|  |
| --- |
| 主専攻実習（定理証明班） |
| 第四回課題レポート |
| 担当：森継 修一 |

|  |
| --- |
| 知識情報システム主専攻 201611502久保川一良  2018年10月29日 |

* 接続環境
  + 自分のローカル環境にReduceをインストールして利用した。
    - 使用したPCのスペックについて： <https://bit.ly/2Cg7hqb>
  + OSやReduceのオプション設定などについては以下の通りである。

username@my\_computer:~ [HH:MM:SS]

$ lsb\_release -a

No LSB modules are available.

Distributor ID: Ubuntu

Description: Ubuntu 16.04.5 LTS

Release: 16.04

Codename: xenial

username@my\_computer:~ [HH:MM:SS]

$ reduce # alias reduce='redcsl -v -w -k 4000 --nogui'

Codemist Standard Lisp revision 4765 for linux-gnu:x86\_64: Sep 19 2018

Created: Wed Sep 19 15:57:15 2018

Reduce (Free CSL version, revision 4765), 19-Sep-18 ...

Memory allocation: 4168 Mbytes

There are 8 processors available

<△ABCにおける外心の証明>

* 入力ファイル

%----------------------------------------------------------------------;

% 三角形ABCにおいて、辺AB, BC, CAそれぞれの中点をD, E, Fとする

% 各中点を通り、それぞれがその辺に垂直な直線を考える

% このとき3つの直線の交点を外心Xとする

%----------------------------------------------------------------------;

% A(0, 0) B(u1, 0) C(u2, u3)

% D(x1, x2) E(x3, x4) F(x5, x6) X(x7, x8)

% ---------------------------------------------------------------------;

% ＜証明＞ -------------------------------------------------------------;

order x8, x7, x6, x5, x4, x3, x2, x1, u3, u2, u1;

factor x8, x7, x6, x5, x4, x3, x2, x1;

% 関数定義読み込み（※ローカル環境へコピーしてきたもの） --------------------------;

in cal\_sys\_relations$

%-------------------------------;

% midpoint(x,a1,a2,b1,b2,c1,c2);

% if x=1 then f:=2\*b1-a1-c1

% else f:=2\*b2-a2-c2;

%-------------------------------;

%----------------------------------------------------------------------;

% 仮定 ----------------------------------------------------------------;

% D は AB の中点 ( A-D-B と考える ) --------------------------------------;

h1:=midpoint(1, 0, 0, x1, x2, u1, 0);

h2:=midpoint(2, 0, 0, x1, x2, u1, 0);

% E は BC の中点 ( B-E-C と考える ) --------------------------------------;

h3:=midpoint(1, u1, 0, x3, x4, u2, u3);

h4:=midpoint(2, u1, 0, x3, x4, u2, u3);

% F は CA の中点 ( C-F-A と考える ) --------------------------------------;

h5:=midpoint(1, u2, u3, x5, x6, 0, 0);

h6:=midpoint(2, u2, u3, x5, x6, 0, 0);

%-------------------------------------------;

% 共線: 直線AB上に点Cがあるときの関数f

% collinear(a1,a2,b1,b2,c1,c2)$

% f:=(a1-b1)\*(b2-c2)-(a2-b2)\*(b1-c1);

%-------------------------------------------;

% 点D, E, Fはそれぞれ辺AB, BC, CA上にある ---------------------------------;

% D は AB上にある( A-D-B と考える ) ---------------------------------------;

h7:=collinear(0, 0, x1, x2, u1, 0);

% E は AB上にある( B-E-C と考える ) ---------------------------------------;

h8:=collinear(u1, 0, x3, x4, u2, u3);

% F は AB上にある( C-F-A と考える ) ---------------------------------------;

h9:=collinear(u2, u3, x5, x6, 0, 0);

%--------------------------------------------;

% 垂線: 直線ABと直線CDが垂直に交わるときの関数f

% vertically(a1,a2,b1,b2,c1,c2,d1,d2)$

% f:=(a1-b1)\*(c1-d1)+(a2-b2)\*(c2-d2);

%--------------------------------------------;

% AB⊥XD --------------------------------------------------------------;

h10:=vertically(0, 0, u1, 0, x7, x8, x1, x2);

% BC⊥XE --------------------------------------------------------------;

h11:=vertically(u1, 0, u2, u3, x7, x8, x3, x4);

% 結論 ----------------------------------------------------------------;

% CA⊥XF --------------------------------------------------------------;

x\_conclusion:=vertically(u2, u3, 0, 0, x7, x8, x5, x6);

showtime;

%----------------------------------------------------------------------;

% Groebner Basis: 結果が1となったら、仮定が誤っている可能性が高い ---------------;

%%% 変数を定義し、lex形式で並べる ------------------------------------------;

torder({x8, x7, x6, x5, x4, x3, x2, x1}, lex)$

%%% 仮定において定義した式からGroebner Basisを求める -------------------------;

gb:=groebner{h1, h2, h3, h4, h5, h6, h7, h8, h9, h10, h11};

%---------------------------------------------------------;

% 「glterms」が出力するのは、グレブナー基底の計算過程で〈ゼロにはならない〉

% と仮定された式のリストである。

%---------------------------------------------------------;

%%% u に関する制約条件 --------------------------------------------------;

glterms;

% => 「u1 = u2」のとき, 点Xは点Fと一致してしまう（x7=x5, x8=x6）が、

% △ABCは成立するので、この条件は無視できる [検証01]

% => 「u2 = 0」のとき, 点Xは点Eと一致してしまう（x7=x3, X8=x4）が、

% △ABCは成立するので、この条件は無視できる [検証02]

% その他、 もし u1, u3 のいずれかが 0 になってしまうと、

% 点が重なり合ってしまったり、共線となってしまい、△ABCは成立しない

%%% gbを法としてgを簡約 --------------------------------------------------;

preduce(x\_conclusion, gb);

% ==> 0 になっていれば、定理は成立 ------------------------------------------;

%-------------------------------------------------------------;

% first(list), second(list)はそれぞれリスト内の1つ目、2つ目の要素を返す

% solve(f, v) は, 関数f の変数をv の方程式としてみて解く

%-------------------------------------------------------------;

% X座標も求めてみる ------------------------------------------------------;

solve(first(gb), x8);

solve(second(gb), x7);

%----------------------------------------------------------------------;

showtime;

;end;

* 出力ファイル

%----------------------------------------------------------------------;

% 三角形ABCにおいて、辺AB, BC, CAそれぞれの中点をD, E, Fとする

% 各中点を通り、それぞれがその辺に垂直な直線を考える

% このとき3つの直線の交点を外心Xとする

%----------------------------------------------------------------------;

% A(0, 0) B(u1, 0) C(u2, u3)

% D(x1, x2) E(x3, x4) F(x5, x6) X(x7, x8)

% ---------------------------------------------------------------------;

% ＜証明＞ -------------------------------------------------------------;

order x8, x7, x6, x5, x4, x3, x2, x1, u3, u2, u1;

factor x8, x7, x6, x5, x4, x3, x2, x1;

% 関数定義読み込み（※ローカル環境へコピーしてきたもの） --------------------------;

in cal\_sys\_relations$

%-------------------------------;

% midpoint(x,a1,a2,b1,b2,c1,c2);

% if x=1 then f:=2\*b1-a1-c1

% else f:=2\*b2-a2-c2;

%-------------------------------;

%----------------------------------------------------------------------;

% 仮定 ----------------------------------------------------------------;

% D は AB の中点 ( A-D-B と考える ) --------------------------------------;

h1:=midpoint(1, 0, 0, x1, x2, u1, 0);

h1 := 2\*x1 - u1

h2:=midpoint(2, 0, 0, x1, x2, u1, 0);

h2 := 2\*x2

% E は BC の中点 ( B-E-C と考える ) --------------------------------------;

h3:=midpoint(1, u1, 0, x3, x4, u2, u3);

h3 := 2\*x3 - u2 - u1

h4:=midpoint(2, u1, 0, x3, x4, u2, u3);

h4 := 2\*x4 - u3

% F は CA の中点 ( C-F-A と考える ) --------------------------------------;

h5:=midpoint(1, u2, u3, x5, x6, 0, 0);

h5 := 2\*x5 - u2

h6:=midpoint(2, u2, u3, x5, x6, 0, 0);

h6 := 2\*x6 - u3

%-------------------------------------------;

% 共線: 直線AB上に点Cがあるときの関数f

% collinear(a1,a2,b1,b2,c1,c2)$

% f:=(a1-b1)\*(b2-c2)-(a2-b2)\*(b1-c1);

%-------------------------------------------;

% 点D, E, Fはそれぞれ辺AB, BC, CA上にある ---------------------------------;

% D は AB上にある( A-D-B と考える ) ---------------------------------------;

h7:=collinear(0, 0, x1, x2, u1, 0);

h7 := - x2\*u1

% E は AB上にある( B-E-C と考える ) ---------------------------------------;

h8:=collinear(u1, 0, x3, x4, u2, u3);

h8 := x4\*( - u2 + u1) + x3\*u3 - u3\*u1

% F は AB上にある( C-F-A と考える ) ---------------------------------------;

h9:=collinear(u2, u3, x5, x6, 0, 0);

h9 := x6\*u2 - x5\*u3

%--------------------------------------------;

% 垂線: 直線ABと直線CDが垂直に交わるときの関数f

% vertically(a1,a2,b1,b2,c1,c2,d1,d2)$

% f:=(a1-b1)\*(c1-d1)+(a2-b2)\*(c2-d2);

%--------------------------------------------;

% AB⊥XD --------------------------------------------------------------;

h10:=vertically(0, 0, u1, 0, x7, x8, x1, x2);

h10 := - x7\*u1 + x1\*u1

% BC⊥XE --------------------------------------------------------------;

h11:=vertically(u1, 0, u2, u3, x7, x8, x3, x4);

h11 := - x8\*u3 + x7\*( - u2 + u1) + x4\*u3 + x3\*(u2 - u1)

% 結論 ----------------------------------------------------------------;

% CA⊥XF --------------------------------------------------------------;

x\_conclusion:=vertically(u2, u3, 0, 0, x7, x8, x5, x6);

x\_conclusion := x8\*u3 + x7\*u2 - x6\*u3 - x5\*u2

showtime;

Time: 20 ms plus GC time: 10 ms

%----------------------------------------------------------------------;

% Groebner Basis: 結果が1となったら、仮定が誤っている可能性が高い ---------------;

%%% 変数を定義し、lex形式で並べる ------------------------------------------;

torder({x8, x7, x6, x5, x4, x3, x2, x1}, lex)$

%%% 仮定において定義した式からGroebner Basisを求める -------------------------;

gb:=groebner{h1, h2, h3, h4, h5, h6, h7, h8, h9, h10, h11};

2 2

gb := {(2\*u3)\*x8 + (u1\*u2 - u2 - u3 ),

2\*x7 - u1,

2\*x6 - u3,

2\*x5 - u2,

2\*x4 - u3,

2\*x3 - (u1 + u2),

x2,

2\*x1 - u1}

%---------------------------------------------------------;

% 「glterms」が出力するのは、グレブナー基底の計算過程で〈ゼロにはならない〉

% と仮定された式のリストである。

%---------------------------------------------------------;

%%% u に関する制約条件 --------------------------------------------------;

glterms;

{u1, - u2 + u1,u2,u3}

% => 「u1 = u2」のとき, 点Xは点Fと一致してしまう（x7=x5, x8=x6）が、

% △ABCは成立するので、この条件は無視できる [検証01]

% => 「u2 = 0」のとき, 点Xは点Eと一致してしまう（x7=x3, X8=x4）が、

% △ABCは成立するので、この条件は無視できる [検証02]

% その他、 もし u1, u3 のいずれかが 0 になってしまうと、

% 点が重なり合ってしまったり、共線となってしまい、△ABCは成立しない

%%% gbを法としてgを簡約 --------------------------------------------------;

preduce(x\_conclusion, gb);

0

% ==> 0 になっていれば、定理は成立 ------------------------------------------;

%-------------------------------------------------------------;

% first(list), second(list)はそれぞれリスト内の1つ目、2つ目の要素を返す

% solve(f, v) は, 関数f の変数をv の方程式としてみて解く

%-------------------------------------------------------------;

% X座標も求めてみる ------------------------------------------------------;

solve(first(gb), x8);

2 2

u3 + u2 - u2\*u1

{x8=-------------------}

2\*u3

solve(second(gb), x7);

u1

{x7=----}

2

%----------------------------------------------------------------------;

showtime;

Time: 30 ms

;

end;

# u に関する制約条件　についての検証１

* 入力ファイル

%----------------------------------------------------------------------;

% 三角形ABCにおいて、辺AB, BC, CAそれぞれの中点をD, E, Fとする

% 各中点を通り、それぞれがその辺に垂直な直線を考える

% このとき3つの直線の交点を外心Xとする

%%% u に関する制約条件の検証 ---------------------------------------------;

% glterms;

% => 「u1 = u2」のとき, 点Xは点Fと一致してしまう（x7=x5, x8=x6）が、

% △ABCは成立するので、この条件は無視できる[検証１]

%----------------------------------------------------------------------;

% A(0, 0) B(u1, 0) C(u2, u3)

% D(x1, x2) E(x3, x4) F(x5, x6) X(x7, x8)

% ---------------------------------------------------------------------;

% ＜検証１＞ -------------------------------------------------------------;

order x8, x7, x6, x5, x4, x3, x2, x1, u3, u2, u1;

factor x8, x7, x6, x5, x4, x3, x2, x1;

% 検証条件

u1:=u2;

% 関数定義読み込み（※ローカル環境へコピーしてきたもの） --------------------------;

in cal\_sys\_relations$

%----------------------------------------------------------------------;

% 仮定 ----------------------------------------------------------------;

% D は AB の中点 ( A-D-B と考える ) --------------------------------------;

h1:=midpoint(1, 0, 0, x1, x2, u1, 0);

h2:=midpoint(2, 0, 0, x1, x2, u1, 0);

% E は BC の中点 ( B-E-C と考える ) --------------------------------------;

h3:=midpoint(1, u1, 0, x3, x4, u2, u3);

h4:=midpoint(2, u1, 0, x3, x4, u2, u3);

% F は CA の中点 ( C-F-A と考える ) --------------------------------------;

h5:=midpoint(1, u2, u3, x5, x6, 0, 0);

h6:=midpoint(2, u2, u3, x5, x6, 0, 0);

% 点D, E, Fはそれぞれ辺AB, BC, CA上にある ---------------------------------;

% D は AB上にある( A-D-B と考える ) ---------------------------------------;

h7:=collinear(0, 0, x1, x2, u1, 0);

% E は AB上にある( B-E-C と考える ) ---------------------------------------;

h8:=collinear(u1, 0, x3, x4, u2, u3);

% F は AB上にある( C-F-A と考える ) ---------------------------------------;

h9:=collinear(u2, u3, x5, x6, 0, 0);

% AB⊥XD --------------------------------------------------------------;

h10:=vertically(0, 0, u1, 0, x7, x8, x1, x2);

% BC⊥XE --------------------------------------------------------------;

h11:=vertically(u1, 0, u2, u3, x7, x8, x3, x4);

% 結論 ----------------------------------------------------------------;

% CA⊥XF --------------------------------------------------------------;

x\_conclusion:=vertically(u2, u3, 0, 0, x7, x8, x5, x6);

showtime;

%----------------------------------------------------------------------;

% Groebner Basis: 結果が1となったら、仮定が誤っている可能性が高い ---------------;

%%% 変数を定義し、lex形式で並べる ------------------------------------------;

torder({x8, x7, x6, x5, x4, x3, x2, x1}, lex)$

%%% 仮定において定義した式からGroebner Basisを求める -------------------------;

gb:=groebner{h1, h2, h3, h4, h5, h6, h7, h8, h9, h10, h11};

%%% u に関する制約条件 --------------------------------------------------;

glterms;

% => u2=0 もしくは u3=0 のとき、△ABCは成立しない

%%% gbを法としてgを簡約 --------------------------------------------------;

preduce(x\_conclusion, gb);

% ==> 0 になっていれば、定理は成立 ------------------------------------------;

% X座標も求めてみる ------------------------------------------------------;

solve(first(gb), x8);

solve(second(gb), x7);

%----------------------------------------------------------------------;

showtime;

;end;

* 出力ファイル

%----------------------------------------------------------------------;

% 三角形ABCにおいて、辺AB, BC, CAそれぞれの中点をD, E, Fとする

% 各中点を通り、それぞれがその辺に垂直な直線を考える

% このとき3つの直線の交点を外心Xとする

%%% u に関する制約条件の検証 ---------------------------------------------;

% glterms;

% => 「u1 = u2」のとき, 点Xは点Fと一致してしまう（x7=x5, x8=x6）が、

% △ABCは成立するので、この条件は無視できる[検証１]

%----------------------------------------------------------------------;

% A(0, 0) B(u1, 0) C(u2, u3)

% D(x1, x2) E(x3, x4) F(x5, x6) X(x7, x8)

% ---------------------------------------------------------------------;

% ＜検証１＞ -------------------------------------------------------------;

order x8, x7, x6, x5, x4, x3, x2, x1, u3, u2, u1;

factor x8, x7, x6, x5, x4, x3, x2, x1;

% 検証条件

u1:=u2;

u1 := u2

% 関数定義読み込み（※ローカル環境へコピーしてきたもの） --------------------------;

in cal\_sys\_relations$

%----------------------------------------------------------------------;

% 仮定 ----------------------------------------------------------------;

% D は AB の中点 ( A-D-B と考える ) --------------------------------------;

h1:=midpoint(1, 0, 0, x1, x2, u1, 0);

h1 := 2\*x1 - u2

h2:=midpoint(2, 0, 0, x1, x2, u1, 0);

h2 := 2\*x2

% E は BC の中点 ( B-E-C と考える ) --------------------------------------;

h3:=midpoint(1, u1, 0, x3, x4, u2, u3);

h3 := 2\*x3 - 2\*u2

h4:=midpoint(2, u1, 0, x3, x4, u2, u3);

h4 := 2\*x4 - u3

% F は CA の中点 ( C-F-A と考える ) --------------------------------------;

h5:=midpoint(1, u2, u3, x5, x6, 0, 0);

h5 := 2\*x5 - u2

h6:=midpoint(2, u2, u3, x5, x6, 0, 0);

h6 := 2\*x6 - u3

% 点D, E, Fはそれぞれ辺AB, BC, CA上にある ---------------------------------;

% D は AB上にある( A-D-B と考える ) ---------------------------------------;

h7:=collinear(0, 0, x1, x2, u1, 0);

h7 := - x2\*u2

% E は AB上にある( B-E-C と考える ) ---------------------------------------;

h8:=collinear(u1, 0, x3, x4, u2, u3);

h8 := x3\*u3 - u3\*u2

% F は AB上にある( C-F-A と考える ) ---------------------------------------;

h9:=collinear(u2, u3, x5, x6, 0, 0);

h9 := x6\*u2 - x5\*u3

% AB⊥XD --------------------------------------------------------------;

h10:=vertically(0, 0, u1, 0, x7, x8, x1, x2);

h10 := - x7\*u2 + x1\*u2

% BC⊥XE --------------------------------------------------------------;

h11:=vertically(u1, 0, u2, u3, x7, x8, x3, x4);

h11 := - x8\*u3 + x4\*u3

% 結論 ----------------------------------------------------------------;

% CA⊥XF --------------------------------------------------------------;

x\_conclusion:=vertically(u2, u3, 0, 0, x7, x8, x5, x6);

x\_conclusion := x8\*u3 + x7\*u2 - x6\*u3 - x5\*u2

showtime;

Time: 20 ms plus GC time: 9 ms

%----------------------------------------------------------------------;

% Groebner Basis: 結果が1となったら、仮定が誤っている可能性が高い ---------------;

%%% 変数を定義し、lex形式で並べる ------------------------------------------;

torder({x8, x7, x6, x5, x4, x3, x2, x1}, lex)$

%%% 仮定において定義した式からGroebner Basisを求める -------------------------;

gb:=groebner{h1, h2, h3, h4, h5, h6, h7, h8, h9, h10, h11};

gb := {2\*x8 - u3,

2\*x7 - u2,

2\*x6 - u3,

2\*x5 - u2,

2\*x4 - u3,

x3 - u2,

x2,

2\*x1 - u2}

%%% u に関する制約条件 --------------------------------------------------;

glterms;

{u2,u3}

% => u2=0 もしくは u3=0 のとき、△ABCは成立しない

%%% gbを法としてgを簡約 --------------------------------------------------;

preduce(x\_conclusion, gb);

0

% ==> 0 になっていれば、定理は成立 ------------------------------------------;

% X座標も求めてみる ------------------------------------------------------;

solve(first(gb), x8);

u3

{x8=----}

2

solve(second(gb), x7);

u2

{x7=----}

2

%----------------------------------------------------------------------;

showtime;

Time: 20 ms

;

end;

# u に関する制約条件　についての検証２

* 入力ファイル

%----------------------------------------------------------------------;

% 三角形ABCにおいて、辺AB, BC, CAそれぞれの中点をD, E, Fとする

% 各中点を通り、それぞれがその辺に垂直な直線を考える

% このとき3つの直線の交点を外心Xとする

%%% u に関する制約条件の検証 ---------------------------------------------;

% glterms;

% => 「u2 = 0」のとき, 点Xは点Eと一致してしまう（x7=x3, X8=x4）が、

% △ABCは成立するので、この条件は無視できる[検証２]

%----------------------------------------------------------------------;

% A(0, 0) B(u1, 0) C(u2, u3)

% D(x1, x2) E(x3, x4) F(x5, x6) X(x7, x8)

% ---------------------------------------------------------------------;

% ＜検証2＞ -------------------------------------------------------------;

order x8, x7, x6, x5, x4, x3, x2, x1, u3, u2, u1;

factor x8, x7, x6, x5, x4, x3, x2, x1;

% 検証条件

u2:=0;

% 関数定義読み込み（※ローカル環境へコピーしてきたもの） --------------------------;

in cal\_sys\_relations$

%----------------------------------------------------------------------;

% 仮定 ----------------------------------------------------------------;

% D は AB の中点 ( A-D-B と考える ) --------------------------------------;

h1:=midpoint(1, 0, 0, x1, x2, u1, 0);

h2:=midpoint(2, 0, 0, x1, x2, u1, 0);

% E は BC の中点 ( B-E-C と考える ) --------------------------------------;

h3:=midpoint(1, u1, 0, x3, x4, u2, u3);

h4:=midpoint(2, u1, 0, x3, x4, u2, u3);

% F は CA の中点 ( C-F-A と考える ) --------------------------------------;

h5:=midpoint(1, u2, u3, x5, x6, 0, 0);

h6:=midpoint(2, u2, u3, x5, x6, 0, 0);

% 点D, E, Fはそれぞれ辺AB, BC, CA上にある ---------------------------------;

% D は AB上にある( A-D-B と考える ) ---------------------------------------;

h7:=collinear(0, 0, x1, x2, u1, 0);

% E は AB上にある( B-E-C と考える ) ---------------------------------------;

h8:=collinear(u1, 0, x3, x4, u2, u3);

% F は AB上にある( C-F-A と考える ) ---------------------------------------;

h9:=collinear(u2, u3, x5, x6, 0, 0);

% AB⊥XD --------------------------------------------------------------;

h10:=vertically(0, 0, u1, 0, x7, x8, x1, x2);

% BC⊥XE --------------------------------------------------------------;

h11:=vertically(u1, 0, u2, u3, x7, x8, x3, x4);

% 結論 ----------------------------------------------------------------;

% CA⊥XF --------------------------------------------------------------;

x\_conclusion:=vertically(u2, u3, 0, 0, x7, x8, x5, x6);

showtime;

%----------------------------------------------------------------------;

% Groebner Basis: 結果が1となったら、仮定が誤っている可能性が高い ---------------;

%%% 変数を定義し、lex形式で並べる ------------------------------------------;

torder({x8, x7, x6, x5, x4, x3, x2, x1}, lex)$

%%% 仮定において定義した式からGroebner Basisを求める -------------------------;

gb:=groebner{h1, h2, h3, h4, h5, h6, h7, h8, h9, h10, h11};

%%% u に関する制約条件 --------------------------------------------------;

glterms;

% => u1=0 もしくは u3=0 のとき、△ABCは成立しない

%%% gbを法としてgを簡約 --------------------------------------------------;

preduce(x\_conclusion, gb);

% ==> 0 になっていれば、定理は成立 ------------------------------------------;

% X座標も求めてみる ------------------------------------------------------;

solve(first(gb), x8);

solve(second(gb), x7);

%----------------------------------------------------------------------;

showtime;

;end;

* 出力ファイル

%----------------------------------------------------------------------;

% 三角形ABCにおいて、辺AB, BC, CAそれぞれの中点をD, E, Fとする

% 各中点を通り、それぞれがその辺に垂直な直線を考える

% このとき3つの直線の交点を外心Xとする

%%% u に関する制約条件の検証 ---------------------------------------------;

% glterms;

% => 「u2 = 0」のとき, 点Xは点Eと一致してしまう（x7=x3, X8=x4）が、

% △ABCは成立するので、この条件は無視できる[検証２]

%----------------------------------------------------------------------;

% A(0, 0) B(u1, 0) C(u2, u3)

% D(x1, x2) E(x3, x4) F(x5, x6) X(x7, x8)

% ---------------------------------------------------------------------;

% ＜検証2＞ -------------------------------------------------------------;

order x8, x7, x6, x5, x4, x3, x2, x1, u3, u2, u1;

factor x8, x7, x6, x5, x4, x3, x2, x1;

% 検証条件

u2:=0;

u2 := 0

% 関数定義読み込み（※ローカル環境へコピーしてきたもの） --------------------------;

in cal\_sys\_relations$

%----------------------------------------------------------------------;

% 仮定 ----------------------------------------------------------------;

% D は AB の中点 ( A-D-B と考える ) --------------------------------------;

h1:=midpoint(1, 0, 0, x1, x2, u1, 0);

h1 := 2\*x1 - u1

h2:=midpoint(2, 0, 0, x1, x2, u1, 0);

h2 := 2\*x2

% E は BC の中点 ( B-E-C と考える ) --------------------------------------;

h3:=midpoint(1, u1, 0, x3, x4, u2, u3);

h3 := 2\*x3 - u1

h4:=midpoint(2, u1, 0, x3, x4, u2, u3);

h4 := 2\*x4 - u3

% F は CA の中点 ( C-F-A と考える ) --------------------------------------;

h5:=midpoint(1, u2, u3, x5, x6, 0, 0);

h5 := 2\*x5

h6:=midpoint(2, u2, u3, x5, x6, 0, 0);

h6 := 2\*x6 - u3

% 点D, E, Fはそれぞれ辺AB, BC, CA上にある ---------------------------------;

% D は AB上にある( A-D-B と考える ) ---------------------------------------;

h7:=collinear(0, 0, x1, x2, u1, 0);

h7 := - x2\*u1

% E は AB上にある( B-E-C と考える ) ---------------------------------------;

h8:=collinear(u1, 0, x3, x4, u2, u3);

h8 := x4\*u1 + x3\*u3 - u3\*u1

% F は AB上にある( C-F-A と考える ) ---------------------------------------;

h9:=collinear(u2, u3, x5, x6, 0, 0);

h9 := - x5\*u3

% AB⊥XD --------------------------------------------------------------;

h10:=vertically(0, 0, u1, 0, x7, x8, x1, x2);

h10 := - x7\*u1 + x1\*u1

% BC⊥XE --------------------------------------------------------------;

h11:=vertically(u1, 0, u2, u3, x7, x8, x3, x4);

h11 := - x8\*u3 + x7\*u1 + x4\*u3 - x3\*u1

% 結論 ----------------------------------------------------------------;

% CA⊥XF --------------------------------------------------------------;

x\_conclusion:=vertically(u2, u3, 0, 0, x7, x8, x5, x6);

x\_conclusion := x8\*u3 - x6\*u3

showtime;

Time: 0 ms plus GC time: 30 ms

%----------------------------------------------------------------------;

% Groebner Basis: 結果が1となったら、仮定が誤っている可能性が高い ---------------;

%%% 変数を定義し、lex形式で並べる ------------------------------------------;

torder({x8, x7, x6, x5, x4, x3, x2, x1}, lex)$

%%% 仮定において定義した式からGroebner Basisを求める -------------------------;

gb:=groebner{h1, h2, h3, h4, h5, h6, h7, h8, h9, h10, h11};

gb := {2\*x8 - u3,

2\*x7 - u1,

2\*x6 - u3,

x5,

2\*x4 - u3,

2\*x3 - u1,

x2,

2\*x1 - u1}

%%% u に関する制約条件 --------------------------------------------------;

glterms;

{u1,u3}

% => u1=0 もしくは u3=0 のとき、△ABCは成立しない

%%% gbを法としてgを簡約 --------------------------------------------------;

preduce(x\_conclusion, gb);

0

% ==> 0 になっていれば、定理は成立 ------------------------------------------;

% X座標も求めてみる ------------------------------------------------------;

solve(first(gb), x8);

u3

{x8=----}

2

solve(second(gb), x7);

u1

{x7=----}

2

%----------------------------------------------------------------------;

showtime;

Time: 20 ms

;

end;