\ D' \wedge Cong\ B\ D\ B'\ D'.$

Lemma $l4\_16 : \forall\ A\ B\ C\ D\ A'\ B'\ C'\ D',$
$FSC\ A\ B\ C\ D\ A'\ B'\ C'\ D' \rightarrow A \neq B \rightarrow Cong\ C\ D\ C'\ D'.$

Lemma $l4\_17 : \forall\ A\ B\ C\ P\ Q,$
$A \neq B \rightarrow Col\ A\ B\ C \rightarrow$
$Cong\ A\ P\ A\ Q \rightarrow Cong\ B\ P\ B\ Q \rightarrow Cong\ C\ P\ C\ Q.$

Lemma $l4\_18 : \forall\ A\ B\ C\ C',$
$A \neq B \rightarrow Col\ A\ B\ C \rightarrow$
$Cong\ A\ C\ A\ C' \rightarrow Cong\ B\ C\ B\ C' \rightarrow C = C'.$

Lemma $l4\_19 : \forall\ A\ B\ C\ C',$
$Bet\ A\ C\ B \rightarrow Cong\ A\ C\ A\ C' \rightarrow Cong\ B\ C\ B\ C' \rightarrow C = C'.$

B.5 Transitivité de `Bet`

Outer connectivity for betweenness Axiom III of Tarski Axiom 18 of Givant

This proof is from Gupta 1965, it used to be an axiom in previous versions of Tarski's axiomatic

Lemma $l5\_1 : \forall\ A\ B\ C\ D,$
$A \neq B \rightarrow Bet\ A\ B\ C \rightarrow Bet\ A\ B\ D \rightarrow Bet\ A\ C\ D \vee Bet\ A\ D\ C.$

Lemma $l5\_2 : \forall\ A\ B\ C\ D,$
$A \neq B \rightarrow Bet\ A\ B\ C \rightarrow Bet\ A\ B\ D \rightarrow Bet\ B\ C\ D \vee Bet\ B\ D\ C.$

Inner connectivity axiom for betweenness
Axiom 17 of Givant

Lemma $l5\_3 : \forall\ A\ B\ C\ D,$
$Bet\ A\ B\ D \rightarrow Bet\ A\ C\ D \rightarrow Bet\ A\ B\ C \vee Bet\ A\ C\ B.$

Predicates to compare segment lengths.

Definition $le := fun\ A\ B\ C\ D \Rightarrow$
$\exists\ y, Bet\ C\ y\ D \wedge Cong\ A\ B\ C\ y.$

Definition $ge := fun\ A\ B\ C\ D \Rightarrow le\ C\ D\ A\ B.$

Lemma $l5\_5\_1 : \forall\ A\ B\ C\ D,$
$le\ A\ B\ C\ D \rightarrow \exists\ x, Bet\ A\ B\ x \wedge Cong\ A\ x\ C\ D.$

Lemma $l5\_5\_2 : \forall\ A\ B\ C\ D,$

$(\exists\ x,\ Bet\ A\ B\ x \wedge Cong\ A\ x\ C\ D) \rightarrow le\ A\ B\ C\ D.$

Lemma *l5_6* : $\forall\ A\ B\ C\ D\ A'\ B'\ C'\ D'$,
$le\ A\ B\ C\ D \wedge Cong\ A\ B\ A'\ B' \wedge Cong\ C\ D\ C'\ D' \rightarrow le\ A'\ B'\ C'\ D'.$

Lemma *le_reflexivity* : $\forall\ A\ B, le\ A\ B\ A\ B.$

Lemma *le_transitivity* : $\forall\ A\ B\ C\ D\ E\ F$,
$le\ A\ B\ C\ D \rightarrow le\ C\ D\ E\ F \rightarrow le\ A\ B\ E\ F\ .$

Two crucials lemmas not found in the book

Lemma *between_cong* : $\forall\ A\ B\ C$,
$Bet\ A\ C\ B \rightarrow Cong\ A\ C\ A\ B \rightarrow C{=}B.$

Lemma *between_cong_2* : $\forall\ A\ B\ D\ E$,
$Bet\ A\ D\ B \rightarrow Bet\ A\ E\ B \rightarrow Cong\ A\ D\ A\ E \rightarrow D = E.$

Lemma *le_anti_symmetry* : $\forall\ A\ B\ C\ D$,
$le\ A\ B\ C\ D \rightarrow le\ C\ D\ A\ B \rightarrow Cong\ A\ B\ C\ D.$

Lemma *segment_construction_2* : $\forall\ A\ Q\ B\ C$,
$A{\neq}Q \rightarrow \exists\ X, (Bet\ Q\ A\ X \vee Bet\ Q\ X\ A) \wedge Cong\ Q\ X\ B\ C.$

Lemma *le_trivial* : $\forall\ A\ C\ D, le\ A\ A\ C\ D\ .$

Lemma *le_cases* : $\forall\ A\ B\ C\ D$,
$le\ A\ B\ C\ D \vee le\ C\ D\ A\ B.$

Lemma *le_zero* : $\forall\ A\ B\ C, le\ A\ B\ C\ C \rightarrow A{=}B.$

Definition *lt* := $fun\ A\ B\ C\ D \Rightarrow le\ A\ B\ C\ D \wedge \neg\ Cong\ A\ B\ C\ D.$
Definition *gt* := $fun\ A\ B\ C\ D \Rightarrow lt\ C\ D\ A\ B.$

B.6 Prédicat de non appartenance à un segment

The predicate out P A B informally means that P is on the line A B outside the segment A B.

Definition *out* := $fun\ P\ A\ B \Rightarrow A{\neq}P \wedge B{\neq}P \wedge (Bet\ P\ A\ B \vee Bet\ P\ B\ A).$

Lemma *l6_2* : $\forall\ A\ B\ C\ P$,
$A{\neq}P \rightarrow B{\neq}P \rightarrow C{\neq}P \rightarrow Bet\ A\ P\ C \rightarrow (Bet\ B\ P\ C \leftrightarrow out\ P\ A\ B).$

Lemma *l6_3_1* : $\forall\ A\ B\ P$,
$out\ P\ A\ B \rightarrow (A{\neq}P \wedge B{\neq}P \wedge \exists\ C,\ C{\neq}P \wedge Bet\ A\ P\ C \wedge Bet\ B\ P\ C).$

Lemma *l6_3_2* : $\forall\ A\ B\ P$,
$(A{\neq}P \wedge B{\neq}P \wedge \exists\ C,\ C{\neq}P \wedge Bet\ A\ P\ C \wedge Bet\ B\ P\ C) \rightarrow out\ P\ A\ B.$

Lemma *l6_4_1* : $\forall\ A\ B\ P$,
$out\ P\ A\ B \rightarrow Col\ A\ P\ B \wedge \neg\ Bet\ A\ P\ B$.

Lemma *l6_4_2* : $\forall\ A\ B\ P$,
$Col\ A\ P\ B \wedge \neg\ Bet\ A\ P\ B \rightarrow out\ P\ A\ B$.

out reflexivity.

Lemma *l6_5* : $\forall\ P\ A$, $A{\neq}P \rightarrow out\ P\ A\ A$.

out symmetry.

Lemma *l6_6* : $\forall\ P\ A\ B$, $out\ P\ A\ B \rightarrow out\ P\ B\ A$.

out transitivity.

Lemma *l6_7* : $\forall\ P\ A\ B\ C$, $out\ P\ A\ B \rightarrow out\ P\ B\ C \rightarrow out\ P\ A\ C$.

Lemma *l6_11_unicity* : $\forall\ A\ B\ C\ R\ X\ Y$,
$R{\neq}A \rightarrow B{\neq}C \rightarrow$
$out\ A\ X\ R \rightarrow Cong\ A\ X\ B\ C \rightarrow$
$out\ A\ Y\ R \rightarrow Cong\ A\ Y\ B\ C \rightarrow$
$X{=}Y$.

Lemma *l6_11_existence* : $\forall\ A\ B\ C\ R$,
$R{\neq}A \rightarrow B{\neq}C \rightarrow$
$\exists\ X$, $out\ A\ X\ R \wedge Cong\ A\ X\ B\ C$.

Lemma *l6_13_1* : $\forall\ P\ A\ B$,
$out\ P\ A\ B \rightarrow le\ P\ A\ P\ B \rightarrow Bet\ P\ A\ B$.

Lemma *l6_13_2* : $\forall\ P\ A\ B$,
$out\ P\ A\ B \rightarrow Bet\ P\ A\ B \rightarrow le\ P\ A\ P\ B$.

Lemma *l6_16_1* : $\forall\ P\ Q\ S\ X$,
$P{\neq}Q \rightarrow S{\neq}P \rightarrow Col\ S\ P\ Q \rightarrow (Col\ X\ P\ Q \rightarrow Col\ X\ P\ S)$.

Lemma *l6_16_2* : $\forall\ P\ Q\ S\ X$,
$P{\neq}Q \rightarrow S{\neq}P \rightarrow Col\ S\ P\ Q \rightarrow (Col\ X\ P\ S \rightarrow Col\ X\ P\ Q)$.

Definition *Inter* := *fun* $X\ P\ Q\ R\ S \Rightarrow$
$P{\neq}Q \wedge R \neq S \wedge Col\ X\ P\ Q \wedge Col\ X\ R\ S \wedge$
$((\exists\ X,\ Col\ X\ P\ Q \wedge \neg\ Col\ X\ R\ S) \vee (\exists\ X, \neg\ Col\ X\ P\ Q \wedge Col\ X\ R\ S))$.

Lemma *col_transitivity_1* : $\forall\ P\ Q\ A\ B$,
$P{\neq}Q \rightarrow Col\ P\ Q\ A \rightarrow Col\ P\ Q\ B \rightarrow Col\ P\ A\ B$.

Lemma *col_transitivity_2* : $\forall\ P\ Q\ A\ B$,
$P{\neq}Q \rightarrow Col\ P\ Q\ A \rightarrow Col\ P\ Q\ B \rightarrow Col\ Q\ A\ B$.

Theorem *l6_21* : $\forall\ A\ B\ C\ D\ P\ Q$,

$\neg\ Col\ A\ B\ C \rightarrow C{\neq}D \rightarrow$
$Col\ A\ B\ P \rightarrow Col\ A\ B\ Q \rightarrow$
$Col\ C\ D\ P \rightarrow Col\ C\ D\ Q \rightarrow$
$P{=}Q.$

Lemma *l6_25* : $\forall\ A\ B,\ A{\neq}B \rightarrow \exists\ C,\ \neg\ Col\ A\ B\ C.$

Theorem *t2_8* :
$\forall\ A\ B\ C\ D\ E : Point,$
$Bet\ A\ B\ C \rightarrow$
$Bet\ D\ B\ E \rightarrow Cong\ A\ B\ D\ B \rightarrow Cong\ B\ C\ B\ E \rightarrow Cong\ A\ E\ C\ D.$

B.7 Propriétés du milieu

Definition *is_midpoint* := $fun\ M\ A\ B \Rightarrow Bet\ A\ M\ B \wedge Cong\ A\ M\ M\ B.$

Lemma *l7_2* : $\forall\ M\ A\ B,\ is\_midpoint\ M\ A\ B \rightarrow is\_midpoint\ M\ B\ A.$

Lemma *l7_3* : $\forall\ M\ A,\ is\_midpoint\ M\ A\ A \rightarrow M{=}A.$

Lemma *l7_3_2* : $\forall\ A,\ is\_midpoint\ A\ A\ A.$

Lemma *symmetric_point_construction* : $\forall\ A\ P,$
$\exists\ P',\ is\_midpoint\ P\ A\ P'.$

Lemma *symmetric_point_unicity* : $\forall\ A\ P\ P1\ P2,$
$is\_midpoint\ P\ A\ P1 \rightarrow is\_midpoint\ P\ A\ P2 \rightarrow P1{=}P2.$

Lemma *l7_9* : $\forall\ P\ Q\ A\ X,$
$is\_midpoint\ A\ P\ X \rightarrow is\_midpoint\ A\ Q\ X \rightarrow P{=}Q.$

Lemma *l7_13* : $\forall\ A\ P\ Q\ P'\ Q',$
$is\_midpoint\ A\ P'\ P \rightarrow is\_midpoint\ A\ Q'\ Q \rightarrow$
$Cong\ P\ Q\ P'\ Q'.$

Symmetry preserves Bet.

Lemma *l7_15* : $\forall\ P\ Q\ R\ P'\ Q'\ R'\ A,$
$is\_midpoint\ A\ P\ P' \rightarrow is\_midpoint\ A\ Q\ Q' \rightarrow is\_midpoint\ A\ R\ R' \rightarrow$
$Bet\ P\ Q\ R \rightarrow Bet\ P'\ Q'\ R'.$

Symmetry preserves Cong.

Lemma *l7_16* : $\forall\ P\ Q\ R\ S\ P'\ Q'\ R'\ S'\ A,$
$is\_midpoint\ A\ P\ P' \rightarrow is\_midpoint\ A\ Q\ Q' \rightarrow$
$is\_midpoint\ A\ R\ R' \rightarrow is\_midpoint\ A\ S\ S' \rightarrow$
$Cong\ P\ Q\ R\ S \rightarrow Cong\ P'\ Q'\ R'\ S'.$

Symmetry preserves midpoint

Lemma *symmetry_preserves_midpoint* :
∀ *A B C D E F Z*,
is_midpoint Z A D → *is_midpoint Z B E* → *is_midpoint Z C F* →
is_midpoint B A C → *is_midpoint E D F*.

Lemma *l7_17* : ∀ *P P' A B*,
is_midpoint A P P' → *is_midpoint B P P'* → *A=B*.

Lemma *l7_20* : ∀ *M A B*,
Col A M B → *Cong M A M B* → *A=B* ∨ *is_midpoint M A B*.

Lemma *l7_21* : ∀ *A B C D P*,
¬ *Col A B C* → *B≠D* → *Cong A B C D* → *Cong B C D A* →
Col A P C → *Col B P D* → *is_midpoint P A C* ∧ *is_midpoint P B D*.

Lemma *l7_22_aux* : ∀ *A1 A2 B1 B2 C M1 M2*,
Bet A1 C A2 → *Bet B1 C B2* →
Cong C A1 C B1 → *Cong C A2 C B2* →
is_midpoint M1 A1 B1 → *is_midpoint M2 A2 B2* → *le C A1 C A2* →
Bet M1 C M2.

Lemma *l7_22* : ∀ *A1 A2 B1 B2 C M1 M2*,
Bet A1 C A2 → *Bet B1 C B2* →
Cong C A1 C B1 → *Cong C A2 C B2* →
is_midpoint M1 A1 B1 → *is_midpoint M2 A2 B2* →
Bet M1 C M2.

Lemma *l7_25* : ∀ *A B C*, *Cong C A C B* → ∃ *X*, *is_midpoint X A B*.

B.8 Orthogonalité et existence du milieu

Definition *Per* := *fun A B C* ⇒
∃ *C'*, *is_midpoint B C C'* ∧ *Cong A C A C'*.

Lemma *l8_2* : ∀ *A B C*, *Per A B C* → *Per C B A*.

Lemma *l8_3* : ∀ *A B C A'*,
Per A B C → *A≠B* → *Col B A A'* → *Per A' B C*.

Lemma *l8_4* : ∀ *A B C C'*,
Per A B C → *is_midpoint B C C'* → *Per A B C'*.

Lemma *l8_5* : ∀ *A B*, *Per A B B*.

Lemma *l8_6* : ∀ *A B C A'*,
Per A B C → *Per A' B C* → *Bet A C A'* → *B=C*.

Lemma *l8_7* : $\forall\ A\ B\ C, Per\ A\ B\ C \rightarrow Per\ A\ C\ B \rightarrow B{=}C.$

Lemma *l8_8* : $\forall\ A\ B, Per\ A\ B\ A \rightarrow A{=}B.$

Lemma *l8_9* : $\forall\ A\ B\ C, Per\ A\ B\ C \rightarrow Col\ A\ B\ C \rightarrow A{=}B \vee C{=}B.$

Lemma *l8_10* : $\forall\ A\ B\ C\ A'\ B'\ C',$
$Per\ A\ B\ C \rightarrow Cong\_3\ A\ B\ C\ A'\ B'\ C' \rightarrow Per\ A'\ B'\ C'.$

Definition *Perp_in* := $fun\ X\ A\ B\ C\ D \Rightarrow$
$A{\neq}B \wedge C{\neq}D \wedge Col\ X\ A\ B \wedge Col\ X\ C\ D \wedge$
$(\forall\ U\ V, Col\ U\ A\ B \rightarrow Col\ V\ C\ D \rightarrow Per\ U\ X\ V).$

Definition *Perp* := $fun\ A\ B\ C\ D \Rightarrow \exists\ X, Perp\_in\ X\ A\ B\ C\ D.$

Lemma *l8_12* : $\forall\ A\ B\ C\ D\ X, Perp\_in\ X\ A\ B\ C\ D \rightarrow Perp\_in\ X\ C\ D\ A\ B.$

Lemma *l8_13_2* : $\forall\ A\ B\ C\ D\ X,$
$A \neq B \rightarrow C \neq D \rightarrow Col\ X\ A\ B \rightarrow Col\ X\ C\ D \rightarrow$
$(\exists\ U, \exists\ V : Point, Col\ U\ A\ B \wedge Col\ V\ C\ D \wedge U{\neq}X \wedge V{\neq}X \wedge Per\ U\ X\ V) \rightarrow$
$Perp\_in\ X\ A\ B\ C\ D.$

Definition *DistLn* := $fun\ A\ B\ C\ D \Rightarrow$
$(\exists\ X, Col\ X\ A\ B \wedge \neg Col\ X\ C\ D) \vee (\exists\ X, \neg\ Col\ X\ A\ B \wedge Col\ X\ C\ D).$

Lemma *l8_14_1* : $\forall\ A\ B, \neg\ Perp\ A\ B\ A\ B.$

Lemma *l8_14_2_1a* : $\forall\ X\ A\ B\ C\ D,$
$Perp\_in\ X\ A\ B\ C\ D \rightarrow Perp\ A\ B\ C\ D.$

Lemma *l8_14_2_1b* : $\forall\ X\ A\ B\ C\ D\ Y,$
$Perp\_in\ X\ A\ B\ C\ D \rightarrow Col\ Y\ A\ B \rightarrow Col\ Y\ C\ D \rightarrow X{=}Y.$

Lemma *l8_14_2_1b_bis* : $\forall\ A\ B\ C\ D\ X,$
$Perp\ A\ B\ C\ D \rightarrow Col\ X\ A\ B \rightarrow Col\ X\ C\ D \rightarrow Perp\_in\ X\ A\ B\ C\ D.$

Lemma *l8_14_2_2* : $\forall\ X\ A\ B\ C\ D,$
$Perp\ A\ B\ C\ D \rightarrow (\forall\ Y, Col\ Y\ A\ B \rightarrow Col\ Y\ C\ D \rightarrow X{=}Y) \rightarrow$
$Perp\_in\ X\ A\ B\ C\ D.$

Lemma *l8_14_3* : $\forall\ A\ B\ C\ D\ X\ Y,$
$Perp\_in\ X\ A\ B\ C\ D \rightarrow Perp\_in\ Y\ A\ B\ C\ D \rightarrow X{=}Y.$

Lemma *l8_15_1* : $\forall\ A\ B\ C\ X,$
$A{\neq}B \rightarrow Col\ A\ B\ X \rightarrow Perp\ A\ B\ C\ X \rightarrow$
$Perp\_in\ X\ A\ B\ C\ X.$

Lemma *l8_15_2* : $\forall\ A\ B\ C\ X,$
$A{\neq}B \rightarrow Col\ A\ B\ X \rightarrow Perp\_in\ X\ A\ B\ C\ X \rightarrow$
$Perp\ A\ B\ C\ X.$

Lemma *l8_16_1* : $\forall\ A\ B\ C\ U\ X$,
$A{\neq}B \rightarrow Col\ A\ B\ X \rightarrow Col\ A\ B\ U \rightarrow U{\neq}X \rightarrow Perp\ A\ B\ C\ X \rightarrow \neg\ Col\ A\ B\ C \wedge Per\ C\ X\ U$.

Lemma *l8_16_2* : $\forall\ A\ B\ C\ U\ X$,
$A{\neq}B \rightarrow Col\ A\ B\ X \rightarrow Col\ A\ B\ U \rightarrow U{\neq}X \rightarrow$
$\neg\ Col\ A\ B\ C \rightarrow Per\ C\ X\ U \rightarrow Perp\ A\ B\ C\ X$.

Lemma *l8_18_unicity* : $\forall\ A\ B\ C\ X\ Y$,
$\neg\ Col\ A\ B\ C \rightarrow$
$Col\ A\ B\ X \rightarrow Perp\ A\ B\ C\ X \rightarrow$
$Col\ A\ B\ Y \rightarrow Perp\ A\ B\ C\ Y \rightarrow$
$X{=}Y$.

Lemma *l8_18_existence* : $\forall\ A\ B\ C$,
$\neg\ Col\ A\ B\ C \rightarrow \exists\ X$,
$Col\ A\ B\ X \wedge Perp\ A\ B\ C\ X$.

Lemma *l8_20_1* : $\forall\ A\ B\ C\ C'\ D\ P$,
$Per\ A\ B\ C \rightarrow is\_midpoint\ P\ C'\ D \rightarrow$
$is\_midpoint\ A\ C'\ C \rightarrow is\_midpoint\ B\ D\ C \rightarrow$
$Per\ B\ A\ P$.

Lemma *l8_20_2* : $\forall\ A\ B\ C\ C'\ D\ P$,
$Per\ A\ B\ C \rightarrow is\_midpoint\ P\ C'\ D \rightarrow$
$is\_midpoint\ A\ C'\ C \rightarrow is\_midpoint\ B\ D\ C \rightarrow$
$B{\neq}C \rightarrow A{\neq}P$.

Lemma *l8_21_aux* : $\forall\ A\ B\ C$,
$A{\neq}\ B \rightarrow \neg\ Col\ A\ B\ C \rightarrow \exists\ P, \exists\ T, Perp\ A\ B\ P\ A \wedge Col\ A\ B\ T \wedge Bet\ C\ T\ P$.

Lemma *l8_21* : $\forall\ A\ B\ C$,
$A{\neq}\ B \rightarrow \exists\ P, \exists\ T, Perp\ A\ B\ P\ A \wedge Col\ A\ B\ T \wedge Bet\ C\ T\ P$.

Lemma *perp_symmetry* : $\forall\ A\ B\ C\ D, Perp\ A\ B\ C\ D \rightarrow Perp\ C\ D\ A\ B$.

Lemma *perp_commutativity* : $\forall\ A\ B\ C\ D, Perp\ A\ B\ C\ D \rightarrow Perp\ B\ A\ D\ C$.

Lemma *perp_left_commutativity* : $\forall\ A\ B\ C\ D,\ Perp\ A\ B\ C\ D \rightarrow Perp\ B\ A\ C\ D$.

Lemma *perp_right_commutativity* : $\forall\ A\ B\ C\ D,\ Perp\ A\ B\ C\ D \rightarrow Perp\ A\ B\ D\ C$.

Lemma *perp_in_left_commutativity* : $\forall\ A\ B\ C\ D\ X,$
$Perp\_in\ X\ A\ B\ C\ D \rightarrow Perp\_in\ X\ B\ A\ C\ D$.

Lemma *perp_in_right_commutativity* : $\forall\ A\ B\ C\ D\ X,$
$Perp\_in\ X\ A\ B\ C\ D \rightarrow Perp\_in\ X\ A\ B\ D\ C$.

Lemma *perp_in_commutativity* : $\forall\ A\ B\ C\ D\ X,$
$Perp\_in\ X\ A\ B\ C\ D \rightarrow Perp\_in\ X\ B\ A\ D\ C$.

Lemma *perp_in_symmetry* : $\forall\ A\ B\ C\ D\ X,$
$Perp\_in\ X\ A\ B\ C\ D \rightarrow Perp\_in\ X\ C\ D\ A\ B$.

Lemma *perp_per_1* : $\forall\ A\ B\ C,\ A{\neq}B \rightarrow Perp\ A\ B\ C\ A \rightarrow Per\ B\ A\ C$.

Lemma *perp_per_2* : $\forall\ A\ B\ C,\ A{\neq}B \rightarrow Perp\ A\ B\ A\ C \rightarrow Per\ B\ A\ C$.

Lemma *perp_col* : $\forall\ A\ B\ C\ D\ E,$
$A{\neq}E \rightarrow$
$Perp\ A\ B\ C\ D \rightarrow Col\ A\ B\ E \rightarrow Perp\ A\ E\ C\ D$.

Lemma *perp_not_eq_1* : $\forall\ A\ B\ C\ D,$
$Perp\ A\ B\ C\ D \rightarrow A{\neq}B$.

Lemma *perp_not_eq_2* : $\forall\ A\ B\ C\ D,$
$Perp\ A\ B\ C\ D \rightarrow C{\neq}D$.

Lemma *perp_perp_in* : $\forall\ A\ B\ C,$
$Perp\ A\ B\ C\ A \rightarrow Perp\_in\ A\ A\ B\ C\ A$.

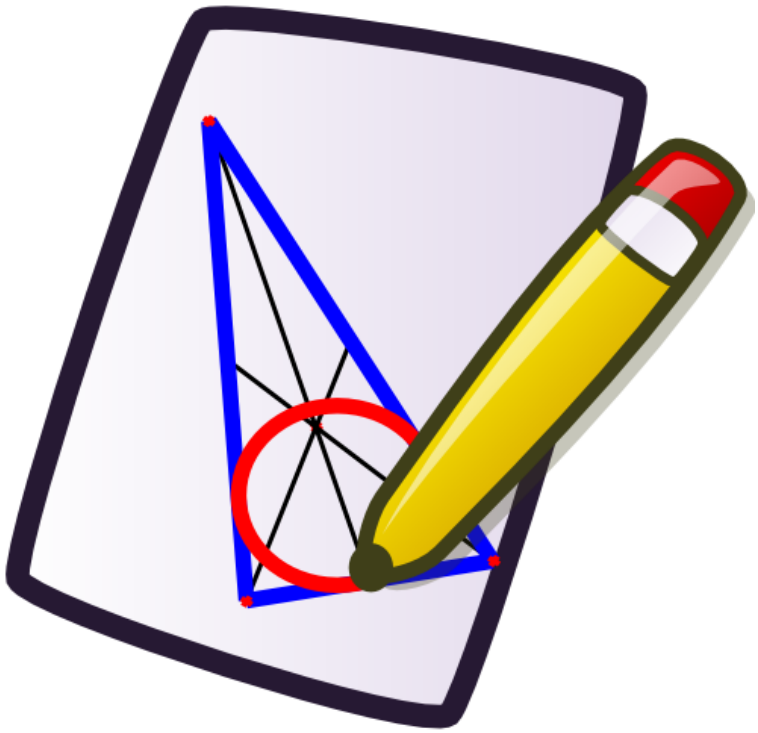
Lemma *midpoint_existence_aux* : $\forall\ A\ B\ P\ Q\ T,$
$A{\neq}B \rightarrow Perp\ A\ B\ Q\ B \rightarrow Perp\ A\ B\ P\ A \rightarrow$
$Col\ A\ B\ T \rightarrow Bet\ Q\ T\ P \rightarrow$
$le\ A\ P\ B\ Q \rightarrow$
$\exists\ X : Point,\ is\_midpoint\ X\ A\ B$.

Lemma *midpoint_existence* : $\forall\ A\ B, \exists\ X,\ is\_midpoint\ X\ A\ B$.

Annexe C

Manuel de référence de *GeoProof*

GeoProof



Manuel de référence

Copyright © 2006 Julien Narboux

Bienvenue dans le manuel de référence de *GeoProof*.

Ce manuel est composé de neuf chapitres :

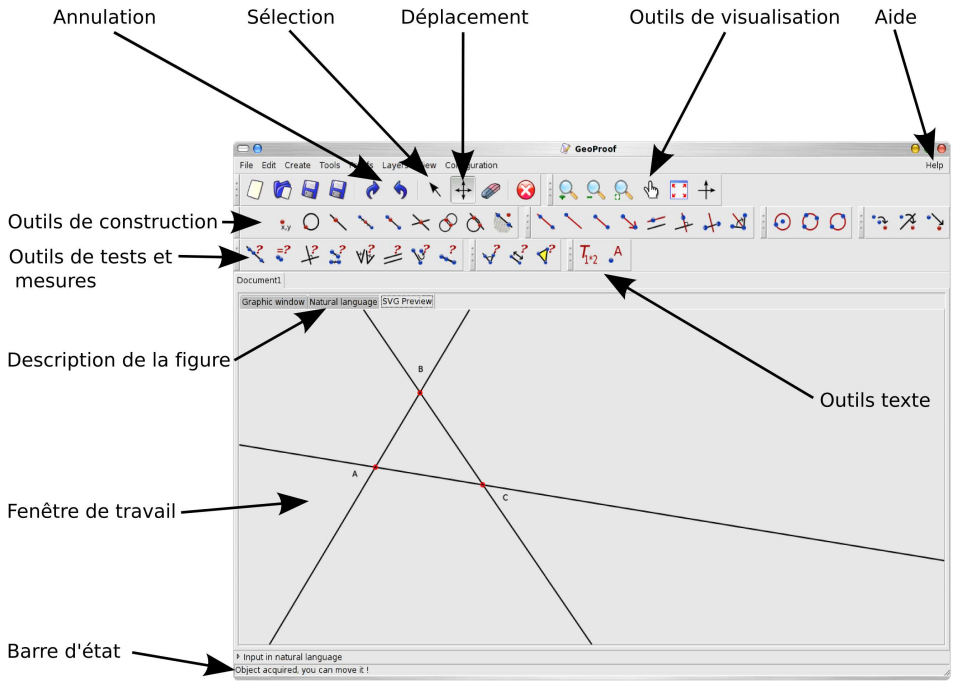
1. Le chapitre « **Installation** » décrit la procédure d'installation sous les différents systèmes supportés.
2. Le chapitre « **Outils de construction** » décrit les différents outils qui permettent de créer les figures dynamiques.
3. Le chapitre « **Outils d'exploration** » présente les diverses fonctionnalités qui peuvent être utilisées pour explorer une figure, réaliser des mesures et faire des conjectures.
4. Le chapitre « **Attributs** » décrit les attributs disponibles pour chaque type d'objet. Les attributs définissent l'aspect graphique des objets.
5. Le chapitre « **Outils de sélection** » décrit les différents modes de sélection des objets.
6. Le chapitre « **Préférences** » décrit comment l'utilisateur peut personnaliser certains paramètres du logiciel.
7. Le chapitre « **Importation/Exportation** » décrit les différents modes de lecture et d'écriture des figures.
8. Le chapitre « **Démonstration automatique** » décrit les fonctions liées au démonstrateur automatique intégré.
9. Le chapitre « **Démonstration interactive** » décrit les fonctions liées à l'interaction avec l'assistant de preuve Coq.

GeoProof est un logiciel libre de géométrie interactive. Il permet de créer des figures géométriques et de les manipuler à la souris. Avec *GeoProof* on peut étudier ce qui se passe lorsque l'on change la position des points libres de la figure et ainsi faire des conjectures. Il est alors possible d'exporter la conjecture dans l'assistant de preuve Coq afin de la prouver interactivement. *GeoProof* est distribué sous la license GPL Version 2.

Pour obtenir la dernière version de *GeoProof* ou nous faire part d'un *bug*, vous pouvez vous rendre sur le site :

`http://home.gna.org/geoproof/`

Avant d'entrer dans le détail des fonctionnalités de *GeoProof*. Voici un aperçu de son interface.



C.1 Installation

C.1.1 Windows

L'installation de *GeoProof* sous windows est très simple puisqu'il suffit de lancer le logiciel d'installation automatique. Après avoir choisi un emplacement pour installer *GeoProof*, vous aurez à choisir si vous désirez installer les fichiers *.dll* de GTK. Si vous n'êtes pas sûr de la réponse laissez cette option cochée puisque ces fichiers sont nécessaires au bon fonctionnement de *GeoProof*. Si vous êtes certain d'avoir déjà une version des *.dll* de GTK installée sur votre ordinateur vous pouvez désactiver cette option.

C.1.2 Linux

Sous linux, *GeoProof* peut être installé au moyen des fichiers `.rpm` fournis ou bien en le compilant soi-même à partir des sources. Pour plus d'information sur la procédure de compilation voir le fichier `INSTALL`.

C.1.3 MacOSX

GeoProof ne comporte par encore de procédure d'installation automatique pour MacOsX, il vous faudra donc compiler depuis les sources.

C.2 Outils de construction

Voici une liste des différents outils de construction de *GeoProof*. Afin de retenir plus facilement la signification de chacun des icônes souvenez vous que l'objet créé apparaît en rouge sur l'icône désignant l'outil. Les outils sont regroupés suivant le type d'objet qu'ils construisent.

Points		
Point libre		Crée un nouveau point libre à la souris.
Point libre (défini par ses coordonnées)		Crée un nouveau point libre en définissant ses coordonnées initiales.
Point sur cercle		Crée un point sur un cercle.
Point sur droite		Crée un point sur une droite.
Point sur segment		Crée un point sur un segment.
Point dans demi-plan		Crée un point dans un demi-plan.
Milieu d'un segment		Crée le milieu d'un segment.
Intersection droites		Crée l'intersection de deux droites.
Intersection cercles		Crée les intersections de deux cercles.
Intersection cercle-droite		Crée les intersections entre un cercle et une droite.

Droites		
Droite simple		Crée une droite passant par deux points.
Demi-droite		Crée une demie droite dont l'extrémité est le premier point passant par le deuxième.
Segment		Crée un segment dont les extrémités sont les deux points.
Vecteur		Crée un vecteur.
Droite parallèle		Crée la droite parallèle à une autre passant par un point.
Droite perpendiculaire		Crée la droite perpendiculaire à une autre passant par un point.
Médiatrice		Crée la médiatrice d'un segment.
Bissectrice		Crée la bissectrice de l'angle formé par trois points.

Cercles		
Cercle centre-point		Crée un cercle de centre le premier point passant par le second point.
Cercle trois points		Crée un cercle passant par trois points.
Cercle diamètre		Crée un cercle selon son diamètre.

Transformations		
Symétrie centrale		Crée un objet par symétrie par rapport à un point.
Symétrie axiale		Crée un objet par symétrie par rapport à une droite.
Translation		Crée un objet par translation par rapport à un vecteur.

Textes		
Texte libre		Crée un label qui peut être déplacé librement sur la feuille.
Texte lié		Crée un label dont la position est relative à celle d'un point.

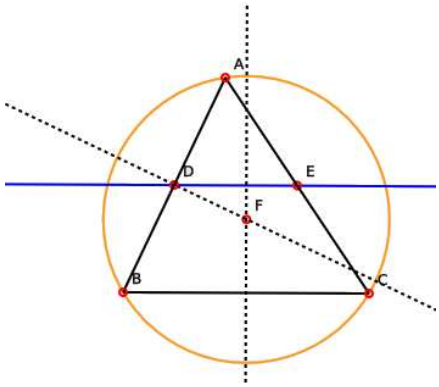
Outils d'édition		
Sélection		Permet de sélectionner des objets.
Interrompre		Annule la construction en cours.
Annuler		Annule la dernière construction/suppression.
Refaire		Refait la dernière action annulée.
Supprimer		Supprime un objet ainsi que tous les objets qui en dépendent.

C.3 Outils d'exploration

Visualisation		
Déplacement		Permet de déplacer les objets libres ou semi-libres
Zoom Avant		Effectue un grossissement d'un facteur 2.
Zoom Arrière		Effectue un grossissement d'un facteur $\frac{1}{2}$.
Zoom Automatique		Ajuste le zoom de telle manière que tous les objets de taille finie soient visibles en entier.
Déplace la feuille		Permet de déplacer la feuille à la souris.
Mode plein écran		Active ou désactive le mode plein écran.
Repère		Cache ou montre le repère.

Propriétés		
Collinéaire		Test si trois points sont colinéaires (sur cette instance de la figure). Crée un label qui est mis à jour en temps réel.
Égalité		Test si deux points sont confondus.
Entre deux		Test si un point appartient à un segment.
A gauche de		Test si un point est dans un demi-plan.
Parallèle		Test si deux droites sont parallèles.
Perpendiculaire		Test si deux droites sont perpendiculaires.
Congruence des segments		Test si deux segments ont même longueur.
Congruence des angles		Test si deux angles ont même mesure.

Vue en SVG *Vue en langue naturelle*



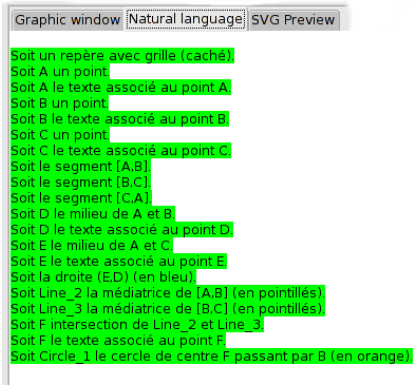


FIG. C.1 – Exemple de traduction en langue naturelle

Mesures		
Angle		Mesure l'angle formé par trois points, crée un label qui est mis à jour en temps réel.
Distance		Mesure la distance entre deux points.
Aire		Mesure l'aire d'un triangle.

C.3.1 Description en langue (pseudo-)naturelle

La fenêtre principale de *GeoProof* propose trois onglets. L'un d'entre eux permet d'avoir une description de la figure en langue naturelle. La figure C.1 montre un exemple de description.

C.3.2 Expressions

Les labels textuels peuvent contenir des champs dynamiques. Les champs dynamiques contiennent une expression qui est évaluée en temps réel lorsque la figure est déplacée. Les champs dynamiques sont délimités par le signe `#`.

Par exemple pour créer un label qui compare les surfaces des triangles ABC et DEF, vous pouvez rentrer le texte suivant :

```
Le triangle ABC est plus #if area(A,B,C)>area(D,E,F) then
"grand" else "petit"# que le triangle DEF.
```

Cette fonctionnalité peut aussi faire office de calculatrice si l'expression arithmétique ne dépend d'aucun élément de la figure. Mais *GeoProof* fournit une calculatrice qui calcule en précision arbitraire grâce à la bibliothèque

`creal` de Jean-Christophe Filliâtre. La précision des calculs est paramètrable dans le fichier de configuration de *GeoProof* (voir page 183). Ainsi, si on essaie de calculer $\sin(10^{22})$, on peut saisir le texte `#sin(10^22)#` et on obtient -0.8522008498 (si *GeoProof* est configuré pour une précision de 10 décimales).

Cet exemple est connu pour mettre en défaut de nombreuses « calculatrices ». La tableau suivant donne les résultats[1] du même calcul réalisé dans différents systèmes.

Système	$\sin(10^{22})$
Réponse correcte	$-0.852200849\ldots$
KCalc	$+0,462613041$
Google	$+0,462613041$
Scilab 3.0	$+0.4626130$
MPFR	$-0.852200849\ldots$
Mupad	-0.9873536182
Maple 8 (15 digs)	$-0.852200849\ldots$
Maxima 5.9 (bfloat)	$-0.852200849\ldots$
Matlab 6.5 (15 digs)	$-0.852200849\ldots$
O-Matrix 5.5(e format)	$+0.226946577\ldots$
O-Matrix 5.5(d format)	$+0.412143367\ldots$
Scilab 3.0 (15 digs)	$+10^{22}$
DVF 5.0 D (sp)	$+0.2269466\ldots$
DVF 5.0 D (dp)	$+0.412143367\ldots$
Intel Fortran 8 (sp)	$+9.9999998 * 10^{21}$
Intel Fortran 8 (dp)	$+10^{22}$
Intel Fortran 8 (ep)	$-0.852200849\ldots$
Watfor 11.2 (sp)	$+0.2812271\ldots$
Watfor 11.2 (dp)	$+0.4626130\ldots$
TMT Pascal(all precs)	$+0.0$
FranzLisp	$+0.2269465\ldots$
Sharp EL-531VH	$error2$
MS Windows Calc (32 digs)	$-0.852200849\ldots$
PariGP 2.2.7 (28 digs)	$-0.852200849\ldots$
HP 48 GX	$-0.852200849\ldots$
HP 700	0.0
IBM 3090/600S-VF AIX 370	0.0
Matlab 4.2c.1 Sparc	-0.8522
Matlab 4.2c.1 MacIntosh	0.8740
SG Indy	0.87402806
Sharp EL5806	-0.090748172
DEC Station 3100	NaN

[1] Sources : Derek O'Connor, University College, Floating Point Arithmetic or You can't always count on your computer et expériences personnelles.

Fonction	Syntaxe
Expressions arithmétiques	`+ - * / ^`
Sinus	`sin`
Cosinus	`cos`
Tangente	`tan`
Arcsinus	`arcsin`
Arccosinus	`arccos`
Arctangente	`arctan`
Racine carrée	`sqrt`
Exponentielle	`exp`
Logarithme néperien	`ln`
Logarithme en base 10	`log`
Puissance	`pow`
Valeur absolue	`abs`
Minimum	`min`
Maximum	`max`
π	`pi`
e	`e`
Mesures	`signed_area area angle length`
Tests	`parallel orthogonal eq_lengths eq_angles between collinear left_turn`
Opérateurs de comparaisons	`< > <= >= = <>`
Connecteurs logiques	`and or not`
Constantes logiques	`true false`
Chaîne de caractères	`"texte"`
Définition locale	`let id = expr in expr`
Condition	`if cond then expr else expr`

C.3.3 Punaise

Il est possible de décider de fixer un point libre ou semi-libre. Les points fixés ne peuvent pas être déplacés. Cette option peut être utile à un enseignant qui prépare un exercice, il peut ainsi limiter les intéractions possibles avec la figure.

C.3.4 Trace

Si l'option trace est activée, l'objet laisse une trace de son apparence quand il bouge. Cet attribut permet, par exemple, de se faire une idée des différentes positions que peut prendre un point lorsque un autre est déplacé. La figure C.2 montre l'exemple de l'étude du lieu du point E milieu du segment $[AD]$ quand D parcourt le cercle de centre B passant par C.

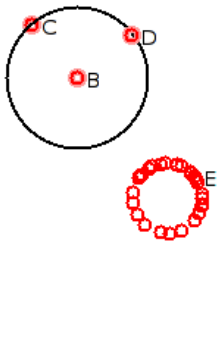


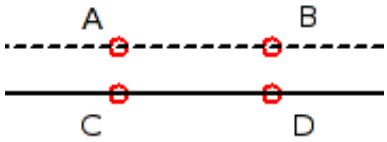
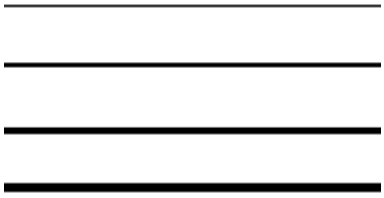
FIG. C.2 – Utilisation de l'outil « trace »

C.4 Attributs

Chaque objet *GeoProof* possède des attributs qui définissent son aspect graphique. A sa création, un objet possède les attributs par défaut définis dans le menu « configuration » de *GeoProof*. Pour changer les attributs d'un objet, il existe deux méthodes :

- utiliser le menu contextuel de l'objet en réalisant un clique droit sur l'objet ou
- sélectionner les objets dont on désire changer les attributs, puis les changer au moyen du menu « édition »

Les attributs possibles sont les suivants :

Attribut	Explication
Couleur	Les couleurs des objets peuvent être les suivantes : noir, gris, bleu, rouge, vert, jaune, orange, violet, rose
Style de trait	Les traits peuvent être pleins ou en pointillés : 
Épaisseur	Il y a quatre épaisseurs de trait possibles : 
Aspect	Les points peuvent avoir les aspects suivants :
Visible/Invisible	Il est possible de cacher temporairement un objet sans le supprimer pour autant. On peut ainsi simplifier une figure, par exemple en omettant les étapes de construction.
Calque	Les objets appartiennent à un calque, chaque calque peut être visible ou caché et avoir une couleur. Si un calque possède une couleur les objets du calque sont affichés avec cette couleur.

C.5 Outils de sélection

La sélection de plusieurs objets permet d'en changer les attributs en une seule opération. Les objets sélectionnés apparaissent entourés d'un halo. Les outils de sélection suivants sont disponibles via le menu « édition » :

Sélection simple Permet de sélectionner un ou plusieurs objets à la souris en cliquant sur les objets. Cliquer à nouveau sur un objet le désélectionne.

Sélection par type d'objet Avec ce mode il est possible de sélectionner tous les objets d'un même type. Les types possibles sont les suivants :

- Objets visibles
- Objets cachés
- Droites
- Segments
- Points
- Vecteurs
- Objets libres

Aucun Desélectionne tous les objets.

Tous Sélectionne tous les objets.

Inverser Inverse les objets sélectionnés et désélectionnés.

C.6 Préférences

Il est possible de modifier certains paramètres de *GeoProof* grâce au menu « Configuration ». D'autres paramètres sont modifiables uniquement en éditant le fichier de configuration (`.geoproof/config.ini`) qui apparaît dans le répertoire de chacun des utilisateurs.

Les paramètres suivants sont disponibles via le menu « Configuration » :

- unités de distance et d'angle,
- couleur du fond,
- attributs par défaut des objets crées,
- visibilité des barres d'outils.

Les paramètres suivants sont modifiables via le fichier de configuration :

Paramètres	
`*_name_prefix`	préfixe utilisé pour la génération automatique des noms des objets
`selection_precision`	la tolérance (en nombre de pixels) du mécanisme de sélection
`automatic_point_labeling`	ajout automatique ou non d'un label lors de la création des points
`number_of_recent_files`	nombre de fichiers récents à faire apparaître dans le menu « fichier »
`icons_size`	taille des icônes (en nombre de pixels)
`precision`	précision des calculs réalisés (en nombre de décimales)
`formal_language_for_Coq_export`	langage à utiliser pour la communication avec Coq

C.7 Importation/Exportation

Il est possible d'exporter les figures créées en utilisant *GeoProof* en trois types de formats :

une image en mode points (ou image *bitmap*) au format PNG, JPEG ou BMP. Ce mode d'exportation réalise l'équivalent d'une capture d'écran, la figure peut ainsi être facilement intégrée dans un traitement de texte, ou une page web. Le format PNG est un format libre de droits qui fournit une compression sans pertes. C'est le plus adapté au graphismes avec des aplats comme les figures géométriques, mais il a l'inconvénient que la transparence n'est pas bien gérée par les versions anciennes de Internet Explorer (<7.0). Le format JPEG produit des images moins lourdes mais la compression n'est pas bien adaptée aux figures géométriques et réduit sensiblement la qualité de l'image produite.

une image en mode vectoriel au format SVG. Le format SVG est le format standard pour le graphisme en mode vectoriel défini par le W3C. Ce format est géré par Firefox (à partir de la version 1.5) et Konqueror et sera bientôt géré aussi par Safari. Il est possible d'ajouter le support SVG à Internet Explorer au moyen d'un plugin.

un script dans le langage *Eukleides* pour insertion de figures dans un document LaTeX. Le langage *Eukleides*[2] permet de réaliser une description de haut niveau de la figure qui sera insérée directement dans un document LaTeX. Cette description est facile à éditer en mode texte. Ainsi pour modifier une figure, il n'est pas nécessaire de l'ouvrir à nouveau avec *GeoProof*. Les labels textuels peuvent contenir du code LaTeX. La figure C.3 montre un exemple de script généré.

GeoProof permet aussi d'importer des figures générées par les logiciels *Kig*[3] (KDE interactive geometry) et *CaR*[4]. Cette fonctionnalité est pour l'instant *expérimentale*.

C.8 Démonstration automatique

Pour prouver un énoncé au moyen du démonstrateur automatique intégré, il suffit de le lancer via le menu « preuve ». Il est aussi possible de le lancer en effectuant un clique droit sur un label comportant un champ dynamique avec un `if`. La condition du `if` est alors traduite en un prédicat qui devient la conclusion du théorème à prouver.

[2] `http://www.eukleides.org/`
[3] `http://edu.kde.org/kig/`
[4] `http://mathsrv.ku-eichstaett.de/MGF/homes/grothman/java/zirkel/doc_en/`

```
frame(-10.00000,6.00000,12.48000,-3.90000,0.93416)
A = point(-3.22000,4.30000)
color(red)
thickness(2)
draw(A,dot)
color(black)
draw("A",A,0.28000,arg(circle(A,1),point(1.40000,1.40000)):)
...
E = barycenter(A,C)
color(red)
thickness(2)
draw(E,dot)
color(black)
draw("E",E,0.28000,arg(circle(E,1),point(1.40000,1.40000)):)
Segment_1 = segment(A,B)
color(black)
thickness(2)
draw(Segment_1,full)
...
Segment_3 = segment(C,A)
color(black)
thickness(2)
draw(Segment_3,full)
Line_1 = line(D,E)
color(blue)
thickness(2)
draw(Line_1,dashed)
```

FIG. C.3 – Extrait d'un script obtenu après exportation dans le langage d'*Eukleides*.

C.9 Démonstration interactive

Pour passer en mode démonstration interactive, il suffit de lancer CoqIDE avec l'option `-with-geoproof` et *GeoProof* sur le même ordinateur et d'initier la communication via le menu « preuve ».

Les définition des objets déjà présents dans *GeoProof* est traduite en syntaxe Coq.

Il est ensuite possible de réaliser une conjecture. Pour cela il faut créer un label dynamique qui teste une propriété (le fait que trois points sont collinéaires par exemple). En utilisant le menu contextuel du label (accessible par un clique droit), le prédicat correspondant à la propriété est traduit en syntaxe Coq et exporté dans CoqIDE.

GeoProof passe alors en mode preuve. Dans ce mode, quand un nouvel objet est créé dans *GeoProof*, une tactique est générée dans CoqIDE, celle-ci tente de prouver que le nouvel objet existe. Il est laissé à la charge de l'utilisateur de prouver les conditions de non dégénérescence de la construction utilisée.

Annexe D

Réécriture abstraite

Cette annexe contient les énoncé prouvés concernant la formalisation du système de preuve diagrammatique présenté au chapitre 6.

Julien Narboux
A diagrammatic formalization of abstract rewriting

Relation definition

Definition $relation := A \Rightarrow A \Rightarrow Prop.$

Definition $reflexive : Prop := \forall\ x : A,\ R\ x\ x.$
Definition $transitive : Prop := \forall\ x\ y\ z : A,\ R\ x\ y \Rightarrow R\ y\ z \Rightarrow R\ x\ z.$
Definition $symmetric : Prop := \forall\ x\ y : A,\ R\ x\ y \Rightarrow R\ y\ x.$
Definition $antisymmetric : Prop := \forall\ x\ y : A,\ R\ x\ y \Rightarrow R\ y\ x \Rightarrow x = y.$

Definition $equiv := reflexive \wedge transitive \wedge symmetric.$

Composition of two relations

Definition $composition : relation \Rightarrow relation \Rightarrow relation :=$
$fun\ R1\ R2 : relation \Rightarrow fun\ x\ y : A \Rightarrow \exists\ z : A,\ R1\ x\ z \wedge R2\ z\ y.$

Union of two relations

Definition $union : relation \Rightarrow relation \Rightarrow relation :=$
$fun\ R1\ R2 : relation \Rightarrow fun\ x\ y : A \Rightarrow R1\ x\ y \vee R2\ x\ y.$

Definition of swappable and commutation
The two definitions appear under the name of "commutation" in the literature

Definition $swappable : Prop := \forall\ x\ y\ z : A,\ R\ x\ y \Rightarrow S\ y\ z \Rightarrow \exists\ w,\ S\ x\ w \wedge R\ w\ z.$

Definition $commutation : Prop := \forall\ x\ y\ z : A,\ R\ x\ y \Rightarrow S\ x\ z \Rightarrow \exists\ w,\ S\ y\ w \wedge R\ z\ w.$

Reflexive closure

Definition $Roreq := fun\ R : relation \Rightarrow fun\ x\ y \Rightarrow x{=}y \vee R\ x\ y.$

Symmetric closure

Definition $Rlr := fun\ R : relation \Rightarrow fun\ x\ y \Rightarrow R\ x\ y \vee R\ y\ x.$

Symmetric transitive closure

Inductive $Rstar : A{\Rightarrow}A{\Rightarrow}Prop :=$
$|Rstar\_cont\_eq : \forall\ x,\ Rstar\ x\ x$
$|Rstar\_induction : \forall\ x\ y\ z,\ R\ x\ y{\Rightarrow}\ Rstar\ y\ z{\Rightarrow}\ Rstar\ x\ z.$

Transitive closure

Inductive $Rplus : A{\Rightarrow}A{\Rightarrow}Prop :=$
$|Rplus\_cont\_R : \forall\ x\ y,\ R\ x\ y \Rightarrow Rplus\ x\ y$
$|Rplus\_induction : \forall\ x\ y\ z,\ R\ x\ y{\Rightarrow}\ Rplus\ y\ z{\Rightarrow}\ Rplus\ x\ z.$

Reflexive, transitive, symmetric closure

Inductive $Rlrstar : A{\Rightarrow}A{\Rightarrow}Prop :=$
$|Rlrstar\_cont\_R : \forall\ x\ y,\ R\ x\ y \Rightarrow Rlrstar\ x\ y$
$|Rlrstar\_cont\_R\_rev : \forall\ x\ y,\ R\ y\ x \Rightarrow Rlrstar\ x\ y$
$|Rlrstar\_cont\_eq : \forall\ x,\ Rlrstar\ x\ x$
$|Rlrstar\_induction : \forall\ x\ y\ z,\ Rlrstar\ x\ y{\Rightarrow}\ Rlrstar\ y\ z{\Rightarrow}\ Rlrstar\ x\ z.$

Definition $joinable := fun\ R : relation \Rightarrow fun\ x\ y \Rightarrow \exists\ z,\ (Rstar\ x\ z) \wedge (Rstar\ y\ z).$

Definition $church\_rosser := \forall\ x\ y,\ Rlrstar\ x\ y \Rightarrow joinable\ R\ x\ y.$

Path of length n

Inductive $Rn : nat \Rightarrow A \Rightarrow A \Rightarrow Prop :=$
$|Rn\_0 : \forall\ x,\ Rn\ 0\ x\ x$
$|Rn\_1 : \forall\ x\ y,\ R\ x\ y \Rightarrow Rn\ 1\ x\ y$
$|Rn\_induction : \forall\ x\ y\ z\ n,\ R\ x\ y{\Rightarrow}\ Rn\ n\ y\ z{\Rightarrow}\ Rn\ (n{+}1)\ x\ z.$

Definition $weak\_commutation\_in\ (x : A) :=$
$\forall\ y\ z : A,\ R\ x\ y \Rightarrow S\ x\ z \Rightarrow \exists\ t,\ S\ y\ t \wedge R\ z\ t.$

Definition $confluence\_in\ (x : A) :=$
$\forall\ y\ z : A,\ Rstar\ x\ y \Rightarrow Rstar\ x\ z \Rightarrow joinable\ R\ y\ z.$

Definition $local\_confluence\_in\ (x : A) :=$
$\forall\ y\ z : A,\ R\ x\ y \Rightarrow R\ x\ z \Rightarrow joinable\ R\ y\ z.$

Definition $strong\_confluence\_in\ (x : A) :=$
$\forall\ y\ z : A,\ R\ x\ y \Rightarrow R\ x\ z \Rightarrow \exists\ t,\ Roreq\ R\ y\ t \wedge Rstar\ z\ t.$

Definition $semi\_confluence\_in\ (x : A) :=$
$\forall\ y\ z : A,\ R\ x\ y \Rightarrow Rstar\ x\ z \Rightarrow joinable\ R\ y\ z.$

Definition $diamond\_property\_in\ (x : A) :=$
$\forall\ y\ z : A,\ R\ x\ y \Rightarrow R\ x\ z \Rightarrow \exists\ t,\ R\ y\ t \wedge R\ z\ t.$

Definition $confluence := \forall\ x,\ confluence\_in\ x.$
Definition $local\_confluence := \forall\ x,\ local\_confluence\_in\ x.$
Definition $semi\_confluence := \forall\ x,\ semi\_confluence\_in\ x.$
Definition $strong\_confluence := \forall\ x,\ strong\_confluence\_in\ x.$
Definition $diamond\_property := \forall\ x, diamond\_property\_in\ x.$

Definition $isNF := fun\ x \Rightarrow \neg\ (\exists\ y,\ R\ x\ y).$
Definition $isWN := fun\ x \Rightarrow \exists\ y,\ Rstar\ x\ y \wedge isNF\ y.$

Definition $reduces\_to\_normal\_form := fun\ x\ y \Rightarrow Rstar\ x\ y \wedge isNF\ y.$

Definition $WN := fun\ R : relation \Rightarrow \forall\ x,\ isWN\ x.$

Inductive $isSN : A \Rightarrow Prop :=$
$|SN\_base : \forall\ x,\ isNF\ x \Rightarrow isSN\ x$
$|SN\_induction : \forall\ x,\ (\forall\ y,\ R\ x\ y \Rightarrow isSN\ y) \Rightarrow isSN\ x.$

Definition $SN := \forall\ x,\ isSN\ x.$

Definition $noetherian :=$
$\forall\ (P : A \Rightarrow Prop),$
$(\forall\ y : A,\ (\forall\ z : A,\ R\ y\ z \Rightarrow P\ z) \Rightarrow P\ y) \Rightarrow \forall\ x : A,\ P\ x.$

Definition $convergent := confluence \wedge SN.$

Lemma $SN\_implies\_noetherian : SN \Rightarrow noetherian.$

Tactics for diagrammatic-like proofs

These tactics uses the implicit rules contained in the Rules hints base.

Infix "" $:= R\ (at\ level\ 50,\ no\ associativity).$

Notation "$x \xrightarrow{*} y$" $:= (Rstar\ S\ R\ x\ y)\ (at\ level\ 50,\ no\ associativity).$
Notation "$x \xrightarrow{+} y$" $:= (Rplus\ S\ R\ x\ y)\ (at\ level\ 50,\ no\ associativity).$
Notation "$x \xrightarrow{=?} y$" $:= (Roreq\ S\ R\ x\ y)\ (at\ level\ 50,\ no\ associativity).$

Lemma $Rstar\_cont\_R : \forall\ x\ y, (x\ \ y) \Rightarrow x \xrightarrow{*} y.$

Lemma $Rstar\_transitivity : \forall\ x\ y\ z : S,\ x \xrightarrow{*} y \Rightarrow y \xrightarrow{*} z \Rightarrow x \xrightarrow{*} z.$

Lemma $Rstar\_cases : \forall\ x\ y,\ x \xrightarrow{*} y \Rightarrow x{=}y \vee \exists\ y',\ x\ \ y' \wedge y' \xrightarrow{*} y.$

Lemma $Rplus\_transitivity : \forall\ x\ y\ z : S,\ x \xrightarrow{+} y \Rightarrow y \xrightarrow{+} z \Rightarrow x \xrightarrow{+} z.$

Lemma $Rstar\_cont\_Rplus : \forall\ x\ y : S,\ x \xrightarrow{+} y \Rightarrow x \xrightarrow{*} y.$

Lemma $Rplus\_cases : \forall\ x\ y,\ x \xrightarrow{+} y \Rightarrow x\ \ y \vee \exists\ y',\ x\ \ y' \wedge y' \xrightarrow{+} y.$

Lemma $Rstar\_R\_Rstar : \forall\ x\ y\ z,\ x \xrightarrow{*} y \Rightarrow y\ \ z \Rightarrow x \xrightarrow{*} z.$

Lemma $Rplus\_R\_Rplus : \forall\ x\ y\ z, x \xrightarrow{+} y \Rightarrow y\ \ z \Rightarrow x \xrightarrow{+} z.$

Lemma $Rplus\_Rstar\_Rplus : \forall\ x\ y\ z, x \xrightarrow{+} y \Rightarrow y \xrightarrow{*} z \Rightarrow x \xrightarrow{+} z.$

Lemma $R\_Rstar\_Rplus : \forall\ x\ y\ z, x\ \ y \Rightarrow y \xrightarrow{*} z \Rightarrow x \xrightarrow{+} z.$

Lemma $Rstar\_cases\_2 : \forall\ x\ y, x \xrightarrow{*} y \Rightarrow x{=}y \vee x \xrightarrow{+} y.$

Lemma $Rstar\_finite\_path : \forall\ x\ y, x \xrightarrow{*} y \Rightarrow \exists\ n, Rn\ S\ R\ n\ x\ y.$

Lemma $aux : \forall\ x\ :nat, x{+}1 > 0.$

Lemma $aux2 : \forall\ x\ :nat, x{+}1 \neq 0.$

Lemma $Rplus\_finite\_path\_non\_null : \forall\ x\ y, x \xrightarrow{+} y \Rightarrow \exists\ n, n{\neq}0 \wedge Rn\ S\ R\ n\ x\ y.$

Lemma $SN\_Rplus\_R : SN\ S\ (Rplus\ S\ R) \Rightarrow SN\ S\ R.$

Theorem *newman* :
$local\_confluence\ S\ R \Rightarrow noetherian\ S\ R \Rightarrow confluence\ S\ R.$
induction

First degenerated case

Second degenerated case

General case

A second version of the Newman's lemma using implicit star_transitivity...

Theorem *newman_shorter* :
$local\_confluence\ S\ R \Rightarrow noetherian\ S\ R \Rightarrow confluence\ S\ R.$
induction

First degenerated case

Second degenerated case

General case

Lemma $Rlrstar\_cont\_Rstar : \forall\ x\ y, x \xrightarrow{*} y \Rightarrow Rlrstar\ S\ R\ x\ y.$

Lemma $Rlrstar\_symmetry : \forall\ x\ y, Rlrstar\ S\ R\ x\ y \Rightarrow Rlrstar\ S\ R\ y\ x.$

Lemma $R\_R\_Rlrstar : \forall\ x\ y\ z, x \xrightarrow{*} y \Rightarrow x \xrightarrow{*} z \Rightarrow Rlrstar\ S\ R\ y\ z.$

Lemma $church\_rosser\_confluence : church\_rosser\ S\ R \Rightarrow confluence\ S\ R.$

Lemma $confluence\_church\_rosser : confluence\ S\ R \Rightarrow church\_rosser\ S\ R.$

Lemma $confluence\_semi\_confluence : confluence\ S\ R \Rightarrow semi\_confluence\ S\ R.$

Lemma *semi_confluence_confluence* : *semi_confluence S R* ⇒ *confluence S R*.

Lemma *Roreq_cases* : ∀ *x y*, *Roreq S R x y* ⇒ *x* = *y* ∨ *x y*.

Lemma *strong_confluence_semi_confluence* : *strong_confluence S R* ⇒ *semi_confluence S R*.

Theorem *strong_confluence_confluence* : *strong_confluence S R* ⇒ *confluence S R*.

Lemma *confluence_normal_form_1* :
confluence S R ⇒ ∀ *x y*, *Rlrstar S R x y* ⇒ *isNF S R y* ⇒ *xâĘŠ× y*.

Lemma *confluence_normal_form_2* :
confluence S R ⇒ ∀ *x y*, *Rlrstar S R x y* ⇒ *isNF S R x* ⇒ *isNF S R y* ⇒
x = *y*.

Lemma *confluence_normal_form_unicity* : *confluence S R* ⇒ ∀ *x y y'*,
reduces_to_normal_form S R x y ⇒ *reduces_to_normal_form S R x y'* ⇒ *y=y'*.

Infix "*Âř*" := *composition* (*at level* 60, *no associativity*).

Notation "$x \xrightarrow{a} y$" := (*R1 x y*) (*at level* 50, *no associativity*).
Notation "$x \xrightarrow{b} y$" := (*R2 x y*) (*at level* 50, *no associativity*).
Notation "$x \xrightarrow{a}{}^{*} y$" := (*Rstar S R1 x y*) (*at level* 50, *no associativity*).
Notation "$x \xrightarrow{b}{}^{*} y$" := (*Rstar S R2 x y*) (*at level* 50, *no associativity*).
Notation "$x \xrightarrow{a\cup b}{}^{*} y$" := (*Rstar S* (*union S R1 R2*) *x y*) (*at level* 50, *no associativity*).
Notation "$x \xrightarrow{a\cup b} y$" := (*union S R1 R2 x y*) (*at level* 50, *no associativity*).
Theorem *example1* :
transitive R1 ⇒ *transitive R2* ⇒ *swappable S R2 R1* ⇒ *transitive* (*R1 Âř R2*).

Lemma *union_R* : ∀ *x y*, $x \xrightarrow{a} y \Rightarrow x \xrightarrow{a\cup b} y$.

Lemma *union_R_rev* : ∀ *x y*, $x \xrightarrow{b} y \Rightarrow x \xrightarrow{a\cup b} y$.

Lemma *union_Rstar* : ∀ *x y*, $x \xrightarrow{a}{}^{*} y \Rightarrow x \xrightarrow{a\cup b}{}^{*} y$.

Lemma *cases_union* : ∀ *x y*, $x \xrightarrow{a\cup b} y \Rightarrow x \xrightarrow{a} y \vee x \xrightarrow{b} y$.

Lemma *union_Rstar_rev* : ∀ *x y*, $x \xrightarrow{b}{}^{*} y \Rightarrow x \xrightarrow{a\cup b}{}^{*} y$.

Lemma *union_star_star_union* :
∀ *x y*, *Rstar S* (*union S R1 R2*) *x y* ⇒ *Rstar S* (*union S* (*Rstar S R1*) (*Rstar S R2*)) *x y*.

Lemma *commutation_union_simple* :

confluence S R1 $\Rightarrow$
commutation (*Rstar S R1*) (*Rstar S R2*) $\Rightarrow$
$\forall\ x\ y\ z,\ x \xrightarrow{a} \star\ y \Rightarrow x \xrightarrow{a \cup b}{}^{*}\ z \Rightarrow \exists\ t,\ y \xrightarrow{a \cup b}{}^{*}\ t \wedge z \xrightarrow{a \cup b}{}^{*}\ t.$

Lemma *confluence_star_eq_confluence* :
confluence S R1 $\Rightarrow$
$(\forall\ x\ y,\ x \xrightarrow{a}{}^{*}\ y \Rightarrow x \xrightarrow{b}{}^{*}\ y) \Rightarrow$
$(\forall\ x\ y,\ x \xrightarrow{b}{}^{*}\ y \Rightarrow x \xrightarrow{a}{}^{*}\ y) \Rightarrow$
confluence S R2.

Lemma *inclusion_start_eq* :
$(\forall\ x\ y,\ x \xrightarrow{a}\ y \Rightarrow x \xrightarrow{b}\ y) \Rightarrow$
$(\forall\ x\ y,\ x \xrightarrow{b}\ y \Rightarrow x \xrightarrow{a}{}^{*}\ y) \Rightarrow$
$(\forall\ x\ y,\ x \xrightarrow{a}{}^{*}\ y \Rightarrow x \xrightarrow{b}{}^{*}\ y) \wedge$
$(\forall\ x\ y,\ x \xrightarrow{b}{}^{*}\ y \Rightarrow x \xrightarrow{a}{}^{*}\ y).$

Table des figures

1 Tous les triangles sont isocèles 3

1.1 Axiome de Pasch 13
1.2 Diagrame réprésentant les inclusions entres modèles 28

2.1 Axiome de construction d'un segment 32
2.2 Axiome des cinq segments 33
2.3 Axiomes de Pasch 34
2.4 Axiomes d'Euclide 37
2.5 Transitivité et pseudo-transitivités 39
2.6 Preuve de l'axiome 18 44

4.1 La réflexion. 67
4.2 Graphe de dépendances entre les modules 71
4.3 Le théorème de Céva 79
4.4 Le théorème de Ménélaus 80
4.5 Le théorème de Pascal 81
4.6 Le théorème de Pappus 82
4.7 Le théorème de Desargues 83
4.8 Le théorème du centre de gravité 84
4.9 Le théorème de la droite de Gauss 85

5.1 Les points d'intérêt d'un triangle 94
5.2 Théorème de la droite des milieux 100
5.3 Preuve d'une propriété avec des hypothèses contradictoires . . 101
5.4 Le théorème de la droite des milieux (procédure de décision) 102
5.5 Point d'intersection de deux droites 105
5.6 Le théorème de la droite des milieux (forme interactive) . . . 105
5.7 Intégration de *GeoProof* dans l'architecture de *Proof General* 106

C.1 *GeoProof* : Exemple de traduction en langue naturelle 178
C.2 *GeoProof* : Utilisation de l'outil « trace » 181
C.3 *GeoProof* : Export dans le langage *Eukleides* 185

Index

Symboles

`apply` 124
`conclusion` 125
`cut` 126
`intros` 124
`reflexivity` 125
`substitute` 125

A

aire orientée 57
Archimède, axiome d' 13
axiomatique
    Chou, Gao et Zhang 24
    Heyting 17
    Hilbert 11
    Tarski 15
    von Plato 19
    Wu 21
axiome
    d'Archimède 13
    d'Euclide 13
    de Dedekind 14
    de Pasch 11, 12

B

between 12

C

calcul des séquents 129
Cas dégénérés 50, 56, 150
catégorique 15
Chasles, axiome de 69
Church-Rosser 119
cinq segments, axiome des 32
clôture réflexive 117
clôture symétrique 117
clôture transitive 117
cohérence 16
complétude 16, 71, 130
composition 119
confluence 119
confluence forte 119
confluence locale 119
congruence 13
construction d'un segment .. 32, 69
continuité, axiome de 13, 36
corps 10
    euclidien 10
    hilbertien 10
    pythagoricien 10
    réel clos 10
correction 128

D

Dedekind, axiome de 14
diagramme 109, 112
diagramme disjonctif 116
diamant, propriété du 111, 119
différence de Pythagore 57
dimension, axiome de 69
droite des milieux, théorème 77

décidabilité 16, 17
décomposition cylindrique 56

E

égalité 131
Euclide, axiome d' 13, 36
euclidien, corps 10

G

GeoProof 89
Gröbner, bases de 56
géométrie
    affine 19
    archimèdienne 15
    dynamique 89
    incidence 12
    métrique 21
    projective 18

H

hilbertien, corps 10
hypothèse factuelle 122
hypothèse universelle 122

I

induction 134
indépendance 17
inversion 122
isocèle 2

L

lemmes d'élimination 59
logique classique 127
logique inuitionniste 127
Ltac ($\mathcal{L}_{tac}$) 50, 66, 104, 140

M

mesure, axiome de la 13
méthode de superposition 2
méthode de Wu 56
méthode des aires 57
méthode des angles 57
méthode des vecteurs 57

N

Newman, lemme de 136

O

ordre 12

P

parallélogramme, axiome du 69
Pasch, axiome de 11, 12, 33
proportions, axiome de 69
pythagoricien, corps 10

R

ratios de distances 57
Rouse Ball 3
réel clos, corps 10
réflexion 67
réflexive, tactique 67
réfléxivité 119
réécriture 109

S

SAS, Side-Angle-Side 2
semi-confluence 119
système de réécriture abstraite . 111

T

théorème
    centre de gravité 84
    Ceva 79
    Desargues 83
    Gauss 85
    Menelaus 80
    Pappus 82
    Pascal 81
transitivité 38, 119

U

union 123

W

Wu, méthode de 56

Bibliographie

[AA92] Ag-Almouloud. *L'ordinateur, outil d'aide à l'apprentissage de la démonstration et de traitement de données didactiques.* PhD thesis, Université de Rennes, 1992.

[ABPR01] Ahmed Amerkad, Yves Bertot, Loïc Pottier, and Laurence Rideau. Mathematics and proof presentation in Pcoq. In *Workshop Proof Transformation and Presentation and Proof Complexities in connection with IJCAR 2001*, June 2001.

[ABY85] John R. Anderson, C. F. Boyle, and Gregg Yost. The geometry Tutor. In *IJCAI Proceedings*, pages 1–7, 1985.

[ALW04] David Aspinall, Christoph Lüth, and Daniel Winterstein. A framework for interactive proof. Technical report, 2004.

[BC04a] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development, Coq'Art : The Calculus of Inductive Constructions.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[BC04b] Marc Bezem and Thierry Coquand. Newman's lemma – a case study in proof automation and geometric logic. *Current trends in Theoretical Computer Science*, 2 :267–282, 2004.

[BC05] Marc Bezem and Thierry Coquand. Automating coherent logic. In Geoff Sutcliffe and Andrei Voronkov, editors, *12th International Conference, LPAR 2005*, volume 3835 of *Lecture Notes in Computer Science*, pages 246–260. Springer-Verlag, 2005.

[Ber93] Philippe Bernat. Chypre : Un logiciel d'aide au raisonnement. Technical Report 10, IREM, 1993.

[BGP03] Yves Bertot, Frédérique Guilhot, and Loïc Pottier. Visualizing geometrical statements with GeoView. *Electronic Notes in Theoretical Computer Science*, 103 :49–65, September 2003.

[BN98] Franz Baader and Tobias Nipkow. *Term rewriting and all that.* Cambridge University Press, New York, USA, 1998.

[Bou97] Samuel Boutin. Using reflection to build efficient and certified decision procedures. In M. Abadi and T. Ito, editors, *Proceedings of TACS'97*, volume 1281 of *LNCS*. Springer-Verlag, 1997.

[BPB91] Dave Barker-Plummer and Sidney C. Bailin. Proofs and pictures, proving the diamond lemma with the GROVER theorem proving system. AAAI Symposium on Reasoning with Diagrammatic Representations, November 1991.

[BPO+02] Nicolas Balacheff, Sylvie Pesty, Michel Occello, Ricardo Caffera, Carine Webber, and Nicolas Peltier. Baghera, 1999-2002. `http://www-baghera.imag.fr`.

[BT98] Yves Bertot and Laurent Thery. A generic approach to building user interfaces for theorem provers. *The Journal of Symbolic Computation*, 25 :161–194, 1998.

[Buc65] Bruno Buchberger. *An Algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal.* PhD thesis, Math. Inst. University of Innsbruck, Austria, 1965.

[BvOK98] Marc Bezem, Vincent van Oostrom, and Jan Willem Klop. Diagram techniques for confluence. *Information and Computation*, 141(2) :172–204, March 1998.

[CG92] Shang-Ching Chou and Xiao-Shan Gao. A class of geometry statements of constructive type and geometry theorem proving. In *Proceeding of CADE 92*, 1992.

[CGZ94] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. *Machine Proofs in Geometry.* World Scientific, Singapore, 1994.

[CGZ96] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. Automated generation of readable proofs with geometric invariants, theorem proving with full angle. *Journal of Automated Reasoning*, 17 :325–347, 1996.

[Cho85] Shang-Ching Chou. *Proving and discovering geometry theorems using Wu's method.* PhD thesis, The University of Texas, Austin, December 1985.

[Cho88] Shang-Ching Chou. *Mechanical Geometry Theorem Proving.* D. Reidel Publishing Company, 1988.

[Col75] George E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Lecture Notes In Computer Science*, volume 33, pages 134–183. Springer-Verlag, 1975.

[Coq04] Coq development team, The. *The Coq proof assistant reference manual, Version 8.0.* LogiCal Project, 2004.

[CPM90] Thierry Coquand and Christine Paulin-Mohring. Inductively defined types. In P. Martin-Löf and G. Mints, editors, *Proceedings of Colog'88*, volume 417 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990.

[Cre04] Jérôme Creci. Gb : une procédure de décision pour le système coq. In *Journées Francophones des Langages Applicatifs*, 2004.

[DD04a] Christophe Dehlinger and Jean-François Dufourd. Formalizing generalized maps in coq. *Theoretical Computer Science*, 323(1-3) :351–397, 2004.

[DD04b] Christophe Dehlinger and Jean-François Dufourd. Formalizing the trading theorem in coq. *Theoretical Computer Science*, 323(1-3) :399–442, 2004.

[DDS00] Christophe Dehlinger, Jean-François Dufourd, and Pascal Schreck. Higher-order intuitionistic formalization and proofs in Hilbert's elementary geometry. In *Automated Deduction in Geometry*, pages 306–324, 2000.

[Del00] David Delahaye. A Tactic Language for the System Coq. In *Proceedings of Logic for Programming and Automated Reasoning (LPAR), Reunion Island (France)*, volume 1955 of *LNCS/LNAI*, pages 85–95. Springer-Verlag, November 2000.

[Del01] David Delahaye. *Conception de langages pour décrire les preuves et les automatisations dans les outils d'aide à la preuve : une étude dans le cadre du système* Coq. PhD thesis, Université Pierre et Marie Curie (Paris 6), décembre 2001.

[FD03] Fulvia Furinghetti and Paoloa Domingo. To produce conjectures and to prove them within a dynamic geometry environment : a case study. In *Proceeding of Psychology of Mathematics 27th international Conference*, pages 397–404, 2003.

[Gao00] Xiao-Shan Gao. Geometry expert, software package, 2000. `http://www.mmrc.iss.ac.cn/~xgao/gex.html`.

[GL02] Xian-Shan Gao and Qiang Lin. MMP/Geometer - a software package for automated geometry reasoning. In F. Winkler, editor, *Proceedings of ADG 2002*, Lecture Notes in Computer Science, pages 44–46. Springer-Verlag, 2002.

[Gon04] Georges Gonthier. A computer checked proof of the four colour theorem., 2004.

[Gre98] Jacques Gressier. Geometrix, 1988-1998. `http://perso.wanadoo.fr/jgressier/ENGLISH/english.html`.

[Gré03] Benjamin Grégoire. *Compilation des termes de preuves : un (nouveau) mariage entre* Coq *et* Ocaml. PhD thesis, Université Paris 7, December 2003.

[Gui02] Frédérique Guilhot. Proofs with coq of theorems in plane geometry using oriented angles. Technical report, INRIA, January 2002.

[Gui04] Frédérique Guilhot. Formalisation en coq d'un cours de géométrie pour le lycée. In *Journées Francophones des Langages Applicatifs*, Janvier 2004.

[Gui05] Frédérique Guilhot. Formalisation en coq et visualisation d'un cours de géométrie pour le lycée. *Revue des Sciences et Technologies de l'Information, Technique et Science Informatiques, Langages applicatifs*, 24 :1113–1138, 2005. Lavoisier.

[Gup65] Haragauri Narayan Gupta. *Contributions to the axiomatic foundations of geometry.* PhD thesis, University of California, Berkley, 1965.

[Har00] Robin Hartshorne. *Geometry : Euclid and beyond.* Undergraduate texts in mathematics. Springer, 2000.

[Har03] John Harrison. Automatic theorem proving examples, 2003. http://www.cl.cam.ac.uk/users/jrh/atp/index.html.

[Her05] Olivier Hermant. *Méthodes Sémantiques en Déduction modulo.* PhD thesis, Université Paris 7, 2005.

[Hey59] Arend Heyting. Axioms for intuitionistic plane affine geometry. In P. Suppes L. Henkin and A. Tarski, editors, *The axiomatic Method, with special reference to Geometry and Physics*, pages 160–173, Amsterdam, 1959. North-Holland.

[Hil71] David Hilbert. *Les fondements de la géométrie.* Dunod, Paris, Jacques Gabay edition, 1971. Edition critique avec introduction et compléments préparée par Paul Rossier.

[HKPM04] Gérard Huet, Gilles Kahn, and Christine Paulin-Mohring. *The* Coq *Proof Assistant - A tutorial - Version 8.0*, April 2004.

[Hue80] Gérard Huet. Confluent reductions : Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4) :797–821, 1980.

[Jac90] Nicholas Jackiw. The geometer's sketchpad, 1990. http://www.keypress.com/.

[Jam01] Mateja Jamnik. *Mathematical Reasoning with Diagrams : From Intuition to Automation.* CSLI Press, 2001.

[Kah95] Gilles Kahn. Constructive geometry according to Jan von Plato. Coq contribution, 1995. Coq V5.10.

[Kir06] Florent Kirchner. A finite first-order theory of classes. 2006.

[Kk99] Ulrich Kortenkamp. *Foundations of Dynamic Geometry.* PhD thesis, ETH Zürich, 1999.

[Kle52] Stephen Cole Kleene. Permutability of inferences in Gentzen's calculi LK and LJ. *Memoirs of the American Mathematical Society*, 10 :1–26, 1952.

[KRG04] Ulrich Kortenkamp and Jürgen Richter-Gebert. Using automatic theorem proving to improve the usability of geometry software. In *Mathematical User Interface*, 2004.

[LB98] Jean-Marie Laborde and Franck Bellemain. Cabri-geometry II, 1993-1998. `http://www.cabri.net`.

[Lue97] Vanda Luengo. *Cabri-Euclide : Un micromonde de Preuve intégrant la réfutation.* PhD thesis, Université Joseph Fourier, 1997.

[Mah05] Assia Mahboubi. Programming and certifying a CAD algorithm in the `coq` system. In *Mathematics, Algorithms, Proofs*, 2005.

[Mah06] Assia Mahboubi. *Contributions à la certification des calculs sur* $\mathbb{R}$ *: théorie, preuves, programmation.* PhD thesis, Université de Nice Sophia-Antipolis, 2006.

[MF03] Laura Meikle and Jacques Fleuriot. Formalizing Hilbert's Grundlagen in Isabelle/Isar. In *Theorem Proving in Higher Order Logics*, pages 319–334, 2003.

[MF05] Laura Meikle and Jacques Fleuriot. Mechanical theorem proving in computation geometry. In Hoon Hong and Dongming Wang, editors, *Automated Deduction in Geometry 2004*, volume 3763 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, November 2005.

[Mil01] Nathaniel Miller. *A diagrammatic formal system for Euclidean geometry.* PhD thesis, Cornell University, May 2001.

[Nad00] Gopalan Nadathur. Correspondences between classical, intuitionistic and uniform provability. *Theoretical Computer Science*, 232 :273–298, 2000.

[Nar04] Julien Narboux. A decision procedure for geometry in Coq. In Slind Konrad, Bunker Annett, and Gopalakrishnan Ganesh, editors, *Proceedings of TPHOLs'2004*, volume 3223 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004.

[Nar05] Julien Narboux. Toward the use of a proof assistant to teach mathematics. In *Proceedings of the 7th International Conference on Technology in Mathematics Teaching (ICTMT7)*, 2005.

[Nar06a] Julien Narboux. A formalization of diagrammatic proofs in abstract rewriting. 2006.

[Nar06b] Julien Narboux. A graphical user interface for formal proofs in geometry. *the Journal of Automated Reasoning special issue on User Interface for Theorem Proving*, 2006. to appear.

[Nar06c] Julien Narboux. Mechanical theorem proving in Tarski's geometry. Proceedings of Automatic Deduction in Geometry 06, 2006.

[Neg03] Sara Negri. Contraction-free sequent calculi for geometric theories with an application to Barr's theorem. *Archives of Mathematic Logic*, 4(42) :389–401, 2003.

[New42] Maxwell Herman Alexander Newman. On theories with a combinatorial definition of 'equivalence'. *Annals of Mathematics*, 43(2) :223–243, 1942.

[NPW] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. Isabelle/HOL - a proof assistant for Higher-Order Logic.

[Pau06] Lawrence C. Paulson. The Isabelle reference manual, 2006.

[PB01] David Pichardie and Yves Bertot. Formalizing convex hulls algorithms. In *Proc. of 14th International Conference on Theorem Proving in Higher Order Logics (TPHOLs'01)*, volume 2152 of *Lecture Notes in Computer Science*, pages 346–361. Springer-Verlag, 2001.

[Pfe04] Franck Pfenning. Automated theorem proving handouts, April 2004. draft.

[Pie99] Mario Pieri. I principi della geometria di posizione composti in sistema logico deduttivo. *Memorie della Reale Accademia delle Scienze di Torino*, 48 :1–62, 1899.

[Py90] Dominique Py. *Reconnaissance de plan pour l'aide à la démonstration dans un tuteur intelligent de la géométrie.* PhD thesis, Université de Rennes, 1990.

[RGKk99] Jürgen Richter-Gebert and Ulrich Kortenkamp. Die interaktive geometrie software cinderella book and cd-rom. German schooledition of the Cinderella software, 1999. `http://cinderella.de`.

[Rob02] Judit Robu. *Geometry Theorem Proving in the Frame of the Theorema Project.* PhD thesis, Johannes Kepler Universität, Linz, September 2002.

[Sch79] Jacob T. Schwartz. Probabilistic algorithms for verification of polynomial identities. In *Symbolic and algebraic computation*, volume 72 of *Lecture Notes in Computer Science*, pages 200–215, Marseille, 1979. Springer-Verlag.

[SST83] Wolfram Schwabhäuser, Wanda Szmielew, and Alfred Tarski. *Metamathematische Methoden in der Geometrie.* Springer-Verlag, Berlin, 1983.

[Tar51] Alfred Tarski. *A decision method for elementary algebra and geometry.* University of California Press, 1951.

[Tar59] Alfred Tarski. What is elementary geometry ? In P. Suppes L. Henkin and A. Tarski, editors, *The axiomatic Method, with special reference to Geometry and Physics*, pages 16–29, Amsterdam, 1959. North-Holland.

[Tar67] Alfred Tarski. The completeness of elementary algebra and geometry, 1967.

[TG99] Alfred Tarski and Steven Givant. Tarski's system of geometry. *The bulletin of Symbolic Logic*, 5(2), June 1999.

[Thé01] Laurent Théry. A machine-checked implementation of Buchberger's algorithm. *Journal of Automated Reasoning*, 26 :107–137, 2001.

[vP95] Jan von Plato. The axioms of constructive geometry. In *Annals of Pure and Applied Logic*, volume 76, pages 169–200, 1995.

[WAL04] Daniel Winterstein, David Aspinall, and Christoph Lüth. PG/Eclipse : A generic interface for interactive proof. Technical report, 2004.

[Wan00] Dongming Wang. GEOTHER (GEOmetry THeorem provER), 2000.

[WF05] Sean Wilson and Jacques Fleuriot. Combining dynamic geometry, automated geometry theorem proving and diagrammatic proofs. In *ETAPS Satellite Workshop on User Interfaces for Theorem Provers (UITP)*, Edinburgh, 2005.

[Wie] Freek Wiedijk. The de Bruijn factor. Technical report, University of Nijmegen.

[Win04a] Daniel Winterstein. Dr.Doodle : A diagrammatic theorem prover. In *Proceedings of IJCAR 2004*, 2004.

[Win04b] Daniel Winterstein. *Using Diagrammatic Reasoning for Theorem Proving in Continous Domain.* PhD thesis, The University of Edinburgh, 2004.

[Wu78] Wen-Tsün Wu. On the decision problem and the mechanization of theorem proving in elementary geometry. In *Scientia Sinica*, volume 21, pages 157–179. 1978.

[Yev04] Oleksiy Yevdokimov. About a constructivist approach for stimulating students' thinking to produce conjectures and their proving in active learning of geometry. In *Fourth Congress of the European Society for Research in Mathematics Education*, 2004.

Résumé : l'objet de cette thèse est la formalisation et l'automatisation du raisonnement géométrique au sein de l'assistant de preuve Coq.

Dans une première partie, nous réalisons un tour d'horizon des principales axiomatiques de la géométrie puis nous présentons une formalisation des huit premiers chapitres du livre de Schwabäuser, Szmielew et Tarski : *Metamathematische Methoden in der Geometrie.*

Dans la seconde partie, nous présentons l'implantation en Coq d'une procédure de décision pour la géométrie affine plane : la méthode des aires de Chou, Gao et Zhang. Cette méthode produit des preuves courtes et lisibles.

Dans la troisième partie, nous nous intéressons à la conception d'une interface graphique pour la preuve formelle en géométrie : *Geoproof.* GeoProof combine un logiciel de géométrie dynamique avec l'assistant de preuve Coq. Enfin, nous proposons un système formel diagrammatique qui permet de formaliser des raisonnements dans le domaine de la réécriture abstraite. Il est par exemple possible de formaliser dans ce système la preuve diagrammatique du lemme de Newman. La correction et la complétude du système sont prouvées vis-à-vis d'une classe de formules appelée logique cohérente.

Mot-clefs : géométrie, formalisation, automatisation, Coq, axiomatique de Tarski, procédure de décision, méthode des aires, diagrames, logique cohérente, réécriture abstraite, géométrie dynamique.

Abstract: This thesis deals with the formalization and automation of geometric reasoning within the Coq proof assistant.

In the first part, we provide an overview of the main axiom systems for geometry and we present a mechanization of the geometry of Tarski. This consists in the formalization of the first eight chapters of the book of Schwabäuser, Szmielew and Tarski: *Metamathematische Methoden in der Geometrie.*

In the second part, we present our implementation in Coq of a decision procedure for affine plane geometry : the area method of Chou, Gao and Zhang. This method produces short and readable proofs.

In the third part, we explain the design of graphical user interface for formal proof in geometry: *GeoProof.* GeoProof combines a dynamic geometry software with the Coq proof assistant.
Finally, we propose a diagrammatic formal system to perform proofs in the field of abstract term rewriting. For instance, using this system we can formalize the diagrammatic proof of the Newman's lemma. The system is proved correct and complete for a class of formulas called the coherent logic.

Keywords: geometry, formalization, automation, Coq, Tarski's axiomatic, decision procedure, area method, diagrams, coherent logic, abstract rewriting, dynamic geometry.