可平面图的判断

程龚 (gcheng@nju.edu.cn)

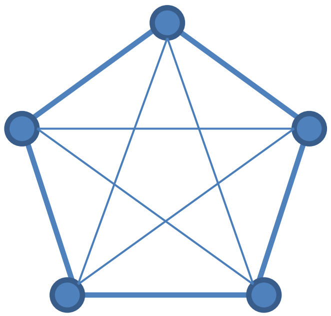
本节课的主要内容

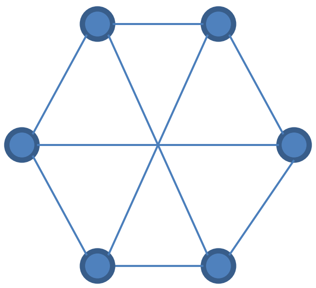
7.3 可平面图的判断

DMP算法

回顾

- 不可平面图
  - K_5
  - $K_{3,3}$

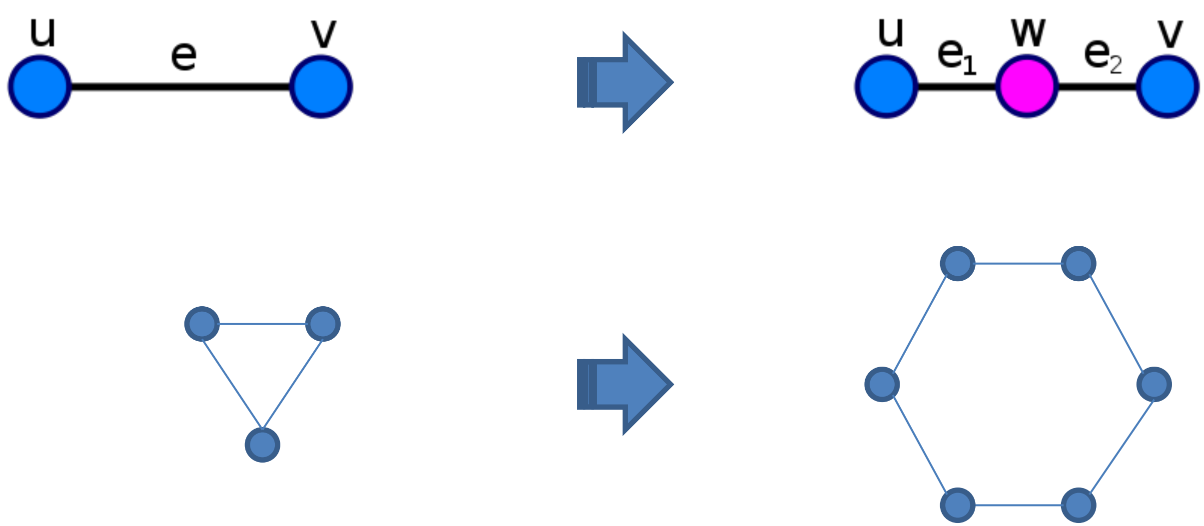




- 极小不可平面图 (minimal nonplanar graph)
  - 任何真子图都是可平面图
    你能举一些例子吗？
    - K_5、$K_{3,3}$
- 可平面图的子图一定是可平面图
  - K_5和$K_{3,3}$一定不是可平面图的子图

剖分

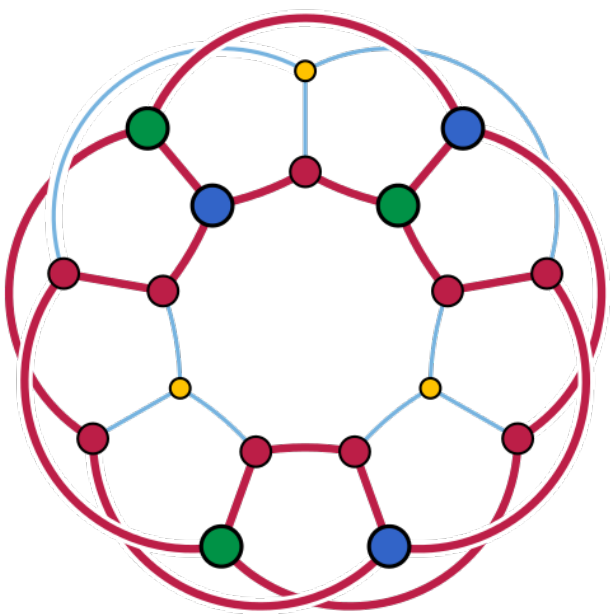
- 剖分 (subdivision)
  – 在一条边上加入一个新的顶点，将其分为两条边



https://en.wikipedia.org/wiki/Subdivision_(graph_theory)#Subdivisions

Kuratowski子图

- Kuratowski子图
  - K_5或$K_{3,3}$的剖分



https://upload.wikimedia.org/wikipedia/commons/thumb/7/73/GP92-Kuratowski.svg/500px-GP92-Kuratowski.svg.png

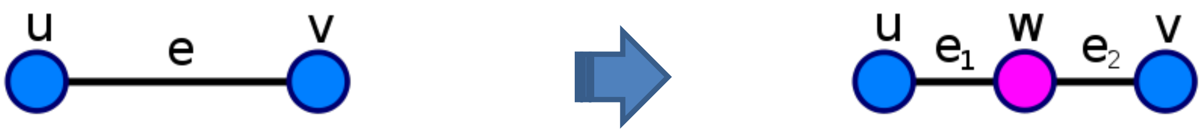
Kuratowski定理

- 可平面图的充要条件：没有Kuratowski子图。

证明：

$\Rightarrow$

剖分并不改变图的可平面性 $\Rightarrow$ K_5或$K_{3,3}$的剖分是不可平面图 $\Rightarrow$ 可平面图没有Kuratowski子图



剖分会不会改变图的可平面性？

Kuratowski定理 (续)

- 可平面图的充要条件：没有Kuratowski子图。

证明：

$\Leftarrow$*

反证法：假设存在一个不包含Kuratowski子图的不可平面图

1. 边最少的不包含Kuratowski子图的不可平面图一定是3-连通的。
2. 不包含Kuratowski子图的3-连通图一定是可平面图。

Kuratowski定理 (续)

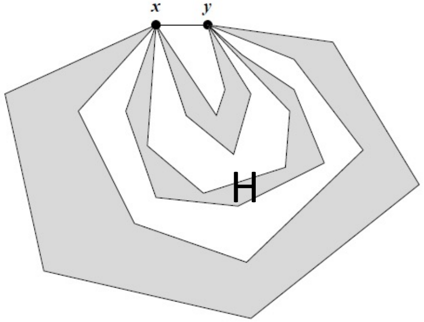
- 1. 边最少的不包含Kuratowski子图的不可平面图G一定是3-连通的。

证明：

1. 从G中删除边e不会引入Kuratowski子图 ⇒ G-e是可平面图 ⇒ G是极小不可平面图
2. （引理1）极小不可平面图是2-连通的 ⇒ G是2-连通的
3. 反证法：假设G不是3-连通的 ⇒ G有2-点割集S={x, y}
4. G-S的一个连通分支中的所有顶点和{x, y}的并在G中的导出子图称作一个S-lobe。
5. （引理2）对于不可平面图G的2-点割集S={x, y}，G有一个S-lobe+(x, y)是不可平面图，记作H。
6. S是G的最小点割集 ⇒ x和y在G-S的每个连通分支中都有相邻顶点 ⇒ ε(H)<ε(G) ⇒ H有Kuratowski子图F ⇒
   - 如果(x, y)在G中 ⇒ F是G的Kuratowski子图 ⇒ 矛盾
   - 如果(x, y)不在G中，怎么办？

     ⇒ 用另一个S-lobe中的x-y路替换F中的(x, y)，得到G的Kuratowski子图 ⇒ 矛盾

   ⇒ G是3-连通的



http://www.cs.rpi.edu/~goldberg/13-GT/18-kurat.pdf

Kuratowski定理 (续)

- 引理1：极小不可平面图是2-连通的。

证明：

反证法：

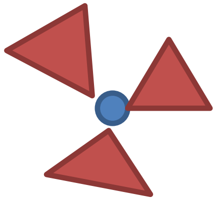
- 如果G是不连通的：
  - 由G是极小不可平面图，你能得出什么结论从而推出矛盾？

  ⇒ G的每个连通分支都是可平面图，你能把G平面嵌入吗？

  ⇒ 可以在某个连通分支的某个面中嵌入其它所有连通分支 ⇒ 得到G的一个平面嵌入 ⇒ 矛盾
- 如果G是连通的但有割点v：
  - 由G是极小不可平面图，你能得出什么结论从而推出矛盾？

  ⇒ G的每个{v}-lobe都是可平面图，你能把G平面嵌入吗？

  ⇒ 它们各自有一个平面嵌入使得v在外部面上 ⇒ 这些平面图在v处粘合形成G的平面嵌入 ⇒ 矛盾



Kuratowski定理 (续)

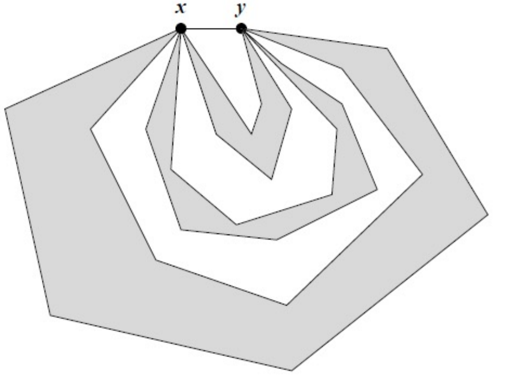
- 引理2：对于不可平面图G的2-点割集S={x, y}，G有一个S-lobe+(x, y)是不可平面图。

证明：

反证法：

- 如果G的每个S-lobe+(x, y)都是可平面图，你能把G平面嵌入吗？

⇒ 它们各自有一个平面嵌入使得(x, y)在外部面上 ⇒ 可以在某个S-lobe+(x, y)的平面嵌入的一个以(x, y)作为边界的面中嵌入其它所有S-lobe+(x, y) ⇒ 如果(x, y)不在G中就去掉 ⇒ 得到G的一个平面嵌入 ⇒ 矛盾



http://www.cs.rpi.edu/~goldberg/13-GT/18-kurat.pdf

Kuratowski定理 (续)

- 2. 不包含Kuratowski子图的3-连通图一定是可平面图。

证明：

为了证明的便利，改为证明一个更强的结论：不包含Kuratowski子图的3-连通图一定有一个无三点共线的凸嵌入（每个面的边界都是凸多边形的平面嵌入）。

1. （引理3）包含不少于5个顶点的3-连通图收缩某条边后仍然是3-连通的。
2. （引理4）不包含Kuratowski子图的图收缩一条边后仍然不包含Kuratowski子图。
3. 数学归纳法（略）。

Kazimierz Kuratowski, 波兰, 1896--1980

他率先提出用集合{{x},{x,y}}来定义有序对<x,y>

http://upload.wikimedia.org/wikipedia/commons/1/18/Kazimierz_Kuratowski.jpg

Wagner定理

- 可平面图G的充要条件：没有可以收缩到K_5或$K_{3,3}$的子图。

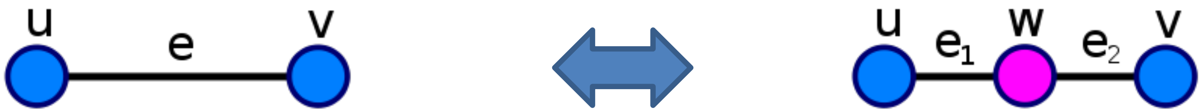
证明：

$\Rightarrow$

- 收缩边不改变图的可平面性 $\Rightarrow$ G的子图收缩边后不可能变成不可平面的K_5或$K_{3,3}$

$\Leftarrow$

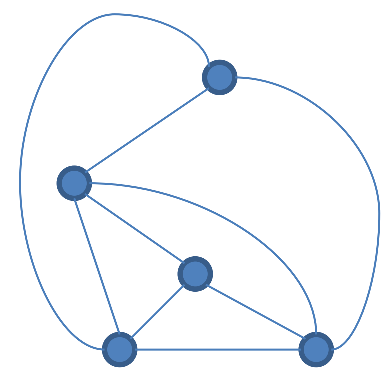
- G的子图不能通过收缩边变成K_5或$K_{3,3}$ $\Rightarrow$ G没有Kuratowski子图 $\Rightarrow$ G是可平面图



可平面性的判断算法

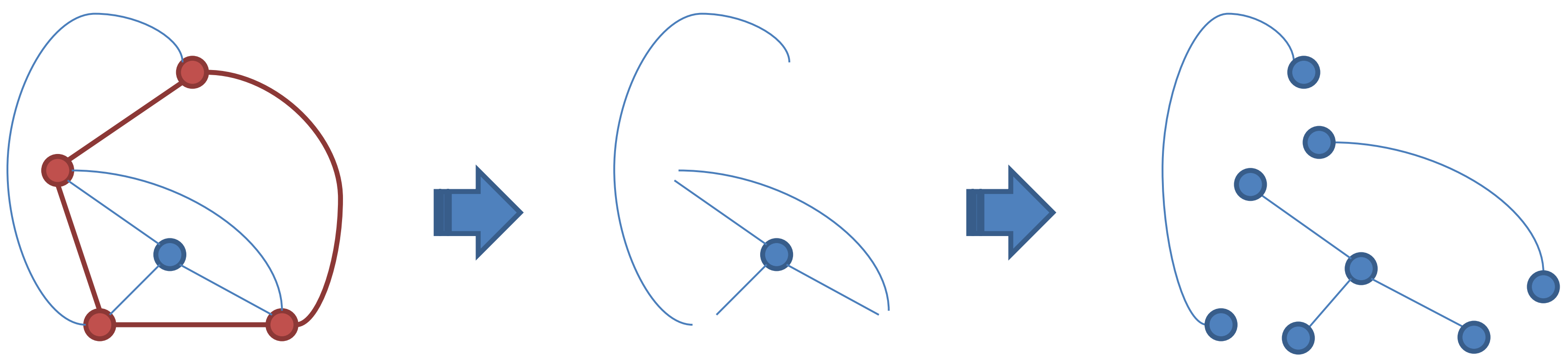
- **D**emoucron-**M**algrange-**P**ertuiset算法：简单的平方算法
- 此外，还有一些较为复杂的线性算法

换作是你，会如何尝试将一个图嵌入到平面中？



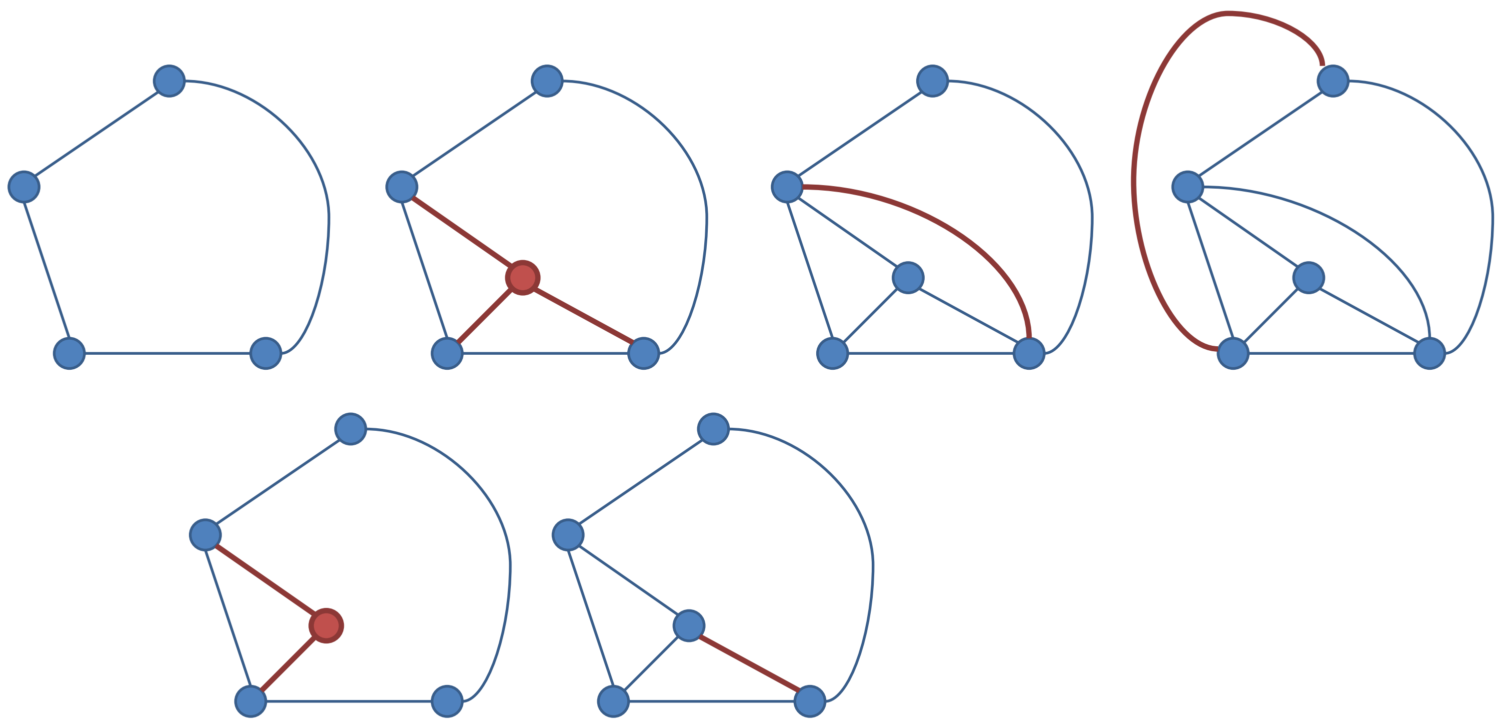
H-fragment*

- 图G的H-fragment
  - 给定G的一个子图H
  - G的H-fragment是G中去掉H后剩余的“连通分支”，即
    - 一条不在H中但两个端点都在H中的边，或者
    - G-V[H]的一个连通分支 + 它连到H的边及其端点
- H及所有H-fragment构成了对G的一种分解



DMP算法*

- 基本思路
  - 迭代地嵌入当前子图的fragment，直至：
    - G全部被嵌入（可平面）
    - 某个fragment无法嵌入（不可平面）
  - 嵌入一个fragment可能难以操作，但总能嵌入其中的一条路



DMP算法 (续)

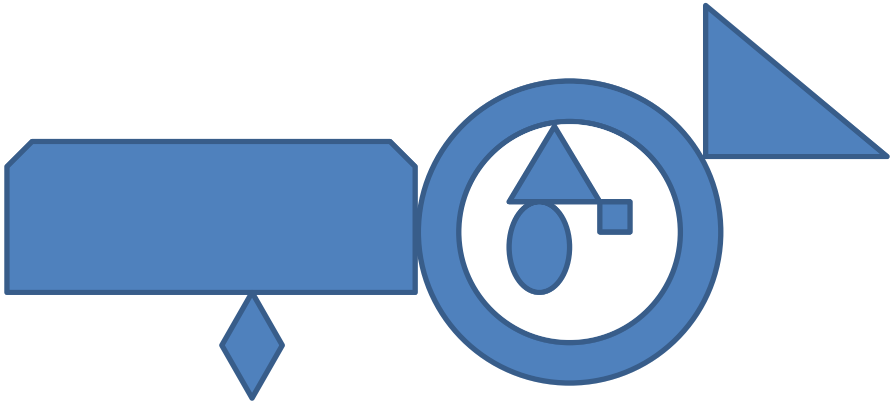
- 图G是可平面的当且仅当G的每个块都是可平面的。

证明：

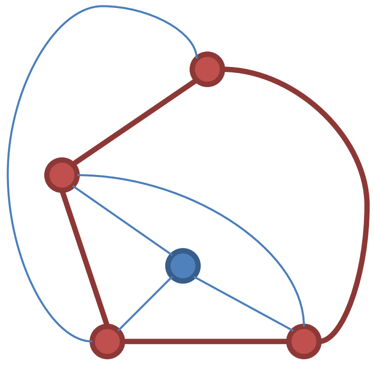
⇒：显然。

⇐：

1. 只需考虑连通图；否则分别考虑每个连通分支即可。
2. 考虑块-割点图：不可能存在“圈” ⇒ 构成“树”
   你能把G平面嵌入吗？
3. 任取一个块作为“树”的根，平面嵌入。
4. 两个块最多只有一个公共顶点（割点） ⇒ 剩余块根据到根的距离由近到远，依次平面嵌入（新块平面嵌入并使割点在外平面上，在割点处与旧块相粘）。



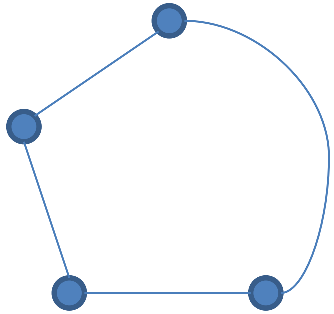
DMP算法 (续)

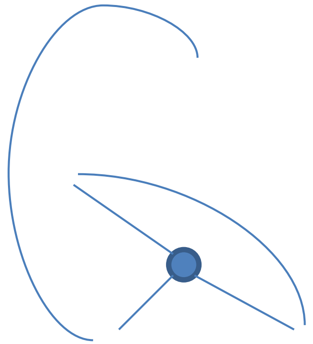


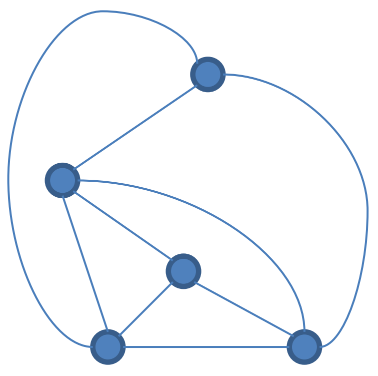
0. 只需检测每个块（2-连通图）是否可平面即可。
1. 从2-连通图G中任取一个圈G_0，平面嵌入。
2. 迭代：
   1. 找到所有G_i-fragment。
   2. 对于每个G_i-fragment（称作B），在G_i中找到所有包含所有B的附着点的面，称作F(B)。
      - 如果某个F(B)为空，则G不可平面。
      - 否则，如果某个|F(B)|=1，则选中这个B。
      - 否则，每个|F(B)|>1，则任选一个B。
   3. 从选中的B中任选一条连接两个附着点的路P，将P平面嵌入到F(B)中的一个面中。（有没有可能B只有一个附着点？）
   4. 将结果记作G_{i+1}。
   5. 如果G_{i+1}=G，则G可平面。否则，继续迭代。

DMP算法举例

- 一个可平面图的例子

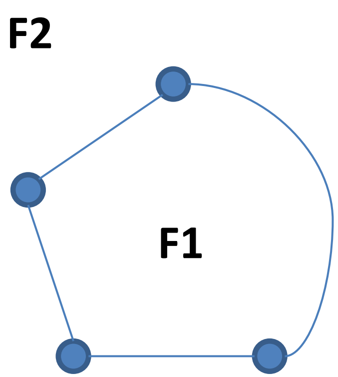


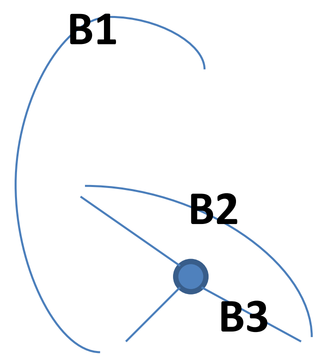


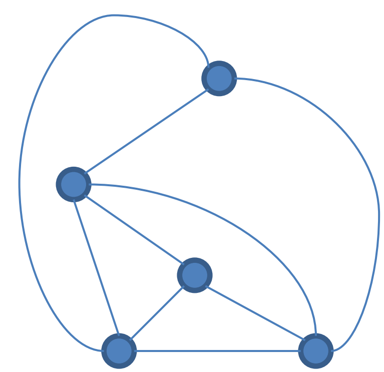


DMP算法举例 (续)

F(B1)={F1, F2}
F(B2)={F1, F2}
F(B3)={F1, F2}

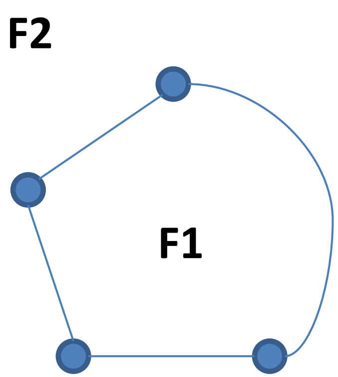


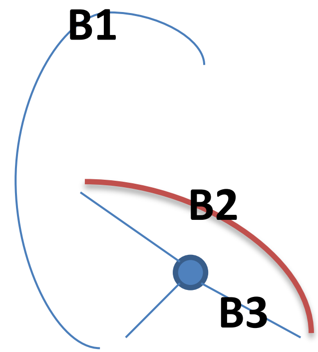


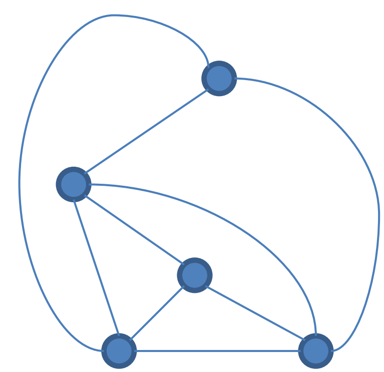


DMP算法举例 (续)

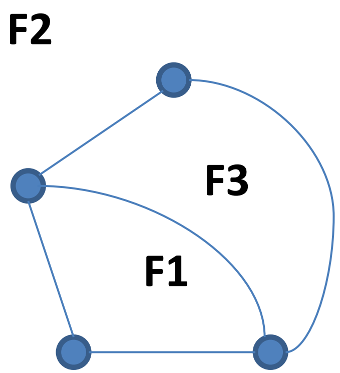
F(B1)={F1, F2}
F(B2)={F1, F2}
F(B3)={F1, F2}

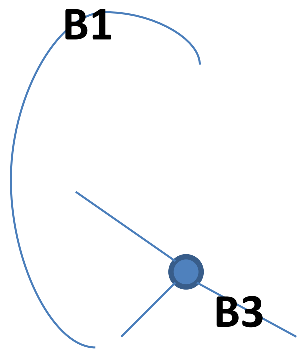


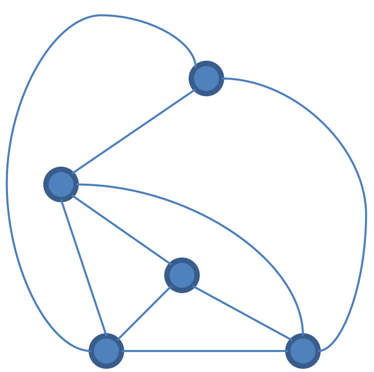




DMP算法举例 (续)

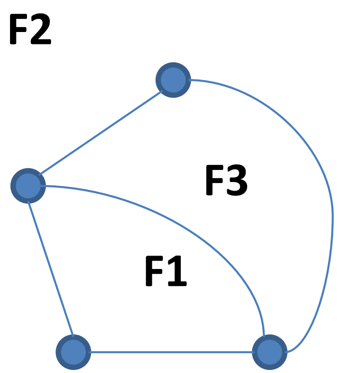


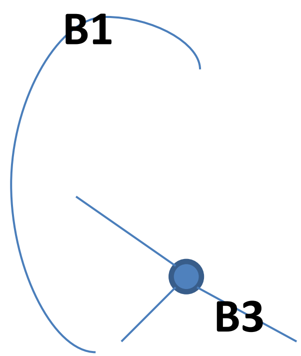


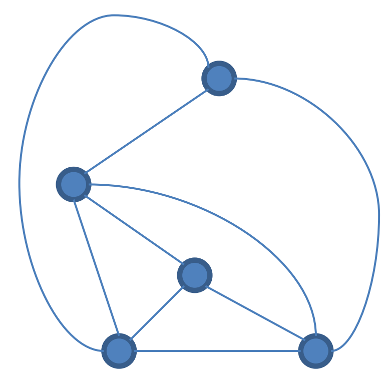


DMP算法举例 (续)

F(B1)={F2}
F(B3)={F1, F2}

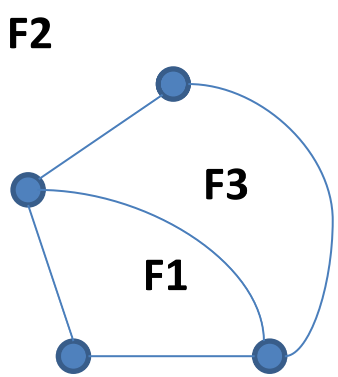


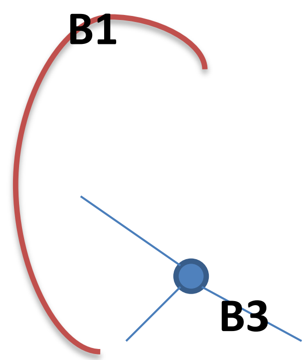


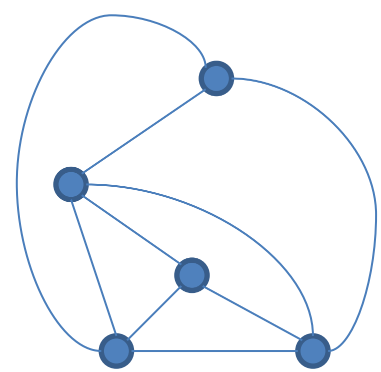


DMP算法举例 (续)

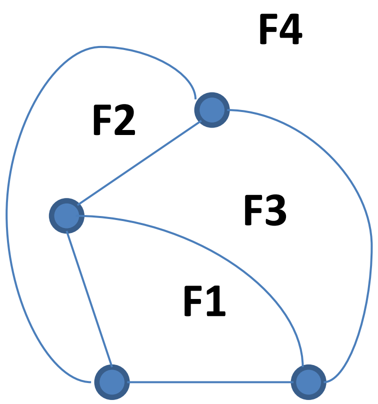
F(B1)={F2}
F(B3)={F1, F2}

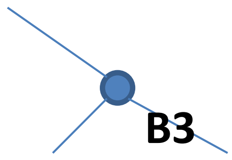


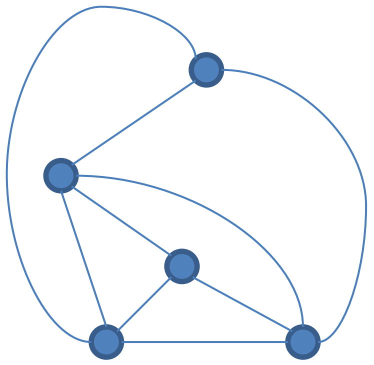




DMP算法举例 (续)

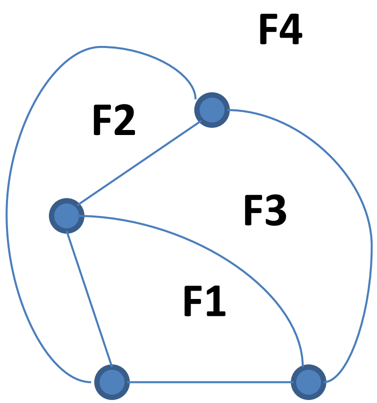


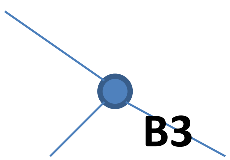


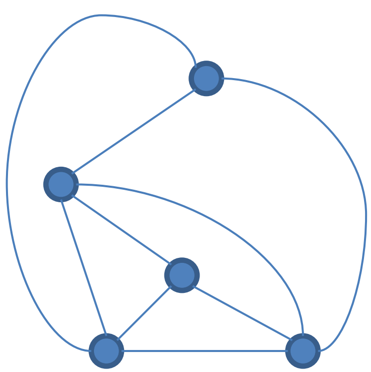


DMP算法举例 (续)

F(B3)={F1}

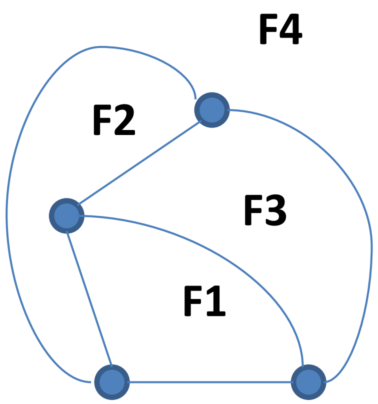




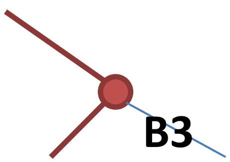


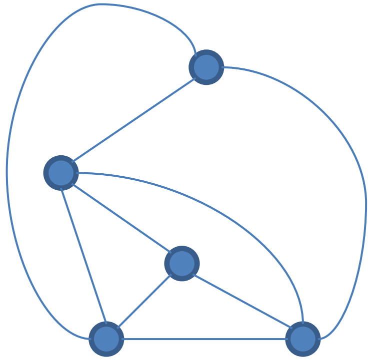
DMP算法举例 (续)

F(B3)={F1}

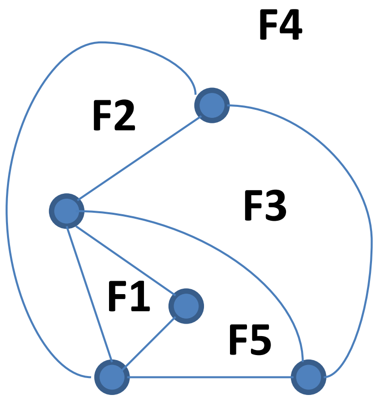


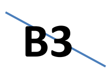
注意：单看这条路径，似乎也可以放在F2中，但不可以！

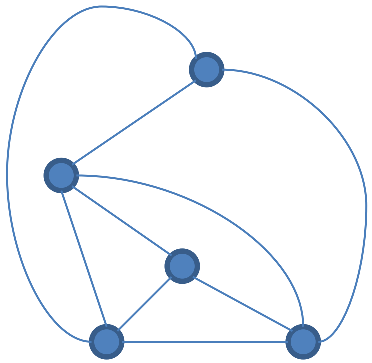




DMP算法举例 (续)

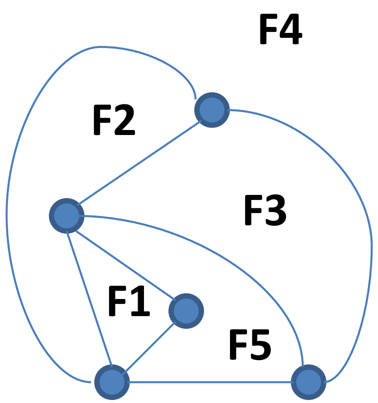


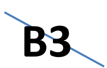


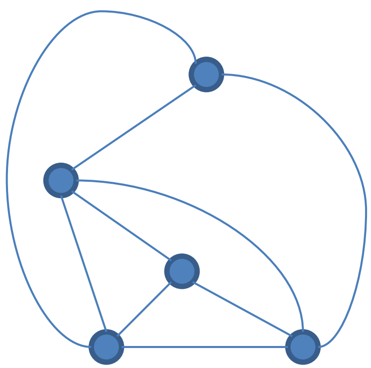


DMP算法举例 (续)

F(B3)={F5}

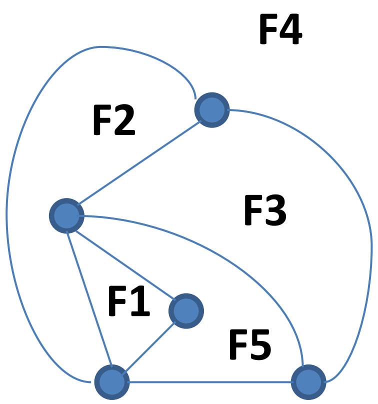


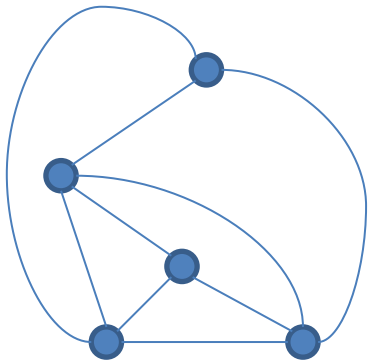




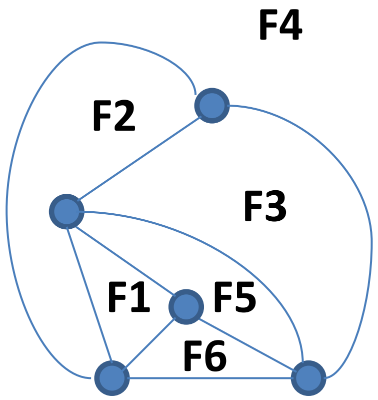
DMP算法举例 (续)

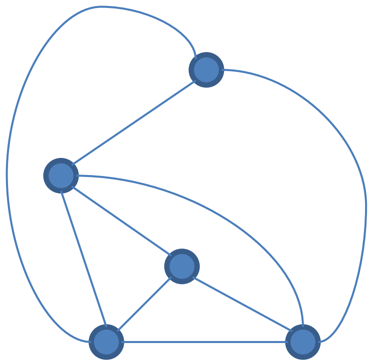
F(B3)={F5}





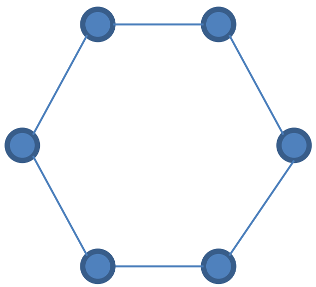
DMP算法举例 (续)

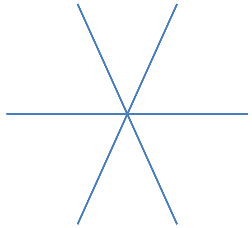


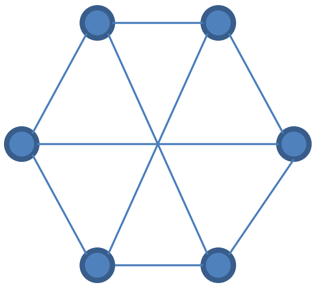


DMP算法举例 (续)

- 一个不可平面图的例子

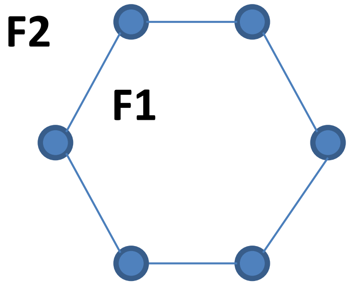


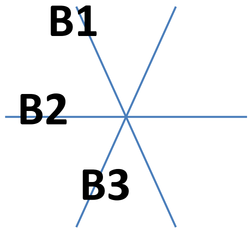


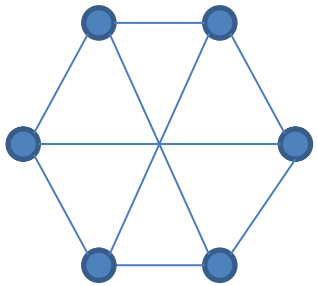


DMP算法举例 (续)

F(B1)={F1, F2}
F(B2)={F1, F2}
F(B3)={F1, F2}

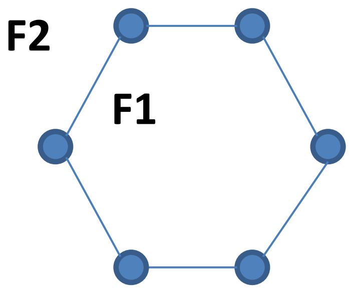


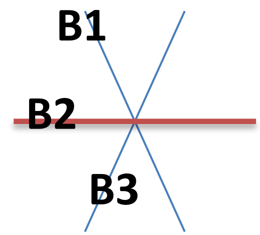


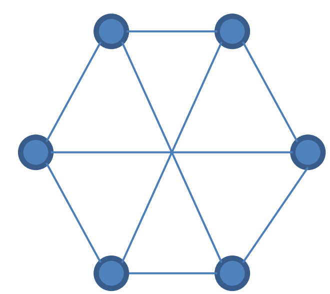


DMP算法举例 (续)

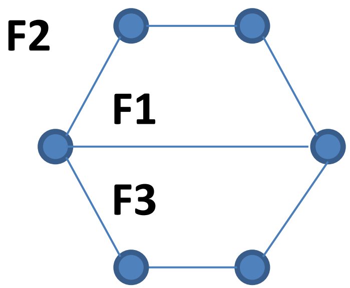
F(B1)={F1, F2}
F(B2)={F1, F2}
F(B3)={F1, F2}

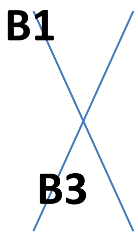


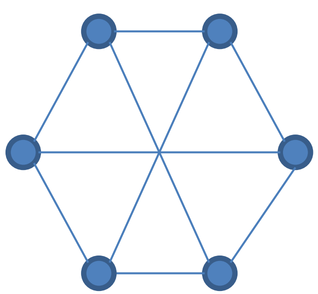




DMP算法举例 (续)

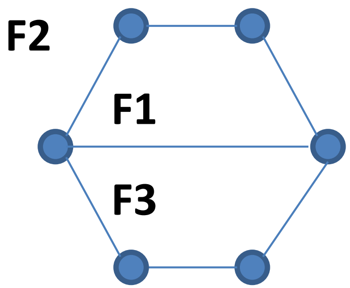


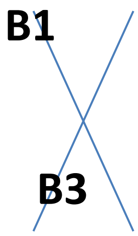


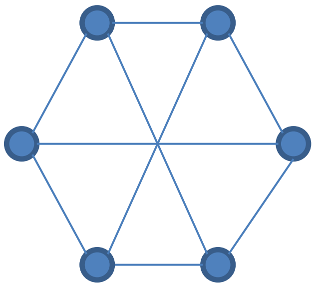


DMP算法举例 (续)

F(B1)={F2}
F(B3)={F2}

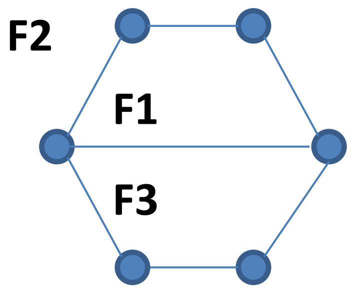


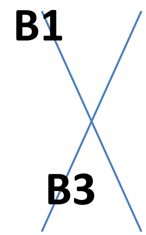


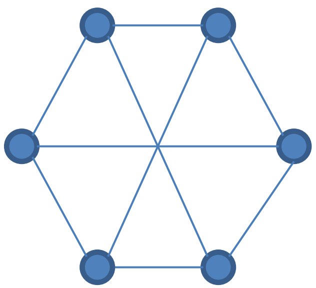


DMP算法举例 (续)

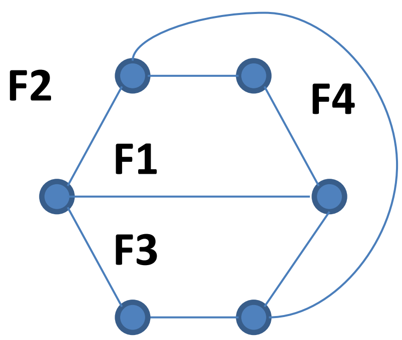
F(B1)={F2}
F(B3)={F2}

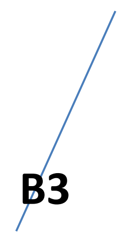


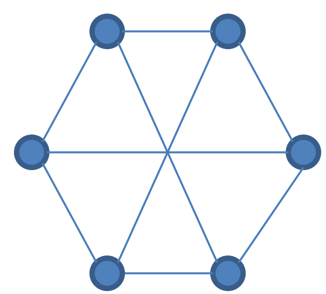




DMP算法举例 (续)

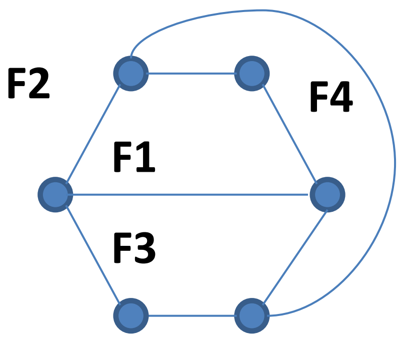


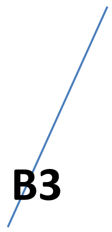


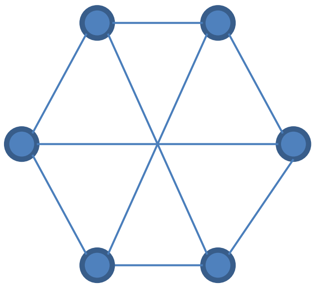


DMP算法举例 (续)

F(B3)=Ø

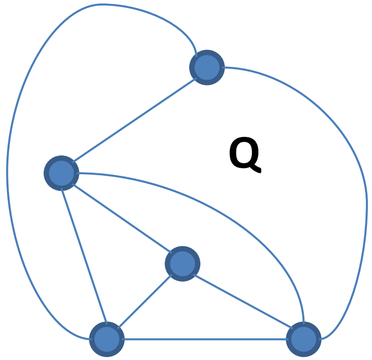






DMP算法的正确性

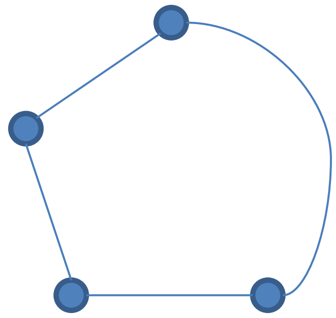
如果G是可平面图，假设其平面嵌入的预期结果是Q。

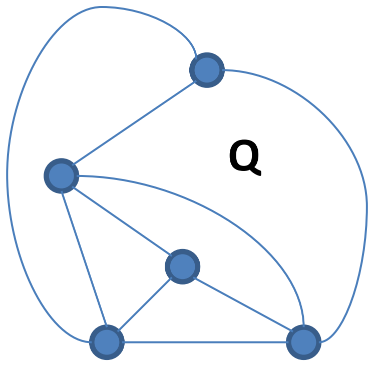


DMP算法的正确性

如果G是可平面图，假设其平面嵌入的预期结果是Q。

1. G中的每个圈在平面嵌入后仍是一个圈 ⇒ 算法先将一个圈平面嵌入：或者与Q中一致，或者方向相反（怎么办？）
   - 并不影响，因为Q可以翻转，因此结果与Q“对称”

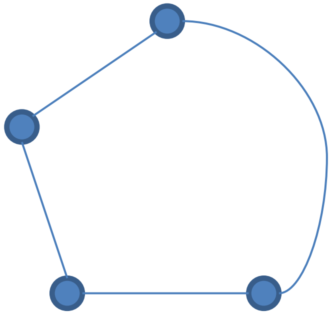


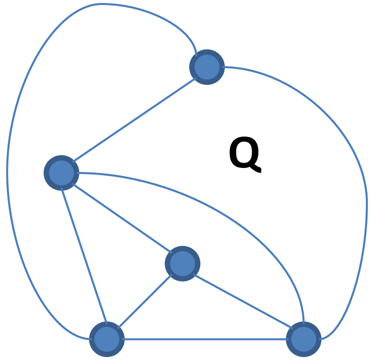


DMP算法的正确性

如果G是可平面图，假设其平面嵌入的预期结果是Q。

1. G中的每个圈在平面嵌入后仍是一个圈 ⇒ 算法先将一个圈平面嵌入：或者与Q中一致，或者方向相反（怎么办？）
   - 并不影响，因为Q可以翻转，因此结果与Q“对称”
2. 只需证明：如果算法截至G_i的嵌入与Q一致或者（局部）对称，那么截至G_{i+1}的嵌入也与Q一致或者（局部）对称。

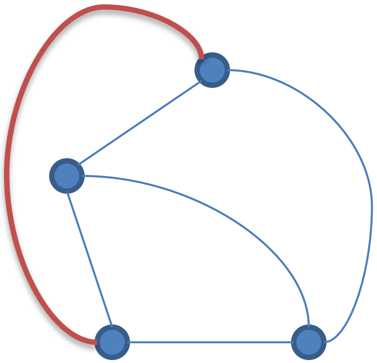


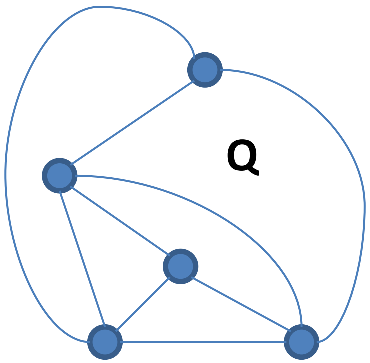


DMP算法的正确性

如果G是可平面图，假设其平面嵌入的预期结果是Q。

1. G中的每个圈在平面嵌入后仍是一个圈 ⇒ 算法先将一个圈平面嵌入：或者与Q中一致，或者方向相反（怎么办？）
   – 并不影响，因为Q可以翻转，因此结果与Q“对称”
2. 只需证明：如果算法截至G_i的嵌入与Q一致或者（局部）对称，那么截至G_{i+1}的嵌入也与Q一致或者（局部）对称。
   – 如果选中的是某个|F(B)|=1，即只有一个面可以将P嵌入，那么这种嵌入方式必然与Q中的方式一致。

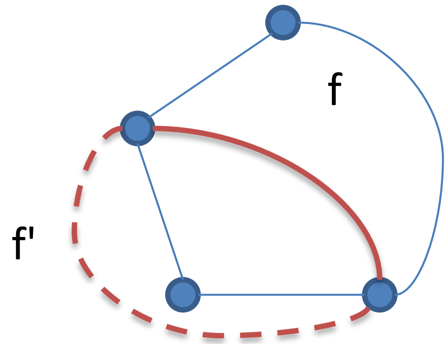


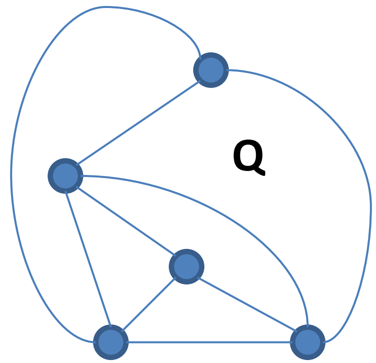


DMP算法的正确性

如果G是可平面图，假设其平面嵌入的预期结果是Q。

1. G中的每个圈在平面嵌入后仍是一个圈 ⇒ 算法先将一个圈平面嵌入：或者与Q中一致，或者方向相反（怎么办？）
   – 并不影响，因为Q可以翻转，因此结果与Q“对称”
2. 只需证明：如果算法截至G_i的嵌入与Q一致或者（局部）对称，那么截至G_{i+1}的嵌入也与Q一致或者（局部）对称。
   – 如果选中的是某个|F(B)|=1，即只有一个面可以将P嵌入，那么这种嵌入方式必然与Q中的方式一致。
   – 如果选中的是某个|F(B)|>1，并且没有将P按照Q中的方式嵌入到面f中，而是“错误地”嵌入到面f′中，怎么办？
      • 这意味着f和f′有公共边界，那么对于之后的fragment，只要将原本嵌入f的改为嵌入f′，将原本嵌入f′的改为嵌入f，即沿f和f′的公共边界对称翻转，便可得到一个与Q（局部）对称的结果。

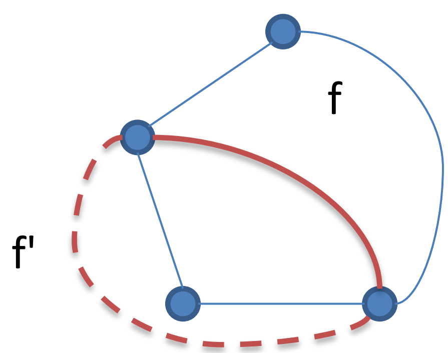


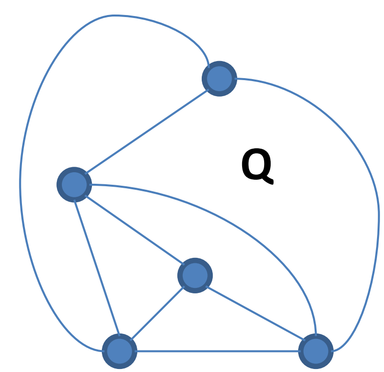


DMP算法的正确性

如果G是可平面图，假设其平面嵌入的预期结果是Q。

1. G中的每个圈在平面嵌入后仍是一个圈 ⇒ 算法先将一个圈平面嵌入：或者与Q中一致，或者方向相反（怎么办？）
   - 并不影响，因为Q可以翻转，因此结果与Q“对称”
2. 只需证明：如果算法截至G_i的嵌入与Q一致或者（局部）对称，那么截至G_{i+1}的嵌入也与Q一致或者（局部）对称。
   - 如果选中的是某个|F(B)|=1，即只有一个面可以将P嵌入，那么这种嵌入方式必然与Q中的方式一致。
   - 如果选中的是某个|F(B)|>1，并且没有将P按照Q中的方式嵌入到面f中，而是“错误地”嵌入到面f′中，怎么办？
     - 这意味着f和f′有公共边界，那么对于之后的fragment，只要将原本嵌入f的改为嵌入f′，将原本嵌入f′的改为嵌入f，即沿f和f′的公共边界对称翻转，便可得到一个与Q（局部）对称的结果。





DMP算法的运行时间

- $O(v^2)$
  - 块分解：$O(v)$，基于DFS
  - 找初始的圈：$O(v)$
  - 迭代轮数？
    - 简单平面图满足$\varepsilon \leq 3v-6$
    - $\varphi-1=\varepsilon-v+1\leq 2v-5\in O(v)$，因为每轮迭代新增一个面
  - 每轮迭代的时间：$O(v)$

作业

- 请用DMP算法将P214的G_2嵌入到平面中，要写出详细步骤。