主専攻実習（定理証明班）

第五回課題レポート

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 11 月 12 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
```

◇入力ファイル

```
%----------------------------------------------------------------------;
% 〈三角形の内心の証明〉
% A(0, 0) B(u1, 0) C(u2, u3) の三点による三角形 ABC を考える
% 内心を作図から求める手順に沿って証明してみる
%----------------------------------------------------------------------;

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

order x7, x6, x5,x4, x3, x2, x1, u8, u7, u6, u5, u4, u3, u2, u1;
factor 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$


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


% --------------------------------------------------------------------;
% 直線 CA 上に、AB=AD となる点 D をとる ----------------------------------------;
% BD の中点を E とすると、△ABD は二等辺三角形だから、AE⊥BD となる -------------------;
% D は CA 上にあるのだから、共線となっている ( C-D-A と考える ) -----------------------;
% また、E は BD 上にあり、共線となっている ( B-E-D と考える ) -------------------------;
% D(x1, u4) E(x2, u5)

h1:=requal(0, 0, u1, 0, 0, 0, x1, u4);
h2:=midpoint(1, u1, 0, x2, u5, x1, u4);
h3:=midpoint(2, u1, 0, x2, u5, x1, u4);
h4:=vertically(0, 0, x2, u5, u1, 0, x1, u4);
h5:=collinear(u2, u3, x1, u4, 0, 0);
h6:=collinear(u1, 0, x2, u5, x1, u4);



% --------------------------------------------------------------------;
```

```
% 直線 BC 上に、BA=BF となる点 F をとる -----------------------------------------;
% FA の中点を G とすると、△BFA は二等辺三角形だから、BG⊥FA となる --------------------;
% F は BC 上にあるのだから、共線となっている ( B-F-C と考える ) -----------------------;
% また、G は FA 上にあり、共線となっている ( F-G-A と考える ) ------------------------;
% F(x3, u6) G(x4, u7)

h7:=requal(u1, 0, 0, 0, u1, 0, x3, u6);
h8:=midpoint(1, x3, u6, x4, u7, 0, 0);
h9:=midpoint(2, x3, u6, x4, u7, 0, 0);
h10:=vertically(u1, 0, x4, u7, x3, u6, 0, 0);
h11:=collinear(u1, 0, x3, u6, u2, u3);
h12:=collinear(x3, u6, x4, u7, 0, 0);




% 点 I は AE 上にあり、かつ BG 上にある (AE と BG の交点である) ------------------------;
% つまり、共線となっている ( A-I-E , B-I-G と考える)
% I(x6, u8)

h13:=collinear(0, 0, x6, u8, x3, u6);
h14:=collinear(u1, 0, x6, u8, x4, u7);



%-------------------------------------------------------------------;
% 結論 --------------------------------------------------------------;
% 二角の二等分線が交わる点 I は CH 上にあり、共線となっている ( C-I-H と考える ) --------;

i_conclusion:=collinear(u2, u3, x6, u8, x5, 0);

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({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, h12, h13, h14};

%-----------------------------------------------------;
% 「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;


%%% gb を法として g を簡約 -------------------------------------------------;
preduce(i_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

```
%----------------------------------------------------------------------;
% 〈三角形の内心の証明〉
% A(0, 0) B(u1, 0) C(u2, u3) の三点による三角形 ABC を考える
% 内心を作図から求める手順に沿って証明してみる
%----------------------------------------------------------------------;

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

order x7, x6, x5,x4, x3, x2, x1, u8, u7, u6, u5, u4, u3, u2, u1;


factor 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$




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


% --------------------------------------------------------------------;
% 直線 CA 上に、AB=AÐ となる点 Ð をとる ----------------------------------------;
% BÐ の中点を E とすると、△ABÐ は二等辺三角形だから、AE⊥BÐ となる --------------------;
% Ð は CA 上にあるのだから、共線となっている（ C-Ð-A と考える ）-----------------------;
% また、E は BÐ 上にあり、共線となっている（ B-E-Ð と考える ）------------------------;
% Ð(x1, u4) E(x2, u5)

h1:=requal(0, 0, u1, 0, 0, 0, x1, u4);


           2     2     2
h1 :=  - x1  - u4  + u1

h2:=midpoint(1, u1, 0, x2, u5, x1, u4);


h2 := 2*x2 - x1 - u1

h3:=midpoint(2, u1, 0, x2, u5, x1, u4);


h3 := 2*u5 - u4

h4:=vertically(0, 0, x2, u5, u1, 0, x1, u4);


h4 := x2*x1 - x2*u1 + u5*u4

h5:=collinear(u2, u3, x1, u4, 0, 0);


h5 :=  - x1*u3 + u4*u2

h6:=collinear(u1, 0, x2, u5, x1, u4);


h6 := x2*u4 - x1*u5 + u5*u1 - u4*u1




% --------------------------------------------------------------------;
% 直線 BC 上に、BA=BF となる点 F をとる ----------------------------------------;
% FA の中点を G とすると、△BFA は二等辺三角形だから、BG⊥FA となる --------------------;
% F は BC 上にあるのだから、共線となっている（ B-F-C と考える ）-----------------------;
% また、G は FA 上にあり、共線となっている（ F-G-A と考える ）------------------------;
% F(x3, u6) G(x4, u7)

h7:=requal(u1, 0, 0, 0, u1, 0, x3, u6);
```

```
          2                2
h7 :=  - x3  + 2*x3*u1 - u6

h8:=midpoint(1, x3, u6, x4, u7, 0, 0);


h8 := 2*x4 - x3

h9:=midpoint(2, x3, u6, x4, u7, 0, 0);


h9 := 2*u7 - u6

h10:=vertically(u1, 0, x4, u7, x3, u6, 0, 0);


h10 :=  - x4*x3 + x3*u1 - u7*u6

h11:=collinear(u1, 0, x3, u6, u2, u3);


h11 := x3*u3 - u6*u2 + u6*u1 - u3*u1

h12:=collinear(x3, u6, x4, u7, 0, 0);


h12 :=  - x4*u6 + x3*u7




% 点 I は AE 上にあり、かつ BG 上にある（AE と BG の交点である）-------------------------;
% つまり、共線となっている（ A-I-E , B-I-G と考える）
% I(x6, u8)

h13:=collinear(0, 0, x6, u8, x3, u6);


h13 := x6*u6 - x3*u8

h14:=collinear(u1, 0, x6, u8, x4, u7);


h14 := x6*u7 - x4*u8 + u8*u1 - u7*u1




%--------------------------------------------------------------------;
% 結論 ---------------------------------------------------------------;
% 二角の二等分線が交わる点 I は CH 上にあり、共線となっている（ C-I-H と考える ）--------;

i_conclusion:=collinear(u2, u3, x6, u8, x5, 0);


i_conclusion :=  - x6*u3 + x5*( - u8 + u3) + u8*u2


showtime;


Time: 0 ms  plus GC time: 20 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({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, h12, h13, h14};
```

```
gb := {1}


%--------------------------------------------------------;
% 「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;


{ - 2*u5 + u4,

 u3,

 u4,

  - 2*u7 + u6,

 u6,

 u7,

 u2,

  - u6*u2 + u6*u1 + u3*u1}



%%% gb を法として g を簡約 ------------------------------------------------;
preduce(i_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);
%solve(second(gb), x7);
%----------------------------------------------------------------------;

showtime;


Time: 0 ms  plus GC time: 9 ms


;

end;
```

◇入力ファイル

```
% [定理 １３] ----------------------------------------------------------;
% 〈メネラウス(Menelaus)の定理の証明〉
% △ABC の 3 辺 BC, CA, AB またはその延長が頂点を通らない１直線と交わる点をそれぞれ
% P, Q, R とする。 ※１直線 P-Q-R が x 軸と一致するように座標を取った
% A(u1, u2) B(u3, u4) C(u5, u6)
% P(x1, 0) Q(x2, 0) R(x3, 0)
%----------------------------------------------------------------------;

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

```
order x3, x2, x1, u6, u5, u4, u3, u2, u1;
factor 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$


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

% P, Q, R は共線となっている ----------------------------------------------;

h1:=collinear(x1, 0, x2, 0, x3, 0);


% 各頂点から PQR に下ろした垂線の足をそれぞれ Ð, E, F とする
% Ð(u1, 0) E(u3, 0) F(u5, 0)
% つまり、PQ⊥BE, QR⊥CF, RP⊥AÐ となる

h2:=vertically(x1, 0, x2, 0, u3, u4, u3, 0);
h3:=vertically(x2, 0, x3, 0, u5, u6, u5, 0);
h4:=vertically(x3, 0, x1, 0, u1, u2, u1, 0);


% A-B-P, B-Q-C, C-R-A はそれぞれ共線となっている -----------------------------;

h5:=collinear(u1, u2, u3, u4, x1, 0);
h6:=collinear(u3, u4, x2, 0, u5, u6);
h7:=collinear(u5, u6, x3, 0, u1, u2);


%----------------------------------------------------------------------;
% 結論 ----------------------------------------------------------------;
% (BP*CQ*AR)^2 = (PC*QA*RB)^2 が導ければよい
% ※各辺の長さは負の数を取りえないので，負の場合を考える必要はない

bp2:=(x1-u3)^2+(0-u4)^2;
cq2:=(x2-u5)^2+(0-u6)^2;
ar2:=(x3-u1)^2+(0-u2)^2;

pc2:=(u5-x1)^2+(u6-0)^2;
qa2:=(u1-x2)^2+(u2-0)^2;
rb2:=(u3-x3)^2+(u4-0)^2;

conclusion:=(bp2*cq2*ar2)-(pc2*qa2*rb2);

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({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};

%--------------------------------------------------------;
% 「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;


%%% gb を法として g を簡約 --------------------------------------------------;
preduce(conclusion, gb);

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

showtime;

;end;
```

◇出力ファイル

```
% [定理 １３] ---------------------------------------------------------;
% 〈メネラウス(Menelaus)の定理の証明〉
% △ABC の 3 辺 BC, CA, AB またはその延長が頂点を通らない１直線と交わる点をそれぞれ
% P, Q, R とする。 ※１直線 P-Q-R が x 軸と一致するように座標を取った
% A(u1, u2) B(u3, u4) C(u5, u6)
% P(x1, 0) Q(x2, 0) R(x3, 0)
%----------------------------------------------------------------------;

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

order x3, x2, x1, u6, u5, u4, u3, u2, u1;


factor 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$




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

% P, Q, R は共線となっている -----------------------------------------------;

h1:=collinear(x1, 0, x2, 0, x3, 0);


h1 := 0



% 各頂点から PQR に下ろした垂線の足をそれぞれ Ð, E, F とする
% Ð(u1, 0) E(u3, 0) F(u5, 0)
% つまり、PQ⊥BE, QR⊥CF, RP⊥AÐ となる

h2:=vertically(x1, 0, x2, 0, u3, u4, u3, 0);


h2 := 0

h3:=vertically(x2, 0, x3, 0, u5, u6, u5, 0);


h3 := 0

h4:=vertically(x3, 0, x1, 0, u1, u2, u1, 0);


h4 := 0



% A-B-P, B-Q-C, C-R-A はそれぞれ共線となっている -----------------------------;

h5:=collinear(u1, u2, u3, u4, x1, 0);


h5 := x1*( - u4 + u2) + u4*u1 - u3*u2

h6:=collinear(u3, u4, x2, 0, u5, u6);


h6 := x2*(u6 - u4) - u6*u3 + u5*u4

h7:=collinear(u5, u6, x3, 0, u1, u2);
```

```
h7 := x3*( - u6 + u2) + u6*u1 - u5*u2



%----------------------------------------------------------------------;
% 結論 -----------------------------------------------------------------;
% (BP*CQ*AR)^2 = (PC*QA*RB)^2 が導ければよい
% ※各辺の長さは負の数を取りえないので, 負の場合を考える必要はない

bp2:=(x1-u3)^2+(0-u4)^2;


         2               2     2
bp2 := x1  - 2*x1*u3 + u4  + u3

cq2:=(x2-u5)^2+(0-u6)^2;


         2               2     2
cq2 := x2  - 2*x2*u5 + u6  + u5

ar2:=(x3-u1)^2+(0-u2)^2;


         2               2     2
ar2 := x3  - 2*x3*u1 + u2  + u1


pc2:=(u5-x1)^2+(u6-0)^2;


         2               2     2
pc2 := x1  - 2*x1*u5 + u6  + u5

qa2:=(u1-x2)^2+(u2-0)^2;


         2               2     2
qa2 := x2  - 2*x2*u1 + u2  + u1

rb2:=(u3-x3)^2+(u4-0)^2;


         2               2     2
rb2 := x3  - 2*x3*u3 + u4  + u3

conclusion:=(bp2*cq2*ar2)-(pc2*qa2*rb2);


                2   2                      2   2       2     2     2     2
conclusion := x3 *x2 *x1*(2*u5 - 2*u3) + x3 *x2 *( - u6  - u5  + u4  + u3 )

                   2      2                      2
               + x3 *x2*x1 *( - 2*u5 + 2*u1) + x3 *x2*x1*(4*u5*u3 - 4*u5*u1)

                   2         2          2             2          2
               + x3 *x2*(2*u6 *u1 + 2*u5 *u1 - 2*u5*u4  - 2*u5*u3 )

                   2   2    2     2     2     2
               + x3 *x1 *(u6  + u5  - u2  - u1 )

                   2            2          2              2          2      2
               + x3 *x1*( - 2*u6 *u3 - 2*u5 *u3 + 2*u5*u2  + 2*u5*u1 ) + x3 *(

                   2   2     2   2     2   2     2   2     2   2     2   2
                 u6 *u4  + u6 *u3  - u6 *u2  - u6 *u1  + u5 *u4  + u5 *u3

                      2   2     2   2          2   2
                  - u5 *u2  - u5 *u1 ) + x3*x2 *x1 *(2*u3 - 2*u1)

                      2
               + x3*x2 *x1*( - 4*u5*u3 + 4*u3*u1)

                      2      2          2          2          2
               + x3*x2 *(2*u6 *u3 + 2*u5 *u3 - 2*u4 *u1 - 2*u3 *u1)
```

```
                         2
               + x3*x2*x1 *(4*u5*u1 - 4*u3*u1) + x3*x2

                       2             2                2             2
              *( - 4*u6 *u3*u1 - 4*u5 *u3*u1 + 4*u5*u4 *u1 + 4*u5*u3 *u1)

                      2         2          2             2          2
               + x3*x1 *( - 2*u6 *u1 - 2*u5 *u1 + 2*u3*u2  + 2*u3*u1 )

                            2             2                   2             2
               + x3*x1*(4*u6 *u3*u1 + 4*u5 *u3*u1 - 4*u5*u3*u2  - 4*u5*u3*u1 ) +

                          2   2          2   2          2      2       2      2
               x3*( - 2*u6 *u4 *u1 - 2*u6 *u3 *u1 + 2*u6 *u3*u2  + 2*u6 *u3*u1

                        2   2          2   2          2      2       2      2
                  - 2*u5 *u4 *u1 - 2*u5 *u3 *u1 + 2*u5 *u3*u2  + 2*u5 *u3*u1 )

                   2   2       2     2     2     2
               + x2 *x1 *( - u4  - u3  + u2  + u1 )

                   2            2          2          2          2      2
               + x2 *x1*(2*u5*u4  + 2*u5*u3  - 2*u3*u2  - 2*u3*u1 ) + x2 *(

                      2   2     2   2     2   2     2   2     2   2     2   2
                  - u6 *u4  - u6 *u3  - u5 *u4  - u5 *u3  + u4 *u2  + u4 *u1

                      2   2     2   2
                  + u3 *u2  + u3 *u1 )

                      2            2          2       2          2
               + x2*x1 *( - 2*u5*u2  - 2*u5*u1  + 2*u4 *u1 + 2*u3 *u1) + x2*x1

                          2             2                2             2
              *( - 4*u5*u4 *u1 - 4*u5*u3 *u1 + 4*u5*u3*u2  + 4*u5*u3*u1 ) + x2*(

                     2   2          2   2          2   2          2   2
                 2*u6 *u4 *u1 + 2*u6 *u3 *u1 + 2*u5 *u4 *u1 + 2*u5 *u3 *u1

                           2   2          2   2          2   2          2   2
                  - 2*u5*u4 *u2  - 2*u5*u4 *u1  - 2*u5*u3 *u2  - 2*u5*u3 *u1 ) +

                 2    2   2     2   2     2   2     2   2     2   2     2   2
               x1 *(u6 *u2  + u6 *u1  + u5 *u2  + u5 *u1  - u4 *u2  - u4 *u1

                      2   2     2   2               2      2       2      2
                  - u3 *u2  - u3 *u1 ) + x1*( - 2*u6 *u3*u2  - 2*u6 *u3*u1

                        2      2       2      2          2   2          2   2
                  - 2*u5 *u3*u2  - 2*u5 *u3*u1  + 2*u5*u4 *u2  + 2*u5*u4 *u1

                           2   2          2   2
                  + 2*u5*u3 *u2  + 2*u5*u3 *u1 )

showtime;


Time: 0 ms  plus GC time: 20 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({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};


gb := {(u2 - u6)*x3 + (u1*u6 - u2*u5),

        - (u4 - u6)*x2 - (u3*u6 - u4*u5),
```

```
       (u2 - u4)*x1 + (u1*u4 - u2*u3)}


%---------------------------------------------------------;
% 「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;


{ - u4 + u2, - u6 + u4, - u6 + u2}



%%% gb を法として g を簡約 ------------------------------------------------;
preduce(conclusion, gb);


      6   4   2       6   3   3     6   2   4       5         4
 - (u1 *u4 *u6  - 2*u1 *u4 *u6  + u1 *u4 *u6  + 2*u1 *u2*u3*u4 *u6


（中略）

          2   2   2       2       3     2   4          4             3   2
    - 6*u2 *u4 *u6  + 2*u2 *u4*u6  + u2 *u6  - 2*u2*u4 *u6 + 2*u2*u4 *u6

             2   3             4     4   2       3   3     2   4
    + 2*u2*u4 *u6  - 2*u2*u4*u6  + u4 *u6  - 2*u4 *u6  + u4 *u6 )

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

showtime;


Time: 40 ms  plus GC time: 30 ms


;

end;
```