Advertisement for the circuit theory of 4-regular graphs

The following ideas appeared in the work of Andre Bouchet and François Jaeger in the 1980s, though I'm not sure either of them ever stated the concluding theorem.

Definition 1. A 4-regular graph consists of vertices, edges and free loops. Free loops are not incident on anything else. A loop is incident on one vertex, and a non-loop edge is incident on two different vertices. Each vertex is incident on four non-loop edges, or a loop and two non-loop edges, or two loops.

Definition 2. Let *F* be a 4-regular graph. Then a *circuit* of *F* is either a free loop or a sequence $v_1,e_1,v_2,\ldots,v_k,e_k,v_1$ of vertices and edges of *F*, such that (a) every edge in the sequence is incident on the vertices listed before and after it and (b) no edge appears twice.

Notice that a circuit may contain sub-circuits.

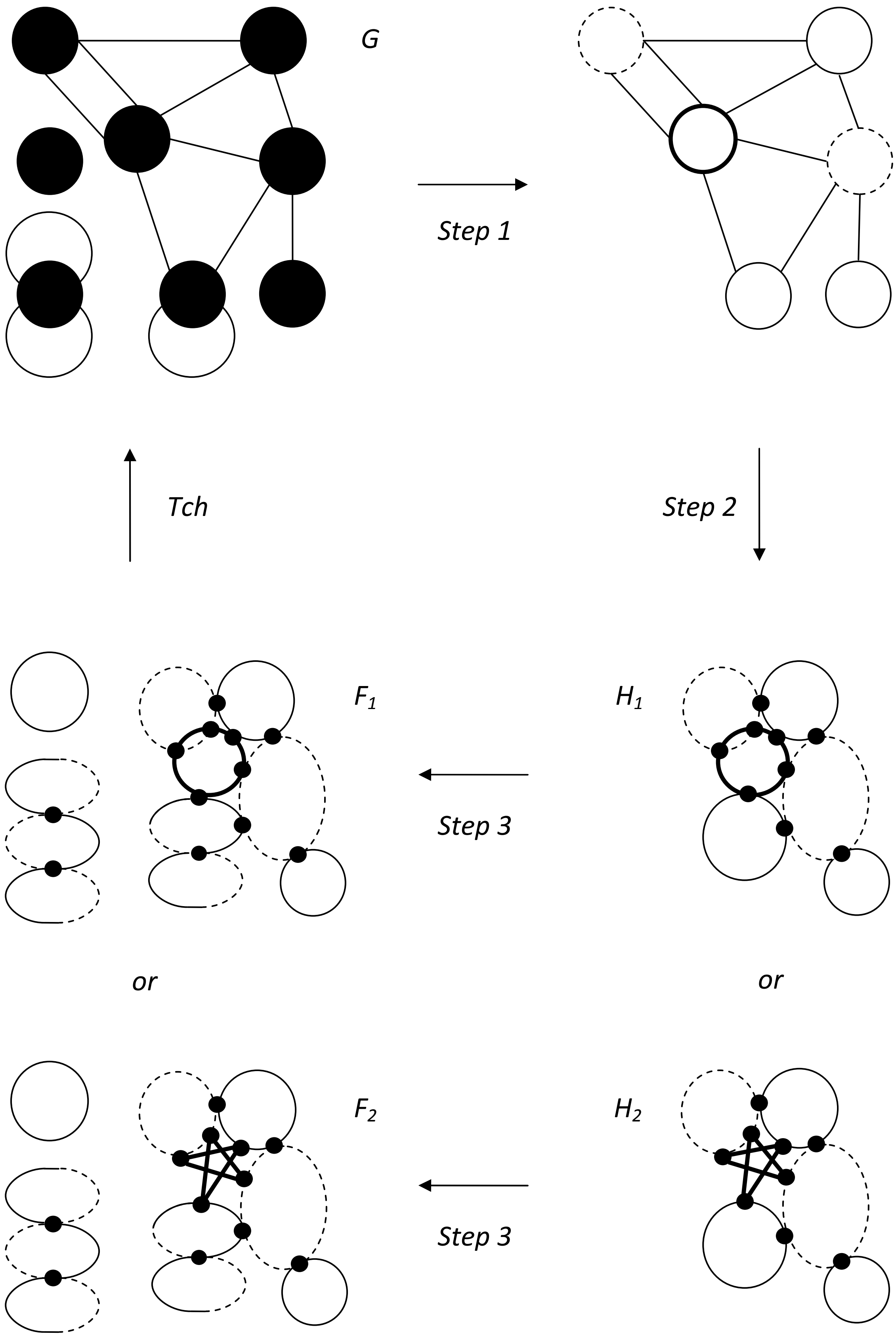
Definition 3. Let *F* be a 4-regular graph. Then a *circuit partition* or *Eulerian partition* of *F* is a set *P* of circuits of *F*, such that every edge of *F* appears in exactly one element of *P*. *P* must include all the free loops of *F*.

Definition 4. Let *F* be a 4-regular graph, with a circuit partition *P*. Then the *touch-graph Tch(P)* has a vertex for each circuit of *P* and an edge for each vertex of *F*. The edge corresponding to *v* is incident on the circuit(s) of *P* incident on *v*.

Theorem. Let *G* be any graph. Then there is a 4-regular graph *F* with a circuit partition *P* such that *Tch(P)* is isomorphic to *G*.

Proof. *F* and *P* are constructed from *G* in three steps. First, remove isolated vertices and loops (but remember they were there). Second, obtain a 4-regular graph *H* by replacing each non-loop edge of *G* with a vertex and replacing each vertex of *G* with a circuit, which includes the vertices of *H* corresponding to edges of *G* incident on that vertex (in some order). Third, obtain *F* by restoring the loops (with figure eights) and the isolated vertices (with free loops). Voilà! You have created *F* and *P*.

An example is given in the figure below. Notice that the construction involves choosing arbitrary edge-orders at each vertex of *G*; as indicated in the figure, these edge-orders can significantly affect the resulting *F*.



G
Step 1
Step 2
Tch
F1
H1
Step 3
or
or
F2
H2
Step 3