主専攻実習（定理証明班）

第七回課題レポート

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 26 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

```
% [定理 09] ----------------------------------------------------------;
% 三角形の内角または外角の２等分線は、対辺を他の２辺の比に分ける
% △ABC において、∠A およびその外角の２等分線が BC および
% その延長を交わる点を D,E とする
% ※ただし、座標軸上で点 E が x 軸上に存在するように各点を取る
% A(0, 0) B(u1, u2) C(u3, u4)
% D(x1, x2) E(x3, 0) F(x5, x6) G(x6, x7)
%--------------------------------------------------------------------;

% 関数定義読み込み） ---------------------------------------------------;

load_package groebner;
on comp,gcd,ezgcd;
off allfac,pwrds;

in cal_sys_relations$


% ２点間のユークリッド距離 D^2 ---------------------------------------------;
procedure squared_euclid(a1,a2,b1,b2)$
begin
 scalar d;
 d:=(a1-b1)^2+(a2-b2)^2;
 return d
end$
%--------------------------------------------------------------------;


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

order x7, x6, x5, x4, x3, x2, x1, u4, u3, u2, u1;
factor x7, x6, x5, x4, x3, x2, x1;

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

```
% 点Cを通ってAĐに対して平行に引いた直線とABとの交点をFとする
% AĐ//FC かつ B-A-F かつ B-Đ-C

h1:=parallel(0, 0, x1, x2, x4, x5, u3, u4);
h2:=collinear(u1, u2, 0, 0, x4, x5);
h3:=collinear(u1, u2, x1, x2, u3, u4);

% 点Cを通ってAEに対して平行に引いた直線とABとの交点をGとする
% AE//GC かつ B-C-E かつ B-G-A

h4:=parallel(0, 0, x3, 0, x6, x7, u3, u4);
h5:=collinear(u1, u2, u3, u4, x3, 0);
h6:=collinear(u1, u2, x6, x7, 0, 0);

% ∠AFC=∠BAĐ=∠CAĐ=∠ACF から  ∴ AF=AC
% ∠AGC=∠FAE=∠CAE=∠ACG から  ∴ AG=AC

h7:=requal(0, 0, x4, x5, 0, 0, u3, u4);
h8:=requal(0, 0, x6, x7, 0, 0, u3, u4);

% よって、△AGC, △ACEは二等辺三角形となる
% ここで、二等辺三角形の頂角の二等分線は底辺と直角に交わることから、
% AĐ⊥GC , AE⊥CF である

h9:=vertically(0, 0, x1, x2, x6, x7, u3, u4);
h10:=vertically(0, 0, x3, 0, u3, u4, x4, x5);


% --------------------------------------------------------------------;
% 結論（内角の2等分線）------------------------------------------------;

% 内角の2等分線が、対辺を他の2辺の比に分ける場合、
% BĐ:ĐC=AB:AC  <=>  (BĐ*AC)^2 = (AB*ĐC)^2

bd2:=squared_euclid(u1, u2, x1, x2);
dc2:=squared_euclid(x1, x2, u3, u4);
ab2:=squared_euclid(0, 0, u1, u2);
ac2:=squared_euclid(0, 0, u3, u4);

in_conclusion:=bd2*ac2-ab2*dc2;


%---------------------------------------------------------------------;
% 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({x1, x2, x3, x4, x5, x6, x7}, 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, h7, h9, h6};

%---------------------------------------------------------;
% 「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(in_conclusion, gb);

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



% ####################################################################



% --------------------------------------------------------------------;
% 結論（外角の2等分線）------------------------------------------------;

% 外角の2等分線が、対辺を他の2辺の比に分ける場合、
% BE:EC=AB:AC  <=>  (BE*AC)^2 = (AB*EC)^2
```

```
be2:=squared_euclid(u1, u2, x3, 0);
ec2:=squared_euclid(x3, 0, u3, u4);
% ab2:=squared_euclid(0, 0, u1, u2); % 既出
% ac2:=squared_euclid(0, 0, u3, u4); % 既出

ex_conclusion:=be2*ac2-ab2*ec2;


%----------------------------------------------------------------------;
% 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({x1, x2, x3, x4, x5, x6, x7}, lex)$

%%% 仮定において定義した式から Groebner Basis を求める ------------------------;
gb:=groebner{h4, h5, h6, h7, h10};

%---------------------------------------------------------;
% 「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(ex_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 09] ------------------------------------------------------------;
% 三角形の内角または外角の２等分線は、対辺を他の２辺の比に分ける
% △ABC において、∠A およびその外角の２等分線が BC および
% その延長を交わる点を Ð,E とする
% ※ただし、座標軸上で点 E が x 軸上に存在するように各点を取る
% A(0, 0) B(u1, u2) C(u3, u4)
% Ð(x1, x2) E(x3, 0) F(x5, x6) G(x6, x7)
%----------------------------------------------------------------------;

% 関数定義読み込み） ---------------------------------------------------;

load_package groebner;


on comp,gcd,ezgcd;


off allfac,pwrds;



in cal_sys_relations$




% ２点間のユークリッド距離 Ð^2 ----------------------------------------------;
procedure squared_euclid(a1,a2,b1,b2)$
begin
 scalar d;
 d:=(a1-b1)^2+(a2-b2)^2;
 return d
end$


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

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

order x7, x6, x5, x4, x3, x2, x1, u4, u3, u2, u1;


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



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

% 点 C を通って AĐ に対して平行に引いた直線と AB との交点を F とする
% AĐ//FC かつ B-A-F かつ B-Đ-C

h1:=parallel(0, 0, x1, x2, x4, x5, u3, u4);


h1 :=  - x5*x1 + x4*x2 - x2*u3 + x1*u4

h2:=collinear(u1, u2, 0, 0, x4, x5);


h2 :=  - x5*u1 + x4*u2

h3:=collinear(u1, u2, x1, x2, u3, u4);


h3 := x2*( - u3 + u1) + x1*(u4 - u2) - u4*u1 + u3*u2


% 点 C を通って AE に対して平行に引いた直線と AB との交点を G とする
% AE//GC かつ B-C-E かつ B-G-A

h4:=parallel(0, 0, x3, 0, x6, x7, u3, u4);


h4 :=  - x7*x3 + x3*u4

h5:=collinear(u1, u2, u3, u4, x3, 0);


h5 := x3*( - u4 + u2) + u4*u1 - u3*u2

h6:=collinear(u1, u2, x6, x7, 0, 0);


h6 := x7*u1 - x6*u2


% ∠AFC=∠BAĐ=∠CAĐ=∠ACF から  ∴ AF=AC
% ∠AGC=∠FAE=∠CAE=∠ACG から  ∴ AG=AC

h7:=requal(0, 0, x4, x5, 0, 0, u3, u4);


         2     2     2     2
h7 := x5  + x4  - u4  - u3

h8:=requal(0, 0, x6, x7, 0, 0, u3, u4);


         2     2     2     2
h8 := x7  + x6  - u4  - u3


% よって、△AGC, △ACE は二等辺三角形となる
% ここで、二等辺三角形の頂角の二等分線は底辺と直角に交わることから、
% AĐ⊥GC , AE⊥CF である

h9:=vertically(0, 0, x1, x2, x6, x7, u3, u4);


h9 :=  - x7*x2 - x6*x1 + x2*u4 + x1*u3

h10:=vertically(0, 0, x3, 0, u3, u4, x4, x5);
```

```
h10 := x4*x3 - x3*u3


% ---------------------------------------------------------------------;
% 結論（内角の 2 等分線）------------------------------------------------;

% 内角の 2 等分線が、対辺を他の 2 辺の比に分ける場合、
% BĐ:ĐC=AB:AC  <=>  (BĐ*AC)^2 = (AB*ĐC)^2

bd2:=squared_euclid(u1, u2, x1, x2);


           2             2             2     2
bd2 := x2   - 2*x2*u2 + x1   - 2*x1*u1 + u2   + u1

dc2:=squared_euclid(x1, x2, u3, u4);


           2             2             2     2
dc2 := x2   - 2*x2*u4 + x1   - 2*x1*u3 + u4   + u3

ab2:=squared_euclid(0, 0, u1, u2);


           2     2
ab2 := u2   + u1

ac2:=squared_euclid(0, 0, u3, u4);


           2     2
ac2 := u4   + u3

in_conclusion:=bd2*ac2-bd2*dc2;


                    2     2     2     2     2
in_conclusion := x2 *(u4  + u3  - u2  - u1 )

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

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

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


%---------------------------------------------------------------------;
% 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({x1, x2, x3, x4, x5, x6, x7}, 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, h7, h9, h6};


               2          3          2          2
gb := { - (u1 *u2 + u2  - u2*u3  - u2*u4 )*x1

              3     2              2     2
        + (u1  - u1 *u3 + u1*u2  - u2 *u3)*x7

              2                 2             2     3
        + (u1 *u2*u3 - u1*u2*u3  - u1*u2*u4  + u2 *u3),

          2          3          2          2            2          2          3     2
       (u1 *u2 + u2  - u2*u3  - u2*u4 )*x2 - (u1 *u2 - u1 *u4 + u2  - u2 *u4)*x7
```

```
             2            3        2    2      2   2
        - (u1 *u2*u4 + u2 *u4 - u2 *u3  - u2 *u4 ),

       u2*x4 + u1*x7,

       x5 + x7,

        - u2*x6 + u1*x7,

          2      2    2       2    2      2    2
       (u1  + u2 )*x7  - (u2 *u3  + u2 *u4 )}

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


{u2,

  - u4 + u2,

 u1,

   2     2
 u2  + u1 ,

 u3,

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

 u4,

     2     2     2     2
  - u4  - u3  + u2  + u1 ,

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


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


0


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


% #####################################################################


% ----------------------------------------------------------------------;
% 結論（外角の 2 等分線）--------------------------------------------------;

% 外角の 2 等分線が、対辺を他の 2 辺の比に分ける場合、
% BE:EC=AB:AC  <=>  (BE*AC)^2 = (AB*EC)^2

be2:=squared_euclid(u1, u2, x3, 0);


               2                 2     2
be2 := x3  - 2*x3*u1 + u2  + u1

ec2:=squared_euclid(x3, 0, u3, u4);
```

```
             2              2     2
ec2 := x3  - 2*x3*u3 + u4  + u3

% ab2:=squared_euclid(0, 0, u1, u2); % 既出
% ac2:=squared_euclid(0, 0, u3, u4); % 既出

ex_conclusion:=be2*ac2-ab2*ec2;


ex_conclusion :=

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



%----------------------------------------------------------------------;
% 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({x1, x2, x3, x4, x5, x6, x7}, lex)$



%%% 仮定において定義した式から Groebner Basis を求める -------------------------;
gb:=groebner{h4, h5, h6, h7, h10};


gb := {(u2 - u4)*x3 + (u1*u4 - u2*u3),

       x4 - u3,

         2     2
       x5  - u4 ,

        - u2*x6 + u1*u4,

       x7 - u4}


%-----------------------------------------------------------;
% 「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,u2,u4,u3}



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


       4   2        3        2      2   2   2      2   2   2        2       3
 - (u1 *u4  - 2*u1 *u3*u4  - u1 *u2 *u3  + u1 *u2 *u4  - 2*u1 *u2*u4

         2   2   2      2   4           2   3      4   2        3   2        2   4
     + u1 *u3 *u4  + u1 *u4  + 2*u1*u2 *u3  - u2 *u3  + 2*u2 *u3 *u4 - u2 *u3

         2   2   2       2              2
     - u2 *u3 *u4 )/(u2  - 2*u2*u4 + u4 )

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

showtime;


Time: 0 ms  plus GC time: 9 ms

;

end;
```