対話型幾何ソフトウエアと自動証明 : シンデレラとキッズシンディ

阿原, 一志
明治大学

出版情報：MI lecture note series. 61, pp.128-133, 2015-03-06. 九州大学マス・フォア・インダストリ研究所

対話型幾何ソフトウエアと自動証明 ―シンデレラとキッズシンディ

Interactive geometry software and automated theorem proving - Cinderella and KidsCindy

阿原一志（明治大学）
Kazushi AHARA (Meiji university)

自己紹介

- 明治大学　総合数理学部　先端メディアサイエンス学科の　阿原一志です．
- 専門は位相幾何学に関する研究支援ソフトウエア開発です．
- これまで開発した主なソフトウエアはTeruaki（写像類群ソフト），KidsCindy（動的幾何学ソフト）です．
- 機械証明，グレブナー基底のどちらも専門外です．

大まかな内容

- IGS の紹介
- 自動定理証明機能
- シンデレラ･KidsCindyの紹介
- IGSに潜む問題点とその解決
- シンデレラ･KidsCindyにおける自動定理証明機能のアルゴリズム

IGSとは

- 対話型幾何ソフトウエア（Interactive geometry system)の略です．
- 主にマウス操作によって平面（最近では3Dのものも多い）に作図していく機能を持つソフトウエアです．
- 作図手順が記録されているので，作図をした後から頂点などの幾何要素を動かすことができます（実演します）．

IGSの特性

- IGS開発における大切なこと
  - 対話的な高校数学教材を作ることができます．
  - 作図後に図を動かすことにより，ユーザーは作図の中に普遍性を発見することができます．
  - 「コンパスと定規による作図問題」への試行錯誤が容易にできます．
  - 平面幾何の自動定理証明により，非自明な関係性を見つけるのに役立ちます．

主なIGS

- コレテンカンプ氏（シンデレラの作者）が数年前に数えたところによると100種類以上のIGSがあるそうです．
- 初期にはカブリとシンデレラが主流でした．阿原の近辺ではgeogebra，KSEGなどがありました．
- カブリは高価でしたが，日本に一定数のユーザーがいました．その後，シンデレラの日本語版作成のお手伝いをしましたが，日本での反応はいまひとつでした．

KidsCindyを作った理由

- 日本語版シンデレラの普及が進まなかったのは，高校教員に紹介する機会があまり得られなかったためなのですが，当時は「有償ソフトなのがいけないのかもしれない」と考えていました．
- そのことから似た機能を持つフリーソフトを作ってみることにしたのがKidsCindyです．
- そこで障壁になったのが「自動定理証明機能」です．

自動定理証明機能

- 多くのIGSには自動定理証明機能が搭載されています．（実演します．）
- シンデレラはソースコードを公開しておらず，どのような方法で自動定理証明をしているかはわかりません．
- シンデレラの本には「十分いろいろな場合を試してみればその正しさがわかる」というような記述はあったのですが，数学的な裏付けは書かれておらず，どのようなアルゴリズムであるかはわかりません．（実は今でも知りません．）

コンピュータ幾何学

- 「コンピュータ幾何学」という本を出版いたしました．（数学書房，2014年）
- 前半では，IGSにおける計算アルゴリズムについて紹介しています．（後半は別の話題について説明しています．）
- IGSにおける自動定理証明のアルゴリズムについても解説しています．今日はその内容をご紹介します．

Wuの方法1

- 古典的には「Wuの方法」がよく知られています．
- 自由に動かせる点の座標を変数 $u_1, u_2, \dots$ で表し，自由な点から作図によって求められる点の座標を変数$x_1, x_2, \dots$であらわし，その関係式を
$$f_1\,(x_1, u_1, u_2, \dots) = 0,$$
$$f_2\,(x_1, x_2, u_1, u_2, \dots) = 0,$$
$$f_3\,(x_1, x_2, x_3, u_1, u_2, \dots) = 0, \dots$$
- で表します．
- 結論となる命題を
$$g(u_1, u_2, \dots x_1, x_2, \dots) = 0$$
の形に表現します．

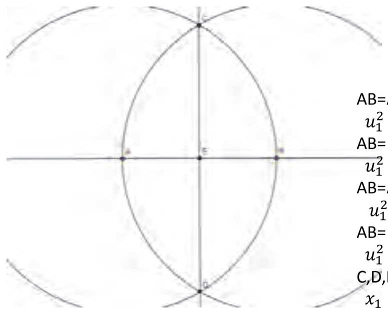
Wuの方法2

- 目標は
$$Q[u_1, u_2, \dots, x_1, x_2, \dots]/(f_1, f_2, \dots)$$
のなかで，$g(u_1, u_2, \dots, x_1, x_2, \dots)$が0と等しいことを証明することです．
- もし，$x_1, x_2, \dots$が
$$x_1 = h_1(u_1, u_2, \dots),$$
$$x_2 = h_2(x_1, u_1, u_2, \dots),$$
$$x_3 = h_3(x_1, x_2, u_1, u_2, \dots), \dots$$
という形で得られるのであれば，これらを順にgに代入すればよいわけですが，多くの作図ではそのような形で得られません．

Wuの方法3

- たとえば，変数がx_1, u_1, u_2に限られているとして，f_1が
$h_1(u_1, u_2)x_1 + h_2(u_1, u_2) = 0$　--(*)
の形で得られていると仮定しましょう．
- $g(x_1, u_1, u_2)$のx_1を 消去するために，
$h_1(u_1, u_2)^s g(x_1, u_1, u_2)$　　---(**)
を計算して(ここで，sは$g(x_1, u_1, u_2)$におけるx_1の次数)，この式(**)に式(*)を代入することによりx_1を消去することができます．（擬剰余）
- すると，
$h_1(u_1, u_2)^s g(x_1, u_1, u_2) \equiv \exists g_0(u_1, u_2)\ \mathrm{mod.}\,(f_1, f_2, \dots)$
とあらわせます．
- 計算の結果，$g_0 = 0$であったとしましょう．すると$h_1 = 0$ または $g = 0$ が想定されます．前者は「図が崩れてしまう場合」，後者は「定理が成立」という形の結論を得ることができます．

Wuの方法（実例）1



$A(0,0), B(u_1,0),$
$C(x_1,x_2), D(x_3,x_4), E(x_5,x_6)$

AB=AC
$u_1^2 - x_1^2 - x_2^2 = 0$
AB=BC
$u_1^2 - (x_1-u_1)^2 - x_2^2 = 0$
AB=AD
$u_1^2 - x_3^2 - x_4^2 = 0$
AB=BD
$u_1^2 - (x_3-u_1)^2 - x_4^2 = 0$
C,D,Eは同一直線上
$x_1\ x_4 + x_2x_5 + x_3x_6 - x_1x_6 - x_2x_3 - x_4x_5 = 0$
A,B,Cは同一直線上
$u_1x_6 = 0$

証明したい性質：AE=EB
$g = x_5^2 + x_6^2 - (x_5-u_1)^2 - x_6^2$

Wuの方法（実例）2

AB=AC
(1) $u_1^2 - x_1^2 - x_2^2 = 0$
AB=BC
(2) $u_1^2 - (x_1-u_1)^2 - x_2^2 = 0$
AB=AD
(3) $u_1^2 - x_3^2 - x_4^2 = 0$
AB=BD
(4) $u_1^2 - (x_3-u_1)^2 - x_4^2 = 0$
C,D,Eは同一直線上
(5) $x_1\ x_4 + x_2x_5 + x_3x_6 - x_1x_6 - x_2x_3 - x_4x_5 = 0$
A,B,Cは同一直線上
(6) $u_1x_6 = 0$

f_1 =(2)-(1)
$= 2x_1u_1 - u_1^2$
f_2 =(1)
$= u_1^2 - x_1^2 - x_2^2$
f_3 =(4)-(3)
$= 2x_3u_1 - u_1^2$
f_4 =(3)
$= u_1^2 - x_3^2 - x_4^2$
$f_5 =$ u_1(5)−$(x_3 - x_1)$(6)
$= u_1(x_1x_4 + x_2x_5 - x_2x_3 - x_4x_5)$
f_6 =(6)
$= u_1x_6$

Wuの方法（実例）3

- $g = x_5^2 + x_6^2 - (x_5-u_1)^2 - x_6^2$
から始めて，$x_6, x_5, \ldots, x_1$の順番に文字を消去していく．その結果
$$4u_1^3(x_2-x_4)\ g(u_1, x_1, \ldots, x_6) \equiv 0$$
$$mod.(f_1, f_2, \ldots, f_6)$$
を示すことができる．
- $u_1 = 0$⇔A=B,
$x_2 - x_4 = 0$⇔C=D
であることを考えれば，A≠B,C ≠Dの仮定のもとにAE=EBが示されることがわかる．

シンデレラはWuの方法を使っていません．

- シンデレラを説明した本で，著者（かつ，シンデレラの作者）であるコルテンカンプは自動定理証明機能について次のように言っています．
- Cinderella does not use symbolic methods to create a formal proof, but a technique called ‘Randomized Theorem Checking.’ First the conjecture “It seems that line *k* always passes through point *K*” is generated. Then the configuration is moved into many different positions and for each of these it is checked whether the conjecture still holds. It may sound ridiculous, but generating enough (!) random(!) examples where the theorem holds is at least as convincing as a computer-generated symbolic proof.

阿原から見たコルテンカンプの思想

- 「点Kは直線k上にある」という命題の証明をするという前提で考えます．
- 自由に動かせる点の座標を変数$x_1, x_2, \ldots$とし，従属的に決まる点の座標を変数を$u_1, u_2, \ldots$で表すことにすると，「点Kが直線k上にある」という関係性は$x_1, x_2, \ldots$の式（一般には根号を含むような式）で表現できます．
- もし$x_1, x_2, \ldots$に「十分に多くの個数のランダムな値を代入してみた結果，式の値が0になる」のならば，その式はそもそも0と等しいと結論できると考えます．

この思想の根本的欠点１

- 「ランダムな値を代入してみた結果，式の値が0になる」ということをコンピュータのソフトウエアで実現しようとすると必ず計算誤差が発生する．「0になる」ことをわれわれが判定するためには「十分に0に近い（閾値の内側である）」という考え方をして判定する必要があり，数学的な厳密さを失ってしまう．

この思想の根本的欠点２

- この思想に基づいて幾何の定理証明を試みるためには，作図によりきめられる「点」や「直線」が自由に動かせる要素の位置にかかわらず大域的に決定できなければならない．しかしそのようなことは不可能である．（依存幾何要素の決定不可能性をコルテンカンプ自身が論文の中で証明している．）
- 「円と直線の交点」「円と円の交点」「２直線の角の2等分線」は大域決定不可能である．

IGSの静的問題

- 「作図手順を記述することにより，図は一意的に定まるか」という問題が静的問題である．答えはNOであるが，作図を行う数学的枠組みを拡張することにより，多くの場合は解決可能である．
- コルテンカンプは射影平面P^2を用いることにより，無限大の問題（たとえば，「２直線の交点」というモジュールにおいて，平行な2直線が入力された場合にどのように出力するか）を解決した．

IGSの動的問題

- 「自由に動かせる要素を連続的に動かした場合，作図全体も連続的に動かせるか」という問題が動的問題である．これも答えはNOである．しかし作図平面を複素射影平面($P^2(C)$)にすることにより，作図要素を代数多様体の上で考えることへと帰着し，その多くは解決することがコルテンカンプにより指摘された．

KidsCindyにおける自動定理証明

- 思想としてはシンデレラに準ずる．
- 自由に動かせる幾何要素について「上下左右の微動」を独立に行い，それぞれの場合について作図を実行してみて，「点Kが直線k上に誤差付で載っているかどうか」を判定する．
- すべての微動について判定がOKであれば，数学の定理として非常に確からしいと判断する．

KidsCindyにおける自動定理証明の有効性

- 平面幾何作図による定理を相当数準備して，KidsCindy が定理判定できるかどうかを確かめたところ（調査した範囲では）１００パーセント正しく判定した．（正１７角形のリッチモンドによる作図も含む．）
- 作図手順の個数から想定される累積誤差を想定した閾値を設定したため，このような結果が得られたと考えられる．（作図手順が１００００～１０００００といった定理がもしあれば，判定に失敗することもありうる．一方で，IGSではこのような巨大作図は想定していない．）

KidsCindyにおける自動定理証明の穴

- KidsCindyでは「正しくないものを正しいと判定してしまう」可能性も残っている．
- たとえば正１７角形の作図においてそれは発生しうる．$\cos(2\pi/17)$の値を十分に近似する作図であれば，真の値でなくともKidsCindyは「正しい」と判定する．ただし，これは可能性を指摘しただけであって，実際に検証してみたわけではない．