```
unit rules250;

 interface

  uses
    globals,cmmnds1,pusherr,pushStack,ruleAtoF;

   procedure r201; procedure r202; procedure r203; procedure
r204; procedure r205;
     procedure r206; procedure r207; procedure r208; procedure
r209; procedure r210;
     procedure r211; procedure r212; procedure r213; procedure
r214; procedure r215;
     procedure r216; procedure r217; procedure r218; procedure
r219; procedure r220;
     procedure r221; procedure r222; procedure r223; procedure
r224; procedure r225;
     procedure r226; procedure r227; procedure r228; procedure
r229; procedure r230;
     procedure r231; procedure r232; procedure r233; procedure
r234; procedure r235;
     procedure r236; procedure r237; procedure r238; procedure
r239; procedure r240;
     procedure r241; procedure r242; procedure r243; procedure
r244; procedure r245;
     procedure r246; procedure r247; procedure r248; procedure
r249; procedure r250;

implementation

procedure r201;
(*************************************************)
(*                                               *)
(*  if P > 5 , connected, and                    *)
(*      P > 3*eind-2 then                        *)
(*           ncov <= 2*eind-mindeg               *)
(*                                               *)
(*************************************************)
begin
  if (activerule[201]) and (max[connct]=1) then
    begin
      rule:='201/ ';
      if min[connct] = 1 then
       begin
         if min[nodes] >= 6 then
            begin
              z:=(min[ncov]+min[mindeg]+1) div 2;
              if z > (min[nodes]+4) div 3 then z:=(min[nodes]+4)
div 3;
              if z > min[eind] then pushmin(eind);
            end;
```

```
         if min[ncov] > 2*max[eind]-min[mindeg] then
            begin
              z:=3*max[eind]-2;
              if z < 6 then z:=5;
              if z < max[nodes] then pushmax(nodes);
            end;
        end;
      if min[nodes] >= 6 then
         begin
           z1:=3*max[eind]-2;
           z:=2*max[eind]-min[mindeg];
           if (min[connct]=1) and (min[nodes] > z1) then
              begin
                if z < max[ncov] then pushmax(ncov);
                z:=2*max[eind]-min[ncov];
                if z < max[mindeg] then pushmax(mindeg);
              end
            else
              if min[ncov] > z then
                if min[nodes] > z1 then
                   begin
                     z:=0;
                     pushmax(connct);
                   end;
         end;
    end;
end;

procedure r202;
(*****************************************************)
(*                                                   *)
(*    if reg then                                    *)
(*       ncov >= P/2+(clique-1)*(clique-2)/2*mindeg  *)
(*                                                   *)
(*****************************************************)
begin
  if (activerule[202]) and (max[reg]=1) then
    begin
      rule:='202/ ';
      z1:=(min[clique]-1)*(min[clique]-2);
      z:=(min[nodes]*max[mindeg]+z1-1) div (2*max[mindeg])+1;
      if min[reg]=1 then
         begin
           if z > min[ncov] then pushmin(ncov);
           if (max[mindeg] < infinity) and (max[ncov] < infinity)
then
              begin
                z:=(2*max[mindeg]*max[ncov]-z1) div max[mindeg];
                if z < max[nodes] then pushmax(nodes);
              end;
           if max[ncov] < infinity then
             begin
```

```
                k:=2*max[ncov]-min[nodes];
                if k > 0 then
                   begin
                     z:=(z1+k-1) div k;
                     if z > min[mindeg] then pushmin(mindeg);
                   end;
                if max[mindeg] < infinity then
                   begin
                     rz:=max[mindeg];
                     rz:=1+4*k*rz;
                     if rz < infinity then
                        begin
                          k:=1+4*k*max[mindeg];
                          if k >= 0 then
                            begin
                              z:=trunk((3+sqrt(k))/2);
                              if z < max[clique] then
pushmax(clique);
                            end;
                        end;
                   end;
             end;
          end
        else
          if max[ncov] < z then
               begin
                 z:=0;
                 pushmax(reg);
               end;
    end;
end;

procedure r203;
(*************************************************)
(*                                               *)
(*  if clique=2 then                             *)
(*        ncov <= (2*P+3-sqrt(8*P+9))/2          *)
(*                                               *)
(*************************************************)
begin
  if (activerule[203]) and (min[clique]=2) then
    begin
      rule:='203/ ';
      z1:=max[nodes];
      if max[clique]=2 then
          begin
            if z1 < infinity then
               begin
                 z:=trunk((2*z1+3-sqrt(8*z1+9))/2);
                 if z < max[ncov] then pushmax(ncov);
               end;
            z1:=min[ncov];
```

```
            z:=round((2*z1-1+sqrt(8*z1+1))/2+hf);
            if z > min[nodes] then pushmin(nodes);
          end
        else
          if z1 < infinity then
              if min[ncov] > trunk((2*z1+3-sqrt(8*z1+9))/2) then
                     begin
                       z:=3;
                       pushmin(clique);
                     end;
    end;
end;

procedure r204;
(*************************************************)
(*                                               *)
(*     if clique=2 and P/2 < ncov <=3*P/5        *)
(*          ncov<=(3*P-sqrt(5*E-P*P))/5          *)
(*                                               *)
(*************************************************)
begin
  if (activerule[204]) and (min[clique]=2) and (max[nodes]
< infinity) then
    begin
      rule:='204/ ';
      z1:=max[nodes];
      if max[clique]=2 then
             begin
               z:=min[nodes];
               if max[ncov] <= 3*z div 5 then
                  begin
                    z:=5*min[edges]-z1*z1;
                    if z >= 0 then
                        begin
                          z:=trunk((3*z1-sqrt(z))/5);
                          if z < (z1+1) div 2 then z:=(z1+1) div
2;
                          if z < max[ncov] then pushmax(ncov);
                        end;
                    if min[ncov] > (z1+1) div 2 then
                      begin
                        z:=2*z1*z1-6*min[ncov]*z1+5
*min[ncov]*min[ncov];
                        if z < max[edges] then pushmax(edges);
                        z:=min[ncov];
                        k:=2*min[edges]-z*z;
                        k1:=max[ncov];
                        k1:=5*k1*k1-9*min[edges];
                        if k1 <= 0 then
                            begin
                              z:=round((3*z+sqrt(k))/2+hf);
                              if z > min[nodes] then
```

```
pushmin(nodes);
                              end;
                         end;
                    end;
          end;
    end;
end;

procedure r205;
(************************************************)
(*                                              *)
(*  if mindeg=2 then                            *)
(*      ecov <= P*(z/(2+z))                     *)
(*         where z= max(4,maxdeg)               *)
(*                                              *)
(************************************************)
begin
  if (activerule[205]) and (min[mindeg] = 2) and
     (max[mindeg] = 2) and (max[maxdeg] < infinity) then
    begin
      rule:='205/ ';
      z1:=max[nodes];
      k:=4;
      if max[maxdeg] > 4 then k:=max[maxdeg];
      if z1 < infinity then
                begin
                  z:=(z1*k) div (2+k);
                  if z < max[ecov] then pushmax(ecov);
                 end;
      z:=(min[ecov]*(2+k)-1) div k+1;
      if z > min[nodes] then pushmin(nodes);
    end;
end;

procedure r206;
(************************************************)
(*                                              *)
(*     ncov <= (P*maxdeg+1)/(maxdeg+1)-1/(mindeg+1)  *)
(*                                              *)
(************************************************)
begin
  if activerule[206] then
    begin
      rule:='206/ ';
      z1:=max[nodes];
      k:=max[maxdeg];
      k1:=max[mindeg];
      if (k < infinity) and (k1 < infinity) then
           begin
             z:=round(((min[ncov]+1/(k1+1))*(k+1)-1)/k+hf);
             if z > min[nodes] then pushmin(nodes);
           end;
```

```
      if z1 < infinity then
         begin
           if k < infinity then
               begin
                 z:=min[ncov];
                 z:=k div (k*(z1-z)-z+1);
                 if z > min[mindeg] then pushmin(mindeg);
                 if k1 < infinity then
                      begin
                        z:=trunk((z1*k+1)/(k+1)-1/(k1+1));
                        if z < max[ncov] then pushmax(ncov);
                      end;
               end;
             if k1 < infinity then
                begin
                  z:=min[ncov];
                  z:=round(((z-1)*(k1+1)+1)/((k1+1)*(z1-
z)-1)+hf);
                  if z > min[maxdeg] then pushmin(maxdeg);
               end;
          end;
    end;
end;

procedure r207;
(*************************************************)
(*                                               *)
(*   if clique=2 then                            *)
(*       if maxdeg >= 3 then                     *)
(*          ncov <=P(maxdeg-6/5)/(maxdeg-1/5)    *)
(*       if connect,not odd cycle nor            *)
(*              even path  and P > 2             *)
(*          ncov <=P(maxdeg-1)/maxdeg +          *)
(*                   1/(maxdeg+1)-1/(mindeg+1)   *)
(*                                               *)
(*************************************************)
begin
  if (activerule[207]) and (min[clique]=2) then
    begin
      rule:='207/ ';
      if max[clique]=2 then
        begin
          z1:=max[nodes];
          k:=max[maxdeg];
          if min[maxdeg] > 2 then
              begin
                if k < infinity then
                   begin
                     if z1 < infinity then
                         begin
                           z:=trunk(z1*(k-1.2)/(k-0.2));
                           if z < max[ncov] then pushmax(ncov);
```

```
                        end;
                     z:=round(min[ncov]*(k-0.2)/(k-1.2)+hf);
                     if z > min[nodes] then pushmin(nodes);
                   end;
                if z1 < infinity then
                     begin
                       z:=(6*z1-min[ncov]-1) div (5*(z1-
min[ncov]))+1;
                       if z > min[maxdeg] then pushmin(maxdeg);
                     end;
              end;
          if (min[connct]=1) and (min[nodes] > 2) and
             ((max[cycle]=0) or ((min[cycle]=1) and
(max[nodes]=min[nodes])
              and (not(odd(min[nodes])))))) and ((min[edges] >=
max[nodes]) or
              (min[maxdeg] > 2) or
              ((max[nodes]=min[nodes]) and (odd(max[nodes]))))
then
                   begin
                     k1:=max[mindeg];
                     if (k < infinity) and (k1 < infinity) then
                          begin
                            if z1 < infinity then
                               begin
                                 z:=trunc(z1*(k-1)/k+1/(k+
1)-1/(k1+1));
                                 if z < max[ncov] then
pushmax(ncov);
                               end;
                            rz:=min[ncov]-1/(k+1)+1/(k1+1);
                            z:=round(rz*k/(k-1)+hf);
                            if z > min[nodes] then
pushmin(nodes);
                          end;
                     if (k < infinity) and (z1 < infinity) then
                          begin
                            rz:=z1;
                            rz:=rz*(k-1)/k+1.0/(k+1)-min[ncov];
                            z:=round(1/rz+hf)-1;
                            if z > min[mindeg] then
pushmin(mindeg);
                          end;
                   end;
         end;
    end;
end;

procedure r208;
(************************************************)
(*                                              *)
(*   if connected and not complete then         *)
```

```
(*        ncov <= (2EP**2 -3P-1)/(2EP+P**2)     *)
(*                                              *)
(************************************************)
begin
  if (activerule[208]) and (max[connct] = 1) then
    begin
      rule:='208/ ';
      z:=max[nodes];
      if (z < infinity) and (min[connct]=1) and (max[compl]=0)
then
             begin
               rz:=max[edges];
               if rz < infinity then
                   begin
                     rz:=rz*2*z;
                     z:=trunk(((rz-3)*z-1)/(rz+z*z));
                     if z < max[ncov] then pushmax(ncov);
                   end;
                rz:=max[nodes];
                rz:=min[ncov]*rz*rz+3*rz+1;
                z:=round(rz/(2*rz*(rz-min[ncov]))+hf);
                if z > min[edges] then pushmin(edges);
             end;
    end;
end;

procedure r209;
(************************************************)
(*                                              *)
(*   ncov <= P*(1-2/(maxdeg+clique+1))          *)
(*                                              *)
(************************************************)
begin
  if activerule[209] then
    begin
      rule:='209/ ';
      k:=max[maxdeg]+max[clique]+1;
      if k < infinity then
         begin
           z:=max[nodes];
           if z < infinity then
             begin
               z:=trunk(z*(1-2/k));
               if z < max[ncov] then pushmax(ncov);
             end;
           z:=round(min[ncov]/(1-2/k)+hf);
           if z > min[nodes] then pushmin(nodes);
         end;
       k:=max[nodes];
       if k < infinity then
          begin
            z:=max[clique];
```

```
            k:=(2*k-1) div (k-min[ncov]);
            if z < infinity then
                begin
                  z:=k-z;
                  if z > min[maxdeg] then pushmin(maxdeg);
                end;
            z:=max[maxdeg];
            if z < infinity then
                begin
                  z:=k-z;
                  if z > min[clique] then pushmin(clique);
                end;
          end;
    end;
end;

procedure r210;
(*******************************************************)
(*                                                     *)
(*    ncov <= ((P-2)maxdeg+clique+mindeg-1)/(maxdeg+1)  *)
(*                                                     *)
(*******************************************************)
begin
  if activerule[210] then
    begin
      rule:='210/ ';
      k:=max[nodes];
      k1:=max[maxdeg];
      k2:=max[clique];
      k3:=max[mindeg];
      if k < infinity then
        begin
          k4:=k-min[ncov]-2;
          if k1 < infinity then
              begin
                if k2 < infinity then
                    begin
                      if (k3 < infinity) and (max[compl]=0) then
                         begin
                           rz:=k-2;
                           rz:=rz*k1+k2+k3-1;
                           z:=trunk(rz/(k1+1));
                           if z < max[ncov] then pushmax(ncov);
                         end;
                      z:=min[ncov]-k3+1-k4*k1;
                      if z > min[clique] then pushmin(clique);
                    end;
                 if k3 < infinity then
                    begin
                      z:=min[ncov]-k2+1-k4*k1;
                      if z > min[mindeg] then pushmin(mindeg);
                    end;
```

```
              end;
          if (k2 < infinity) and (k3 < infinity) and
             (k4 > 0) then
                begin
                  z:=(min[ncov]-k2-k3) div k4+1;
                  if z > min[maxdeg] then pushmin(maxdeg);
                end;
        end;
      if (k1 < infinity) and (k2 < infinity) and (k3 < infinity)
and
         (max[compl]=0) then
          begin
            z:=min[ncov]-k2-k3+k1-1;
            if z >= k1 then
                begin
                  z:=z div k1+min[ncov]+2;
                  if z > min[nodes] then pushmin(nodes);
                end;
          end;
    end;
end;

procedure r211;
(*********************************************)
(*                                           *)
(*   if  ncov > P-nccov then                 *)
(*     ncov <=P*maxdeg/(maxdeg+1) -1/3       *)
(*                                           *)
(*********************************************)
begin
  if activerule[211] then
    begin
      rule:='211/ ';
      z:=max[nodes];
      rz:=max[maxdeg];
      z1:=-1;
      if (z < infinity) and (rz < infinity) then
        begin
          z1:=z-min[nccov];
          z:=trunk(z*rz/(rz+1)-1/3);
          if z < z1 then z:=z1;
          if z < max[ncov] then pushmax(ncov);
        end;
      if min[ncov] > max[nodes]-min[nccov] then
             begin
               rz:=max[maxdeg];
               if rz < infinity then
                 begin
                   z:=round((min[ncov]+1/3)*(rz+1)/rz+hf);
                   if z > min[nodes] then pushmin(nodes);
                 end;
               z:=max[nodes];
```

```
              k:=min[ncov];
              z:=(3*k) div (3*z-3*k-1)+1;
              if z > min[maxdeg] then pushmin(maxdeg);
            end
          else
             if z1 > 0 then
               begin
                 rz:=max[maxdeg];
                 z:=trunk(max[nodes]*rz/(rz+1)-1/3);
                 if min[ncov] > z then
                   begin
                     z:=max[nodes]-min[ncov];
                     if z < max[nccov] then pushmax(nccov);
                     z:=min[ncov]+min[nccov];
                     if z > min[nodes] then pushmin(nodes);
                   end;
                end;
    end;
end;

procedure r212;
(*******************************)
(*                             *)
(*       nccov <= ecov         *)
(*                             *)
(*******************************)
begin
  if activerule[212] then
    begin
      rule:='212/ ';
      z:=max[ecov];
      if z < max[nccov] then pushmax(nccov);
      z:=min[nccov];
      if z > min[ecov] then pushmin(ecov);
    end;
end;

procedure r213;
(*************************)
(*                       *)
(*      dom # <= eind    *)
(*                       *)
(*************************)
begin
  if (activerule[213]) and ((max[dom] > max[eind])
       or (min[eind] < min[dom])) then
    begin
      rule:='213/ ';
      rulea(dom,eind,0);
    end;
end;
```

```
procedure r214;
(*****************************)
(*                           *)
(*       dom  <= nind        *)
(*                           *)
(*****************************)
begin
  if (activerule[214]) and ((max[dom] > max[nind])
      or (min[nind] < min[dom])) then
    begin
      rule:='214/ ';
      rulea(dom,nind,0);
    end;
end;

procedure r215;
(*************************)
(*                       *)
(*     Ncomp <= dom      *)
(*                       *)
(*************************)
begin
  if (activerule[215]) and ((max[ncomp] > max[dom])
         or (min[dom] < min[ncomp])) then
    begin
      rule:='215/ ';
      rulea(ncomp,dom,0);
    end;
end;

procedure r216;
(*************************)
(*                       *)
(*     maxdeg <= echr    *)
(*                       *)
(*************************)
begin
  if (activerule[216]) and ((max[maxdeg] > max[echr])
       or (min[echr] < min[maxdeg])) then
    begin
      rule:='216/ ';
      rulea(maxdeg,echr,0);
    end;
end;

procedure r217;
(*******************************)
(*                             *)
(*       echr <= maxdeg + 1    *)
(*                             *)
(*******************************)
begin
```

```
  if (activerule[217]) and ((max[echr] > max[maxdeg]+1)
         or (min[maxdeg]+1 < min[echr])) then
    begin
      rule:='217/ ';
      rulea(echr,maxdeg,1);
    end;
end;

procedure r218;
(*****************************)
(*                           *)
(*     mindeg <= ncov        *)
(*                           *)
(*****************************)
begin
  if (activerule[218]) and ((max[mindeg] > max[ncov])
     or (min[ncov] < min[mindeg])) then
    begin
      rule:='218/ ';
      rulea(mindeg,ncov,0);
    end;
end;

procedure r219;
(*****************************)
(*                           *)
(*      Econn <= mindeg      *)
(*                           *)
(*****************************)
begin
  if (activerule[219]) and ((max[econn] > max[mindeg])
      or (min[mindeg] < min[econn])) then
    begin
      rule:='219/ ';
      rulea(econn,mindeg,0);
    end;
end;

procedure r220;
(*****************************)
(*                           *)
(*       clique <= chr       *)
(*                           *)
(*****************************)
begin
  if (activerule[220]) and ((max[clique] > max[chr])
       or (min[chr] < min[clique])) then
    begin
      rule:='220/ ';
      rulea(clique,chr,0);
    end;
end;
```

```
procedure r221;
(*****************************)
(*                           *)
(*     chr <= ncov+1         *)
(*                           *)
(*****************************)
begin
  if (activerule[221]) and ((max[chr] > max[ncov]+1)
      or (min[ncov]+1 < min[chr])) then
    begin
      rule:='221/ ';
      rulea(chr,ncov,1);
    end;
end;


procedure r222;
(*************************)
(*                       *)
(*     eind <= ncov      *)
(*                       *)
(*************************)
begin
  if (activerule[222]) and ((max[eind] > max[ncov])
      or (min[ncov] < min[eind])) then
    begin
      rule:='222/ ';
      rulea(eind,ncov,0);
    end;
end;

procedure r223;
(************************)
(*                      *)
(*    nind <= nccov     *)
(*                      *)
(************************)
begin
  if (activerule[223]) and ((max[nind] > max[nccov])
     or (min[nccov] < min[nind])) then
    begin
      rule:='223/ ';
      rulea(nind,nccov,0);
    end;
end;

procedure r224;
(*************************)
(*                       *)
(*     nccov <= eccov    *)
(*                       *)
```

```
(***************************)
begin
  if (activerule[224]) and ((max[nccov] > max[eccov])
     or (min[eccov] < min[nccov])) then
    begin
      rule:='224/ ';
      rulea(nccov,eccov,0);
    end;
end;

procedure r225;
(**********************)
(*                    *)
(*     rad <= diam    *)
(*                    *)
(**********************)
begin
  if (activerule[225]) and ((max[radius] > max[diam])
        or (min[diam] < min[radius])) then
    begin
      rule:='225/ ';
      rulea(radius,diam,0);
    end;
end;

procedure r226;
(*************************)
(*                       *)
(*    Nconn <= Econn     *)
(*                       *)
(*************************)
begin
  if (activerule[226]) and ((max[nconn] > max[econn])
      or (min[econn] < min[nconn])) then
    begin
      rule:='226/ ';
      rulea(nconn,econn,0);
    end;
end;

procedure r227;
(***********************)
(*                     *)
(*    girth <= circ    *)
(*                     *)
(***********************)
begin
  if (activerule[227]) and ((max[girth] > max[circ])
     or (min[circ] < min[girth])) then
    begin
      rule:='227/ ';
      rulea(girth,circ,0);
```

```
    end;
end;

procedure r228;
(**********************)
(*                    *)
(*      chr <= circ     *)
(*                    *)
(**********************)
begin
  if (activerule[228]) and (( max[chr] > max[circ])
      or (min[circ] < min[chr])) then
    begin
      rule:='228/ ';
      rulea(chr,circ,0);
    end;
end;

procedure r229;
(***************************)
(*                         *)
(*     genus <= Xnum       *)
(*                         *)
(***************************)
begin
  if (activerule[229]) and ((max[genus] > max[xnum])
        or (min[xnum] < min[genus])) then
    begin
      rule:='229/ ';
      rulea(genus,xnum,0);
    end;
end;

procedure r230;
(******************************)
(*                            *)
(*      mindeg <= circ - 1    *)
(*                            *)
(******************************)
begin
  if (activerule[230]) and ((max[mindeg] > max[circ] -1)
      or (min[circ]-1 < min[mindeg])) then
    begin
      rule:='230/ ';
      z:=max[circ]-1;
      if (max[circ] < infinity) and (z < max[mindeg]) then
            pushmax(mindeg);
      z:=min[mindeg]+1;
      if z > min[circ] then pushmin(circ);
    end;
end;
```

```
procedure r231;
(*******************************)
(*                             *)
(*      chr <= Bwidth + 1      *)
(*                             *)
(*******************************)
begin
  if (activerule[231]) and ((max[chr] > max[bwidth]+1)
       or (min[bwidth]+1 < min[chr])) then
    begin
      rule:='231/ ';
      rulea(chr,bwidth,1);
    end;
end;

procedure r232;
(*************************)
(*                       *)
(*     mindeg <= Bwidth  *)
(*                       *)
(*************************)
begin
  if (activerule[232]) and ((max[mindeg] > max[bwidth])
      or (min[bwidth] < min[mindeg])) then
    begin
      rule:='232/ ';
      rulea(mindeg,bwidth,0);
    end;
end;

procedure r233;
(***********************)
(*                     *)
(*    P <= nind*chr    *)
(*                     *)
(***********************)
begin
  if activerule[233] then
    begin
      rule:='233/ ';
      ruleb(nodes,nind,chr);
    end;
end;

procedure r234;
(*************************)
(*                       *)
(*    P <= nccov*clique  *)
(*                       *)
(*************************)
begin
  if activerule[234] then
```

```
    begin
      rule:='234/ ';
      ruleb(nodes,nccov,clique);
    end;
end;

procedure r235;
(**************************)
(*                        *)
(*     E <= echr*eind     *)
(*                        *)
(**************************)
begin
  if activerule[235] then
    begin
      rule:='235/ ';
      ruleb(edges,echr,eind);
    end;
end;

procedure r236;
(******************************)
(*                            *)
(*      E <= ncov*maxdeg      *)
(*                            *)
(******************************)
begin
  if activerule[236] then
    begin
      rule:='236/ ';
      ruleb(edges,ncov,maxdeg);
    end;
end;

procedure r237;
(**********************************)
(*                                *)
(*      ncov <= Bwidth*nind       *)
(*                                *)
(**********************************)
begin
  if activerule[237] then
    begin
      rule:='237/ ';
      ruleb(ncov,bwidth,nind);
    end;
end;

procedure r238;
(*******************************)
(*                             *)
(*      chr <= spectr + 1      *)
```

```
(*                                *)
(**********************************)
begin
  if (activerule[238]) and ((max[chr] > lammax+1)
        or (min[chr] > lammin+1)) then
    begin
      rule:='238/ ';
      rulec(chr,1);
    end;
end;

procedure r239;
(**********************************)
(*                                *)
(*       P = ncov + nind          *)
(*                                *)
(**********************************)
begin
  if (activerule[239]) and ((max[nodes] <> max[ncov]+max[nind])
       or (min[nodes] <> min[ncov] + min[nind])) then
    begin
      rule:='239/ ';
      ruled(nodes,ncov,nind);
      rulee(nodes,ncov,nind,0);
    end;
end;

procedure r240;
(*******************************)
(*                             *)
(*    P = ecov + eind          *)
(*                             *)
(*******************************)
begin
  if (activerule[240]) and ((max[nodes] <> max[ecov]+max[eind])
    or (min[nodes] <> min[ecov]+min[eind])) then
    begin
      rule:='240/ ';
      ruled(nodes,ecov,eind);
      rulee(nodes,ecov,eind,0);
    end;
end;

procedure r241;
(**********************************)
(*                                *)
(*  if clique =2 then             *)
(*       chr <= (3*P+36) div 16   *)
(*                                *)
(**********************************)
begin
  if (activerule[241]) and (min[clique] <= 2) then
```

```
    begin
      rule:='241/ ';
      if max[clique]=2 then
          begin
            if max[nodes] < infinity then
                begin
                  z:=(3*max[nodes]+36) div 16;
                  if z < max[chr] then pushmax(chr);
                end;
            z:=(16*min[chr]+2) div 3-12;
            if z > min[nodes] then pushmin(nodes);
          end
        else
          if (min[chr] > (3*max[nodes]+36) div 16) and
             (max[nodes] < infinity) then
             begin
               z:=3;
               pushmin(clique);
             end;
    end;
end;

procedure r242;
(*******************************)
(*                             *)
(*    P >= nccov + chr - 1     *)
(*                             *)
(*******************************)
begin
  if (activerule[242]) and (( max[nccov]+max[chr]-1 > max[nodes])
      or (min[nodes] < min[nccov]+min[chr]-1)) then
    begin
      rule:='242/ ';
      rulee(nodes,nccov,chr,-1);
    end;
end;

procedure r243;
(*******************************)
(*                             *)
(*     P >= dom + maxdeg       *)
(*                             *)
(*******************************)
begin
  if (activerule[243]) and ((max[dom]+max[maxdeg] > max[nodes])
       or (min[nodes] < min[dom]+min[maxdeg])) then
    begin
      rule:='243/ ';
      rulee(nodes,dom,maxdeg,0);
    end;
end;
```

```
procedure r244;
(*********************************************)
(*                                           *)
(*   if nind=2 then nccov <= 3(P+12) div 16   *)
(*                                           *)
(*********************************************)
begin
  if (activerule[244]) and (min[nind] <= 2) then
    begin
      rule:='244/ ';
      if max[nind]=2 then
          begin
            if max[nodes] < infinity then
                begin
                  z:=(3*max[nodes]+36) div 16;
                  if z < max[nccov] then pushmax(nccov);
                end;
            z:=(16*min[nccov]+2) div 3-12;
            if z > min[nodes] then pushmin(nodes);
          end
        else
          if (min[nccov] > (3*max[nodes]+36) div 16) and
             (max[nodes] < infinity) then
             begin
               z:=3;
               pushmin(nind);
             end;
    end;
end;

procedure r245;
(*********************************************)
(*                                           *)
(*     if mindeg >= 2 then                   *)
(*        eccov <= 2(P-2+2*genus)-4(Ncomp-1)    *)
(*                                           *)
(*********************************************)
begin
  if (activerule[245]) and (max[mindeg] > 1) then
    begin
      rule:='245/ ';
      k:=max[nodes];
      k1:=max[genus];
      if min[mindeg] > 1 then
         begin
           if k < infinity then
             begin
               z:=round((min[eccov]/2+2*min[ncomp]-k)/2+hf);
               if z > min[genus] then pushmin(genus);
               if k1 < infinity then
                  begin
                    z:=2*(k+2*k1-2*min[ncomp]);
```

```
                    if z < max[eccov] then pushmax(eccov);
                    z:=trunk((k+2*k1-min[eccov]/2)/2);
                    if z < max[ncomp] then pushmax(ncomp);
                  end;
              end;
             if k1 < infinity then
                begin
                  z:=(min[eccov]+1) div 2+2*min[ncomp]-2*k1;
                  if z > min[nodes] then pushmin(nodes);
                end;
           end
         else
            if (k < infinity) and (k1 < infinity) then
              if min[eccov] > 2*(k+2*k1-2*min[ncomp]) then
                  begin
                    z:=1;
                    pushmax(mindeg);
                  end;
    end;
end;

procedure r246;
(*********************************************)
(*                                           *)
(*  if P >= 3 then                           *)
(*    eccov <= 2(P-2+2*genus)-(Ncomp-1)      *)
(*                                           *)
(*********************************************)
begin
  if activerule[246] then
    begin
      rule:='246/ ';
      k:=max[nodes];
      k1:=max[genus];
      if (min[nodes] > 2) and (k < infinity) then
          begin
            z:=(min[eccov]+min[ncomp]+6-2*k) div 4;
            if z > min[genus] then pushmin(genus);
            if k1 < infinity then
               begin
                 rz:=k1;
                 if 4*rz < infinity then
                     begin
                       z:=(min[eccov]+min[ncomp]+4-4*k1) div 2;
                       if z > min[nodes] then pushmin(nodes);
                     end;
                 rhb:=k;
                 rz:=2*rhb+4*k-3-min[ncomp];
                 if rz < infinity then
                  begin
                    z:=2*k+4*k1-3-min[ncomp];
                    if z < max[eccov] then pushmax(eccov);
```

```
                  z:=z+min[ncomp]-min[eccov];
                  if z < max[ncomp] then pushmax(ncomp);
                end;
              end;
          end;
    end;
end;

procedure r247;
(*************************************************)
(*                                               *)
(*   nind <= P/(1+mindeg/maxdeg)                 *)
(*                                               *)
(*************************************************)
begin
  if activerule[247] then
    begin
      rule:='247/ ';
      if max[nodes] < infinity then
         begin
           if max[maxdeg] < infinity then
             begin
               z:=trunk(max[nodes]/(1+min[mindeg]/max[maxdeg]));
               if z < max[nind] then pushmax(nind);
               z:=trunk(max[maxdeg]*(max[nodes]/min[nind]-1));
               if z < max[mindeg] then pushmax(mindeg);
             end;
           z:=(min[mindeg]*min[nind]-1) div (max[nodes]-
min[nind])+1;
           if z > min[maxdeg] then pushmin(maxdeg);
         end;
      if max[maxdeg] < infinity then
         begin
           z:=round(min[nind]*(1+min[mindeg]/max[maxdeg])+hf);
           if z > min[nodes] then pushmin(nodes);
         end;
    end;
end;

procedure r248;
(*************************************************)
(*                                               *)
(*  ncov >= P/(1+maxdeg/mindeg)                  *)
(*                                               *)
(*************************************************)
begin
  if activerule[248] then
    begin
      rule:='248/ ';
      if max[maxdeg] < infinity then
         begin
           z:=round(min[nodes]/(1+max[maxdeg]/min[mindeg])+hf);
```

```
            if z > min[ncov] then pushmin(ncov);
            if max[ncov] < infinity then
               begin
                 rz:=min[nodes]/max[ncov]-1;
                 if rz > 1 then
                   begin
                     z:=trunk(max[maxdeg]/rz);
                     if z < max[mindeg] then pushmax(mindeg);
                   end;
                 z:=trunk(max[ncov]*(1+max[maxdeg]/min[mindeg]));
                 if z < max[nodes] then pushmax(nodes);
               end;
         end;
       if max[ncov] < infinity then
            begin
               z:=round(min[mindeg]*(min[nodes]/max[ncov]-1)+hf);
               if z > min[maxdeg] then pushmin(maxdeg);
            end;
    end;
end;

procedure r249;
(*********************************************)
(*                                           *)
(*      nind >= P/(Bwidth+1)                 *)
(*                                           *)
(*********************************************)
begin
  if activerule[249] then
    begin
      rule:='249/ ';
      if max[bwidth] < infinity then
         begin
           z:=(min[nodes]-1) div (max[bwidth]+1)+1;
           if z > min[nind] then pushmin(nind);
           if max[nind] < infinity then
             begin
               z:=(min[nodes]-1) div max[nind];
               if z > min[bwidth] then pushmin(bwidth);
               z:=max[nind]*(max[bwidth]+1);
               if z < max[nodes] then pushmax(nodes);
             end;
         end;
      if max[nind] < infinity then
         begin
           z:=(min[nodes]-1) div max[nind];
           if z > min[bwidth] then pushmin(bwidth);
         end;
    end;
end;

procedure r250;
```

```
(*************************************************)
(*                                               *)
(*     ncov <= P/(1+1/Bwidth)                    *)
(*                                               *)
(*************************************************)
begin
  if activerule[250] then
    begin
      rule:='250/ ';
      if max[nodes] < infinity then
        begin
          if max[bwidth] < infinity then
            begin
              z:=trunk(max[nodes]/(1+1/max[bwidth]));
              if z < max[ncov] then pushmax(ncov);
            end;
          z:=round(1/(max[nodes]/min[ncov]-1)+hf);
          if z > min[bwidth] then pushmin(bwidth);
        end;
      if max[bwidth] < infinity then
          begin
            z:=round(min[ncov]*(1+1/max[bwidth])+hf);
            if z > min[nodes] then pushmin(nodes);
          end;
    end;
end;


end.
```