Generation without isomorphs

Brendan McKay
Australian National University

and various colleagues to be mentioned...

Exhaustive generation

For some reason, lots of people want to generate exhaustive lists of combinatorial objects.

Tens of thousands of examples are published, involving many fields of science.

Usually, but not always, there is a concept of equivalent objects, and it is desired to obtain only one member of each equivalence class.

We will focus on graphs.

In 1974 it took 6 hours to generate all of the 274,668 graphs on 9 vertices (Baker, Dewdeny, Szilard). Now it takes 0.1 seconds.

It is practical to generate all 50,502,031,367,952 graphs on 13 vertices and plausible to generate all 29,054,155,657,235,488 graphs on 14 vertices.

Recursive generation

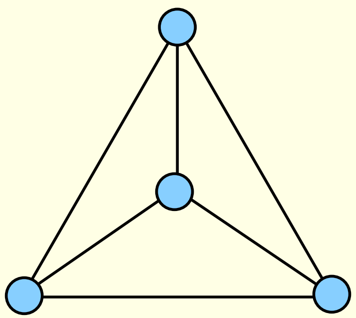
A recursive construction of a class of graphs consists of

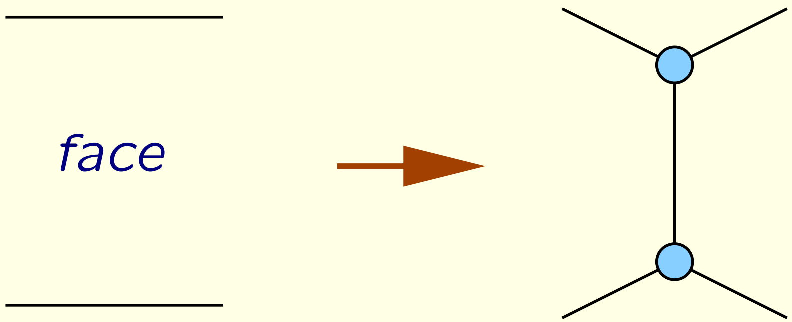
1. a set of irreducible graphs in the class, and
2. a set of expansions that can be performed on graphs in the class,

such that each graph in the class can be constructed from an irreducible graph via a sequence of expansions while staying within the class.

The reverse of an expansion is a reduction.

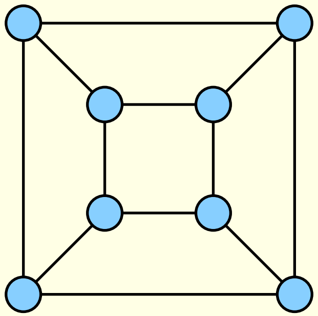
Example: 3-connected planar cubic graphs

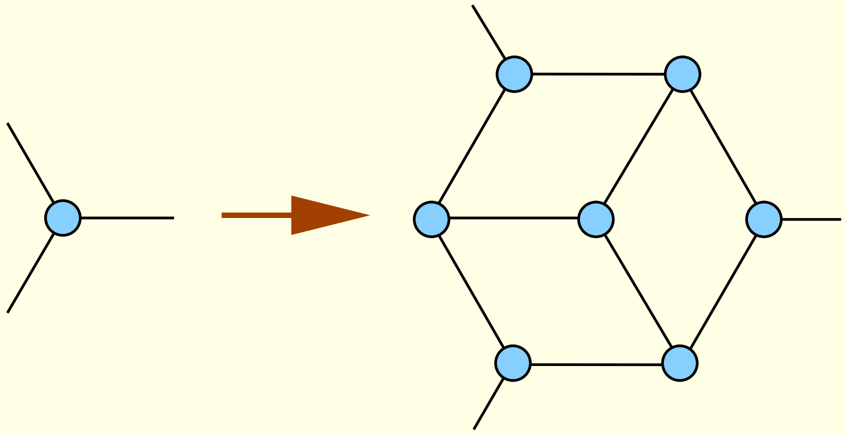


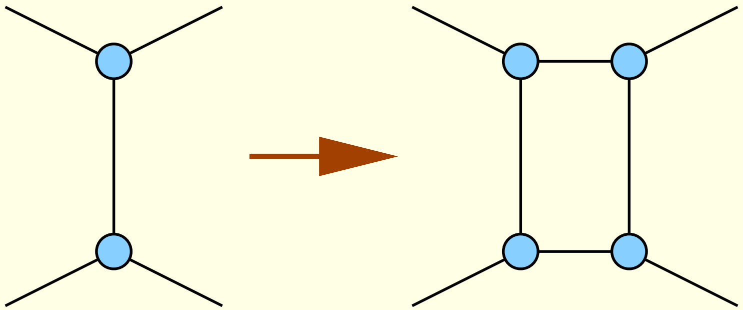


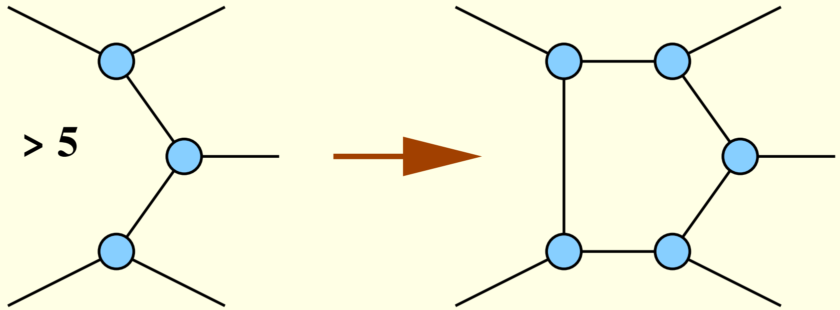
Example: 3-connected planar cubic graphs without triangles

The following generation method is a slight improvement on one discovered by Batagelj (1989).

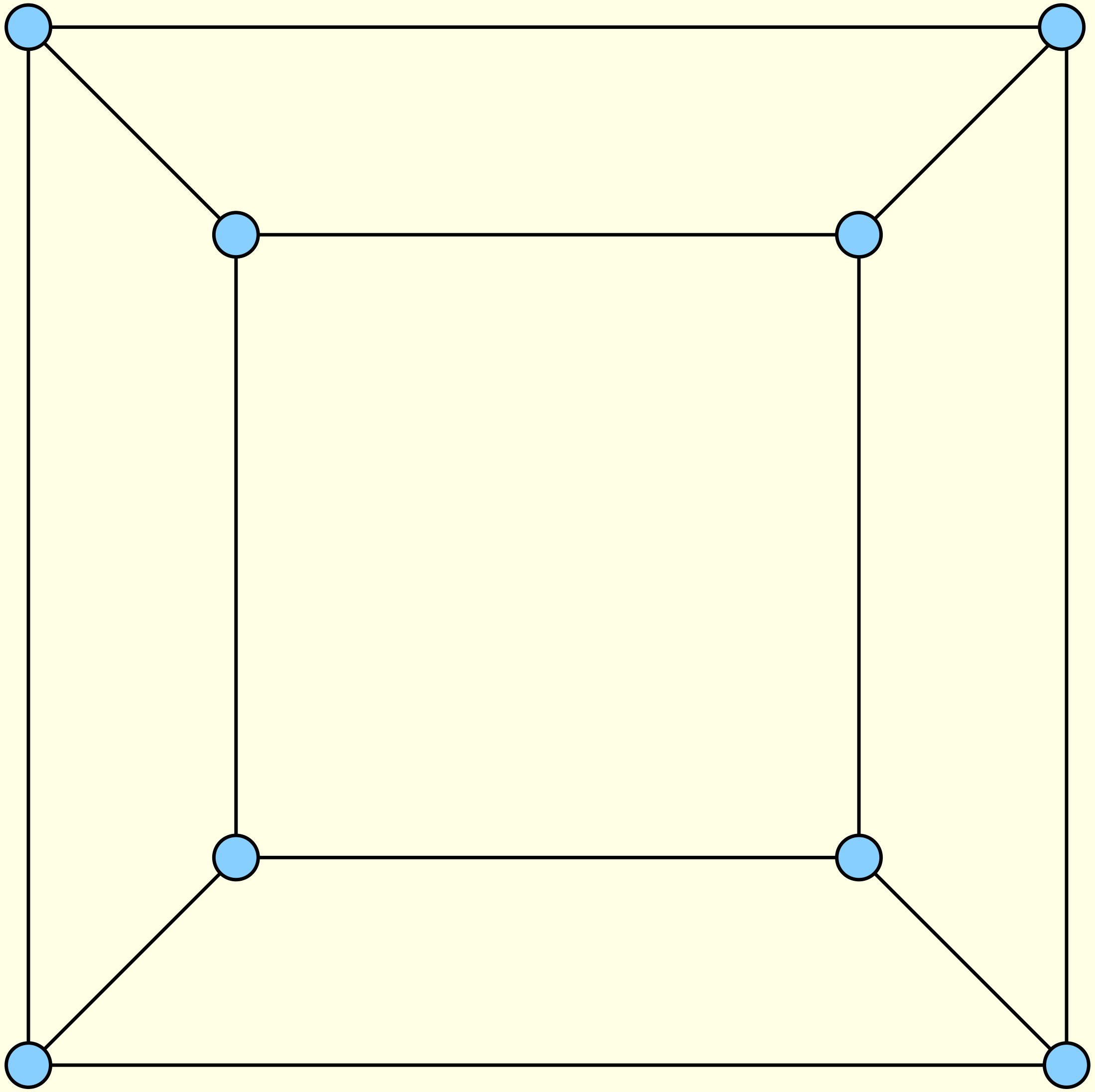




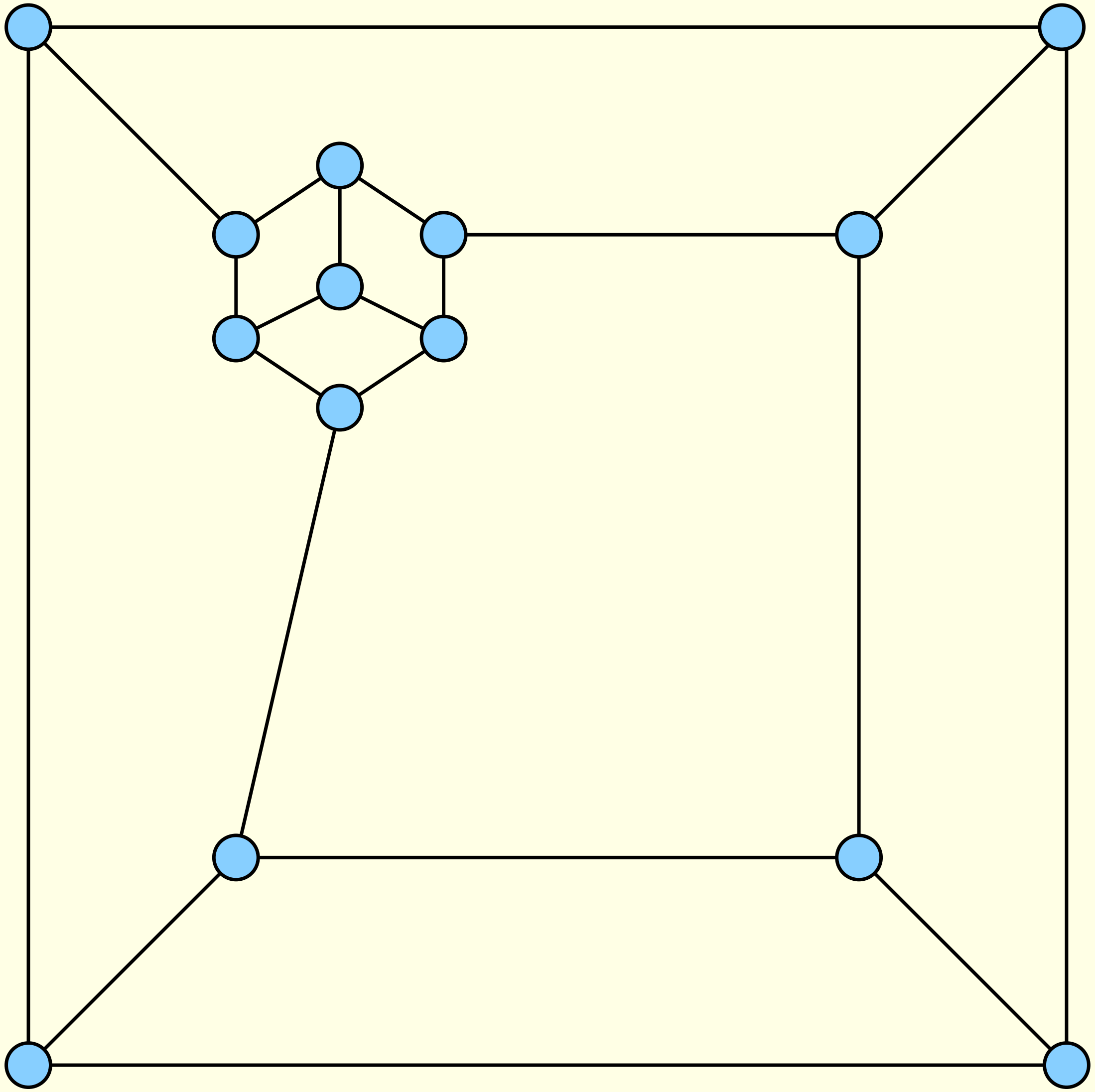




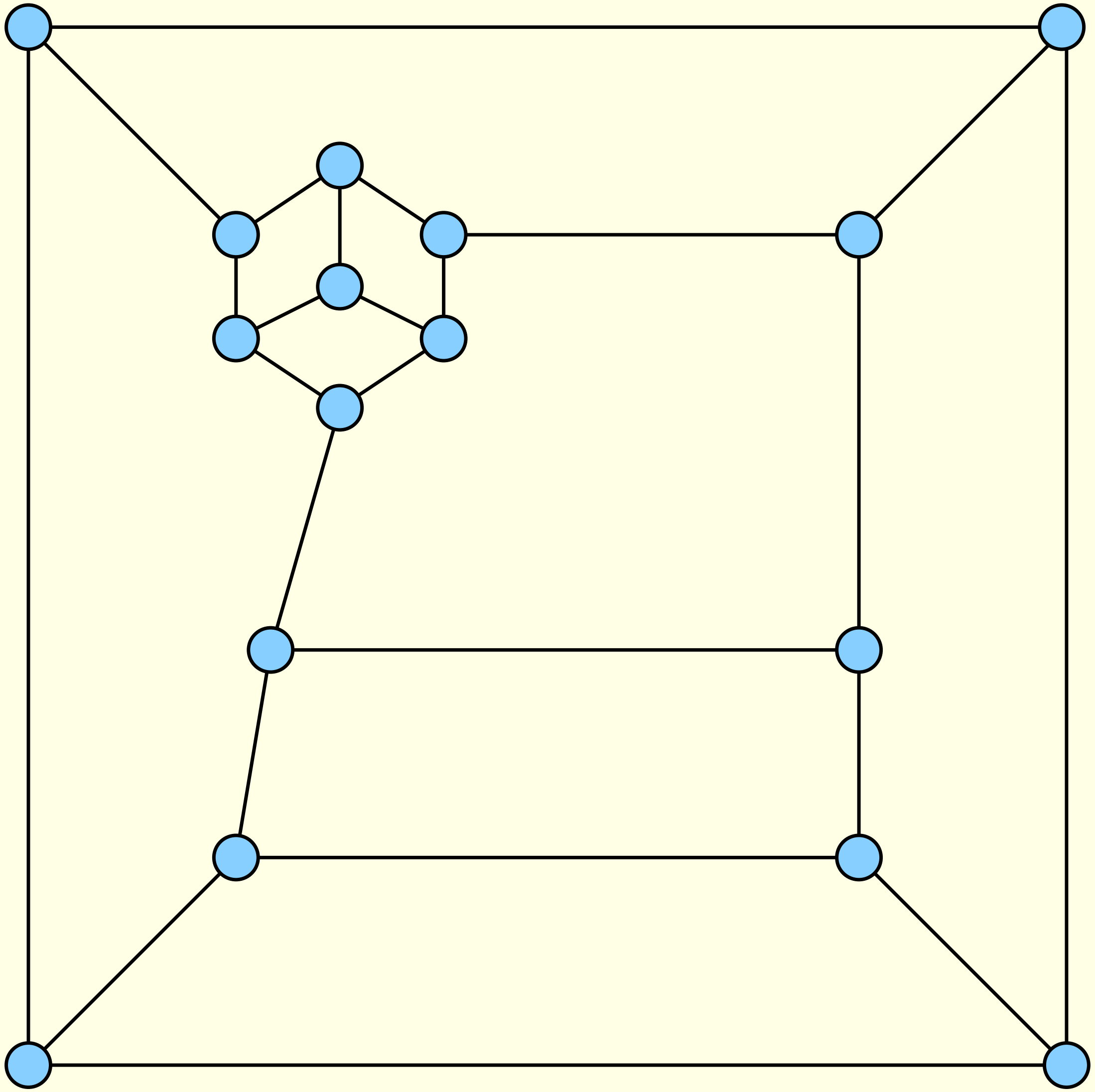
For example:



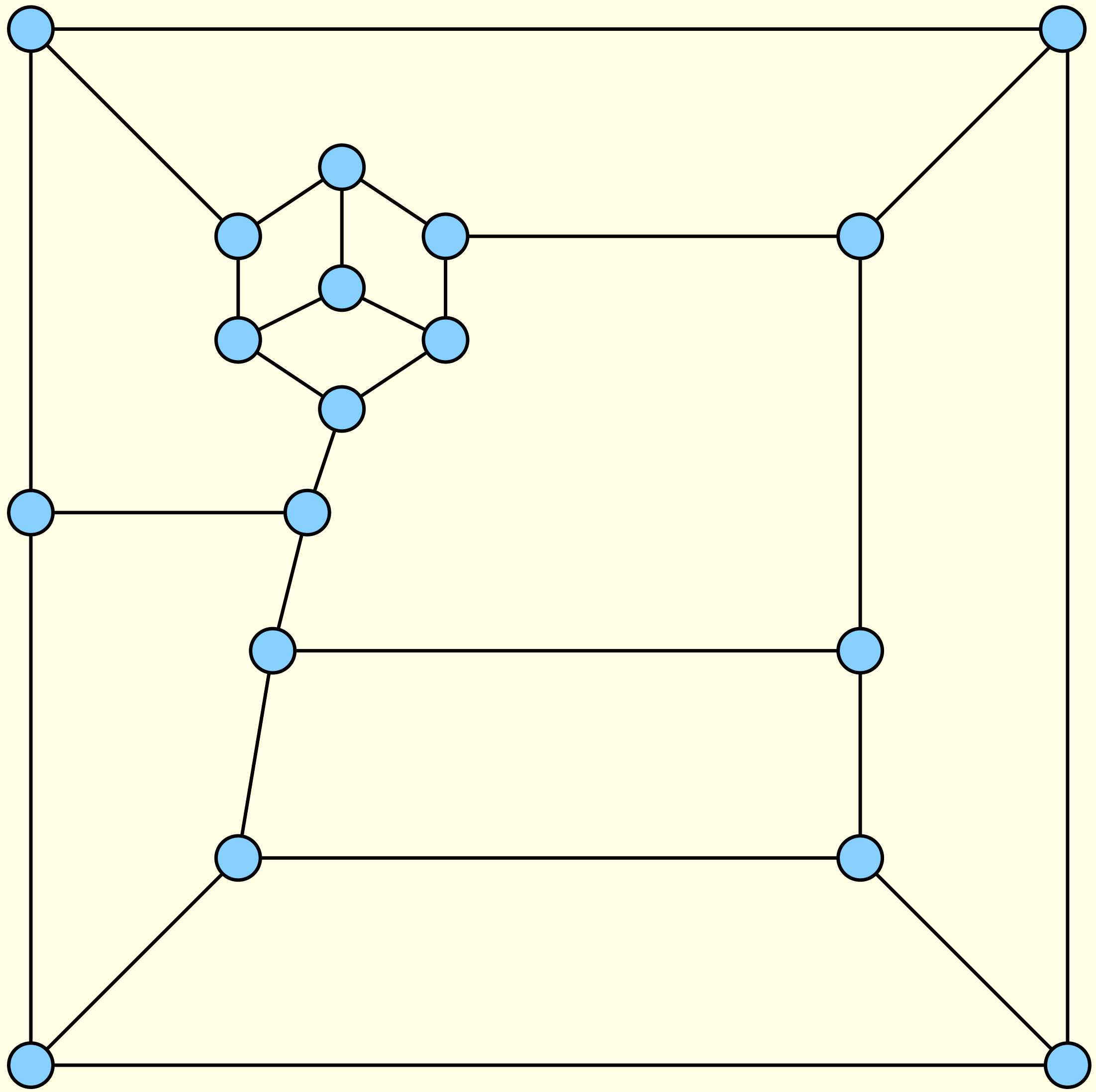
For example:



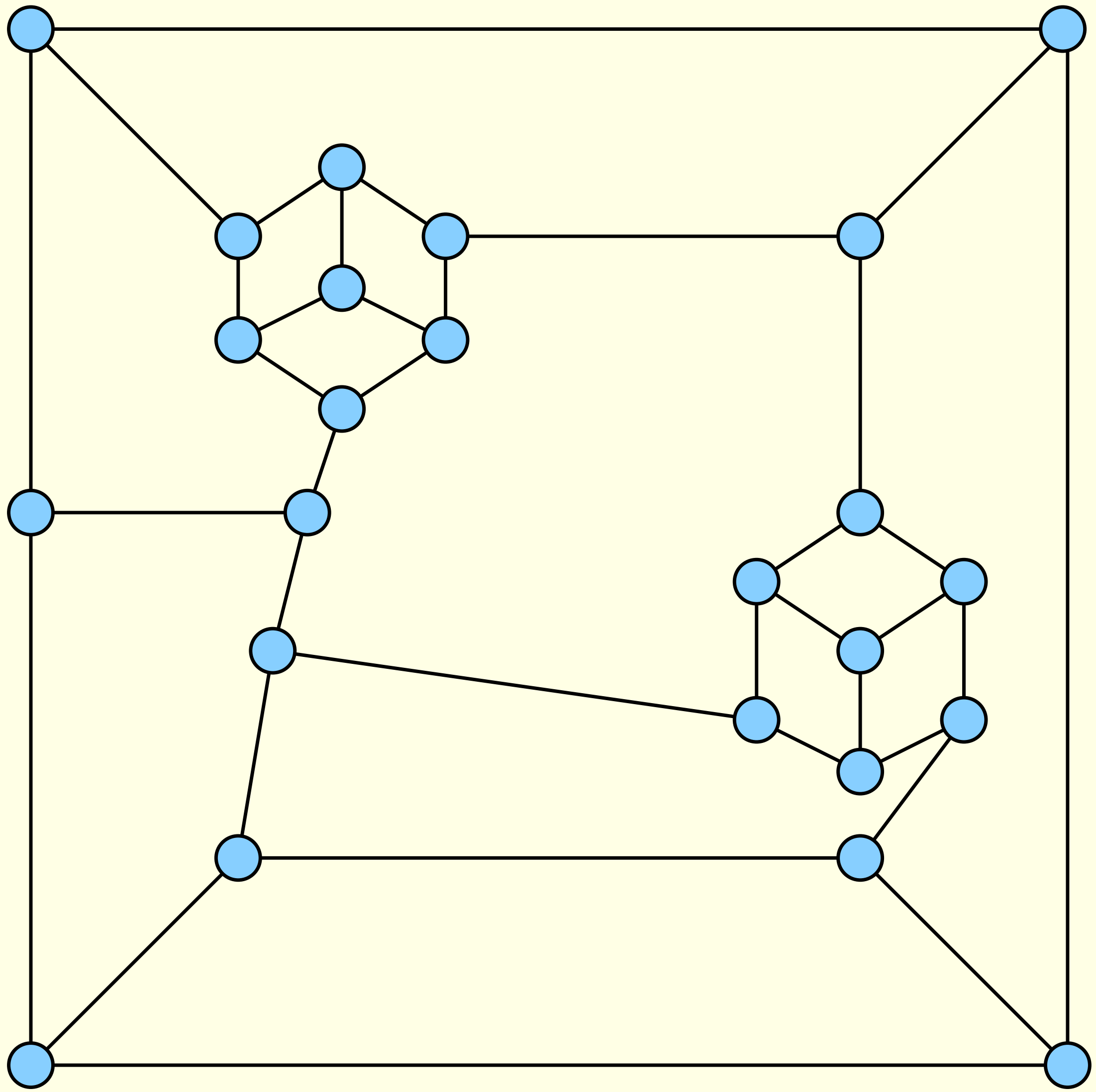
For example:



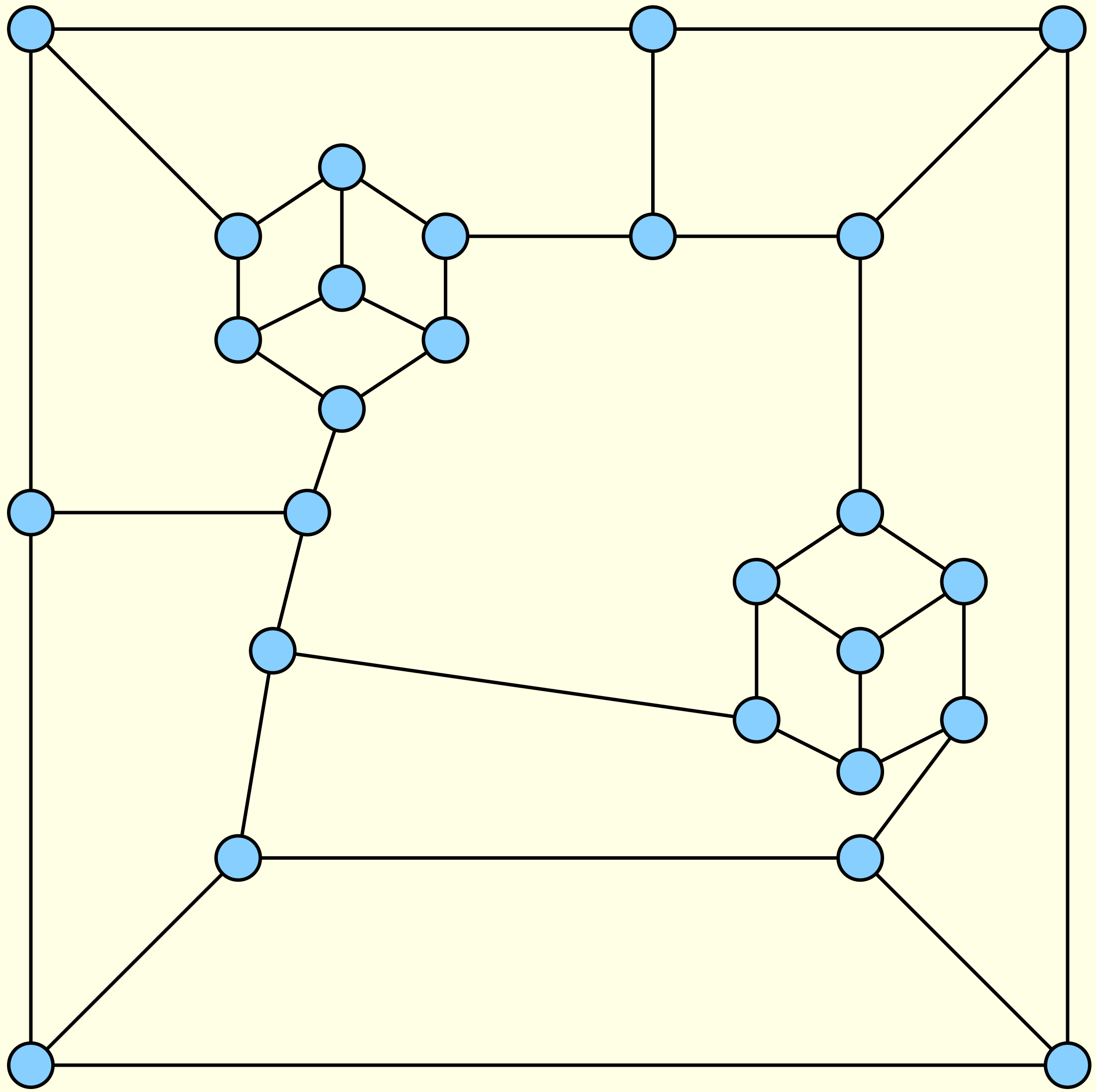
For example:



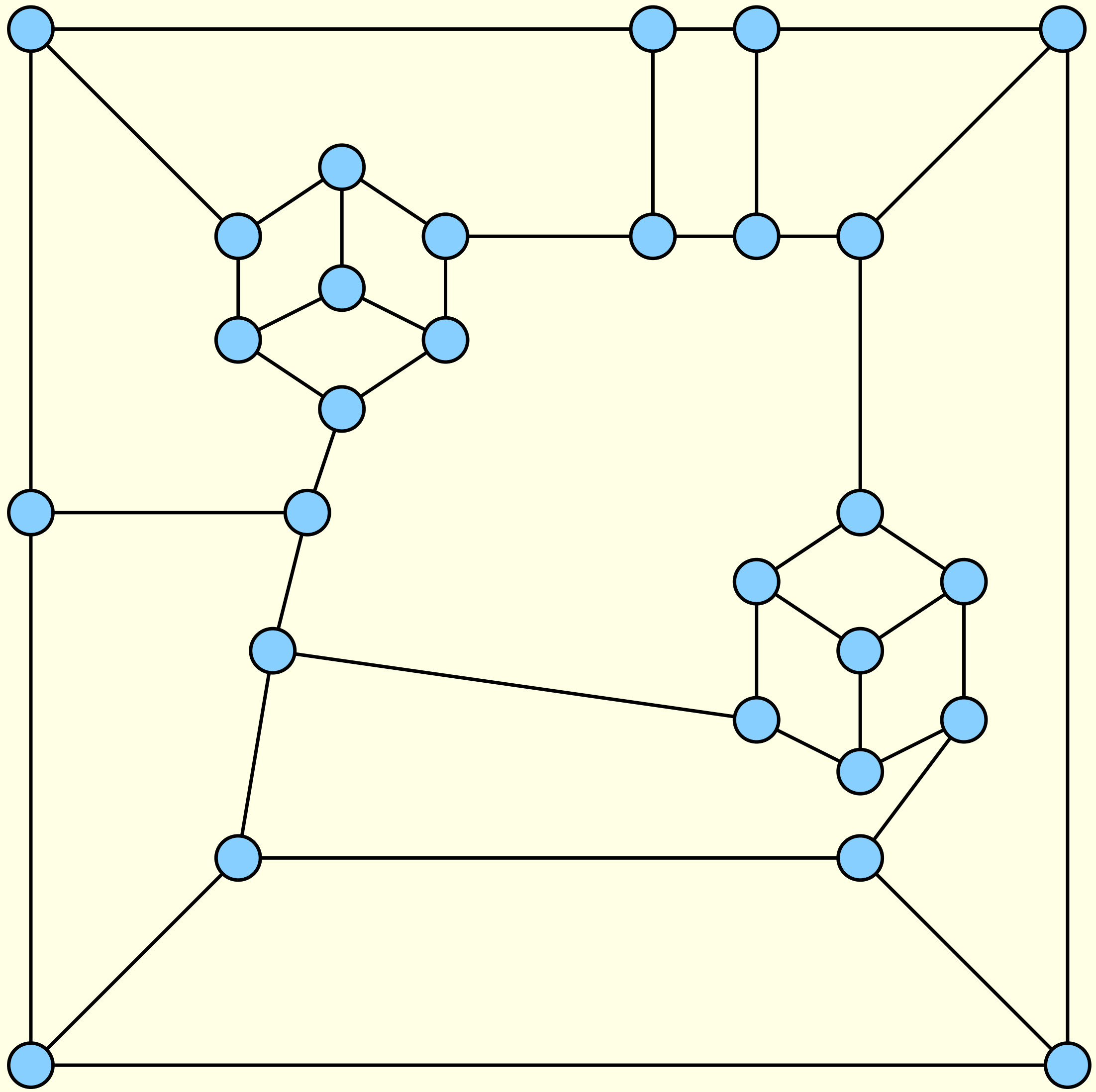
For example:



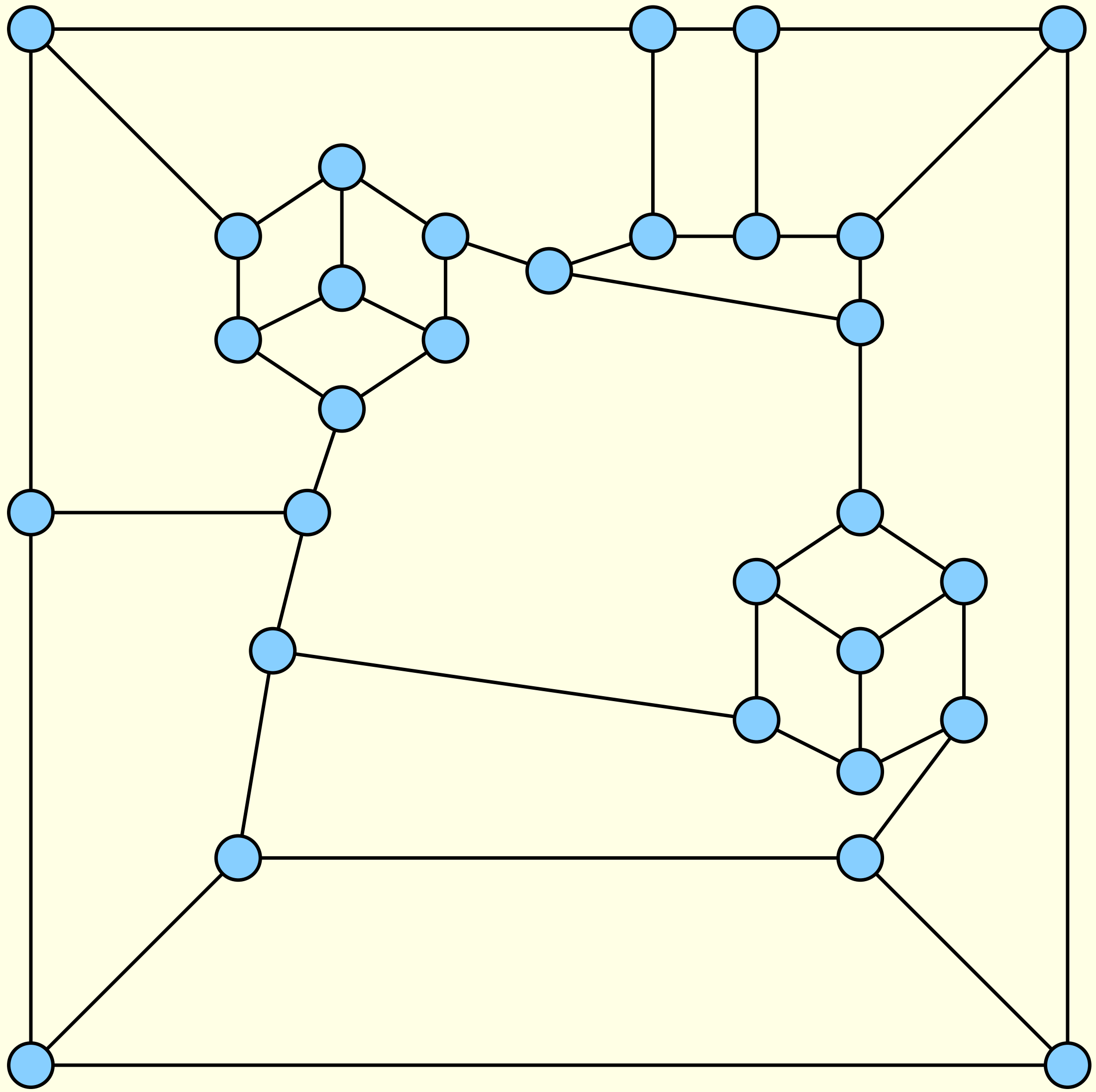
For example:



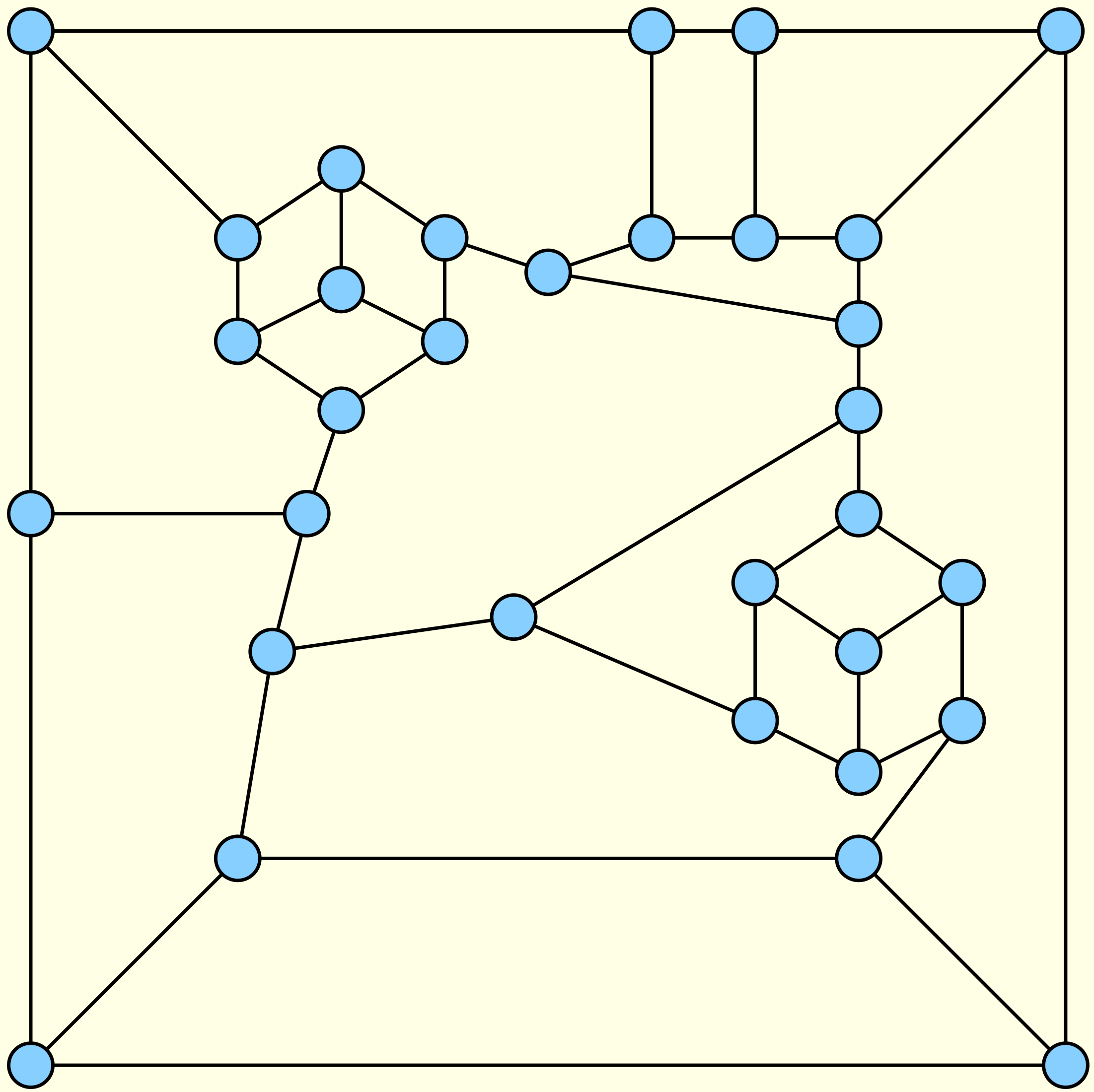
For example:



For example:

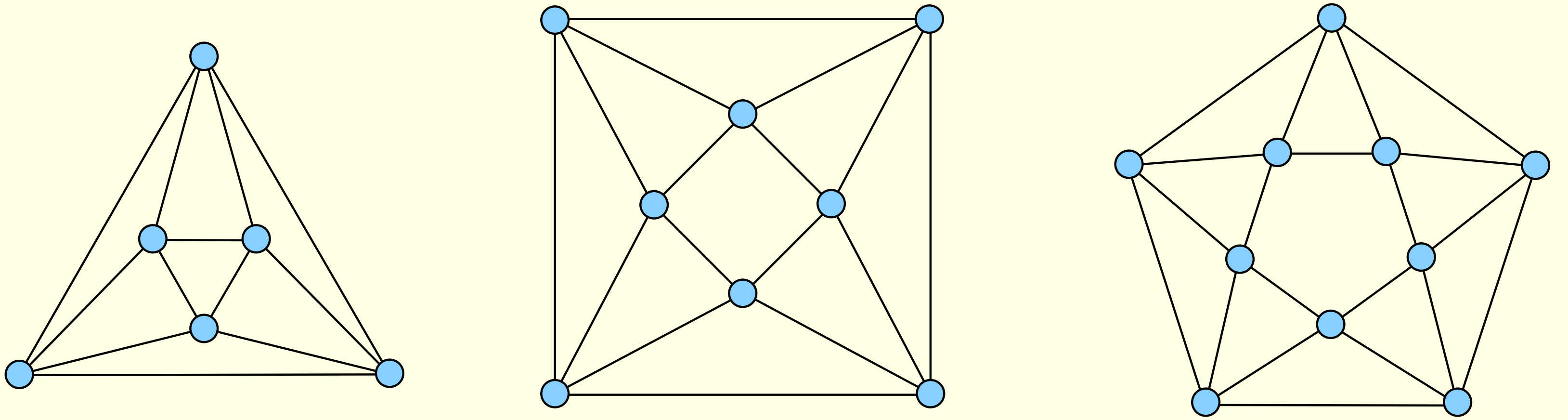


For example:

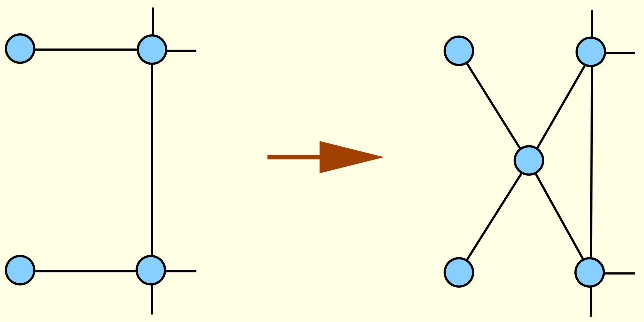


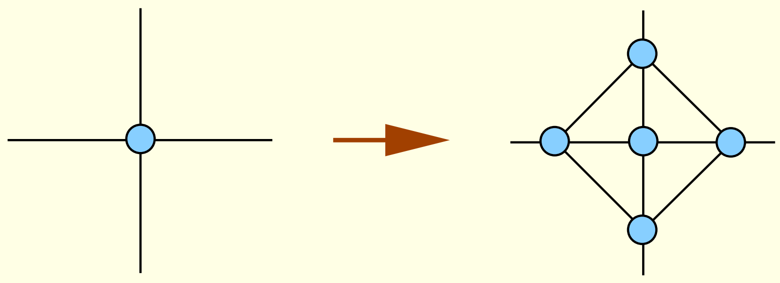
Example: 3-connected planar quartic graphs

Idea of Batagelj proved by Brinkmann, Greenberg, Greenhill, McKay and Thomas (2005).



etc

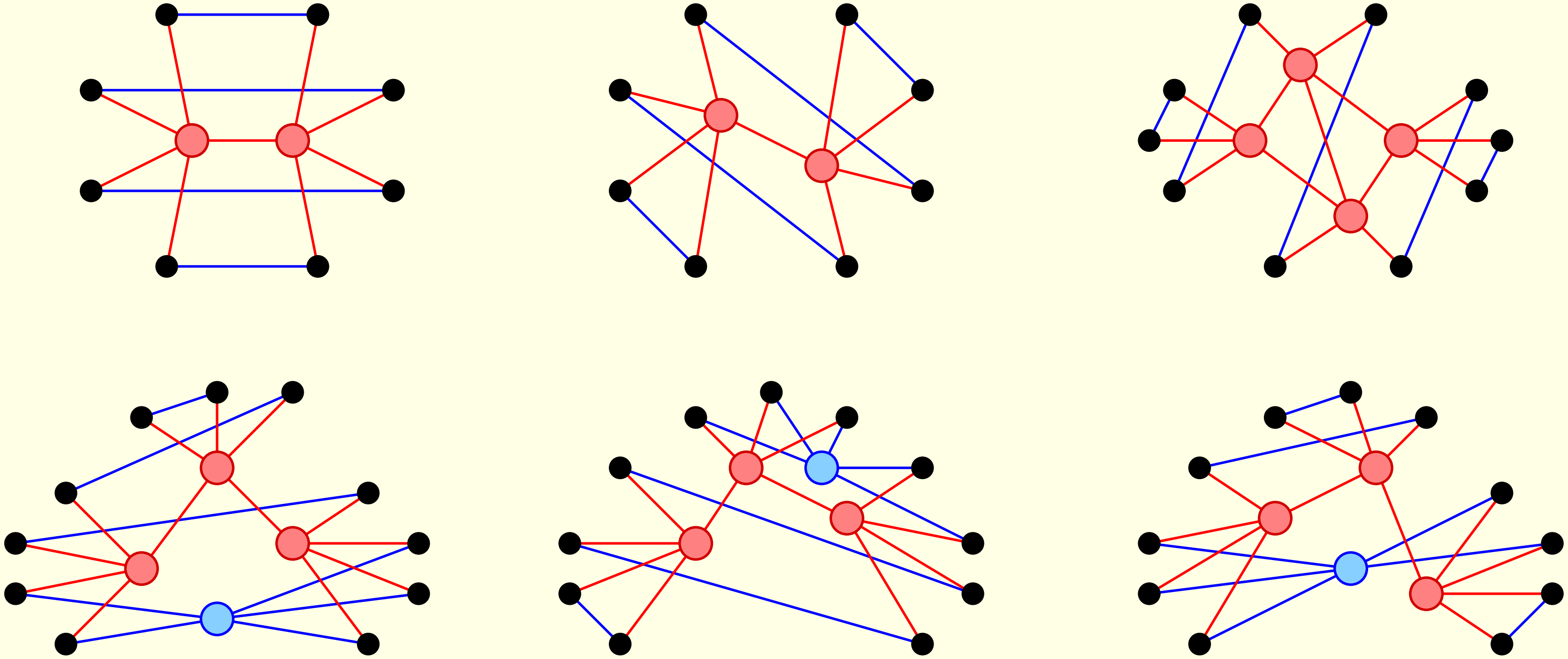




5-regular simple planar graphs

Coauthors: Mahdieh Hasheminezhad, Tristan Reeves

Expansion is to replace blue by red. Black vertices need not be distinct.

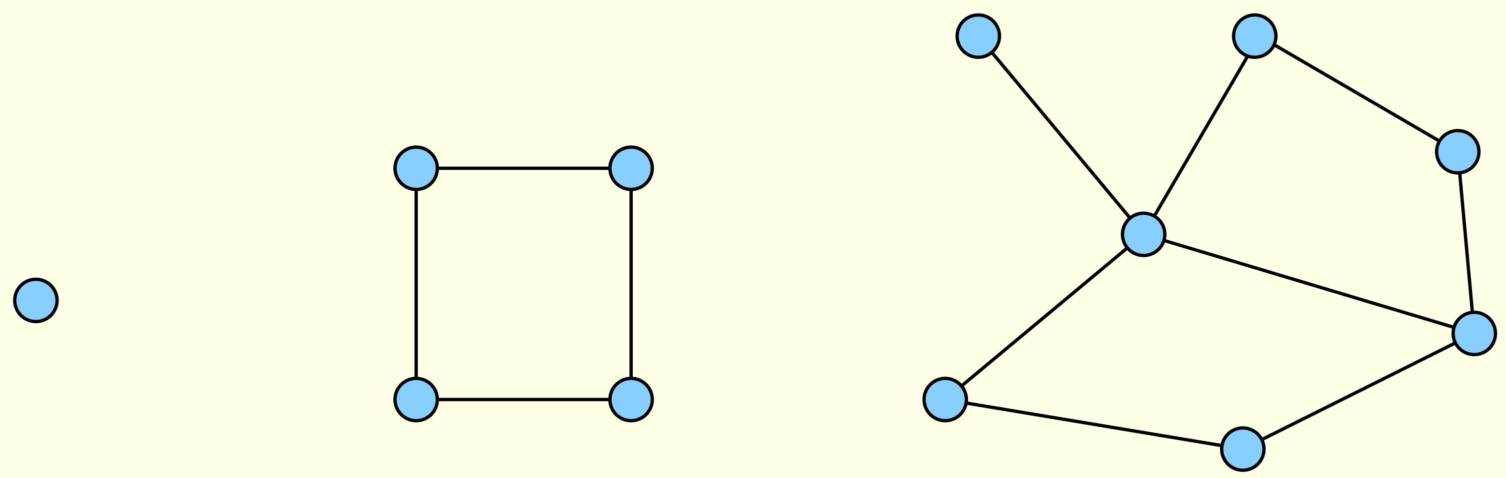


There one simple infinite class of irreducible graphs, and several sporadic irreducible graphs up to 72 vertices.

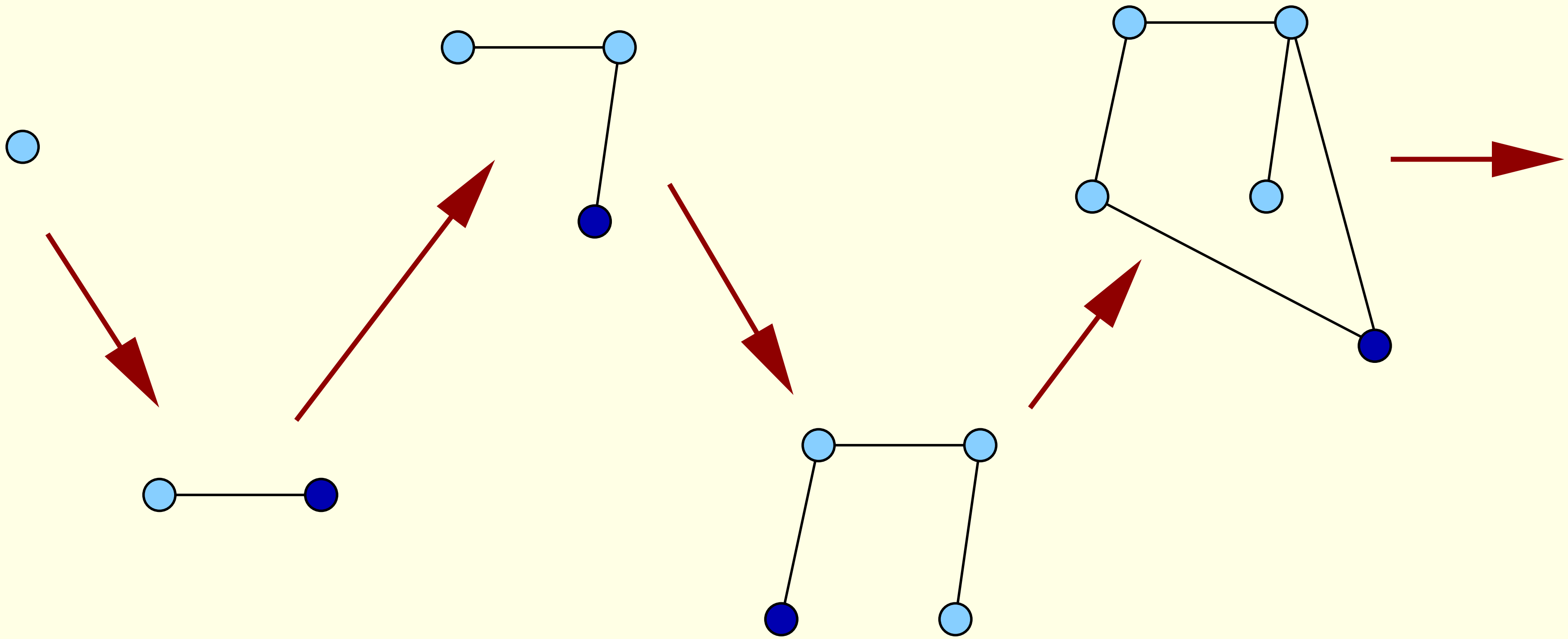
Isomorph-free generation

Once we have a recursive characterization, we can generate the graphs in the class, but how to we eliminate isomorphic copies?

Toy Example: connected triangle-free planar graphs



Obvious recursive construction: add one vertex at a time starting with one vertex:



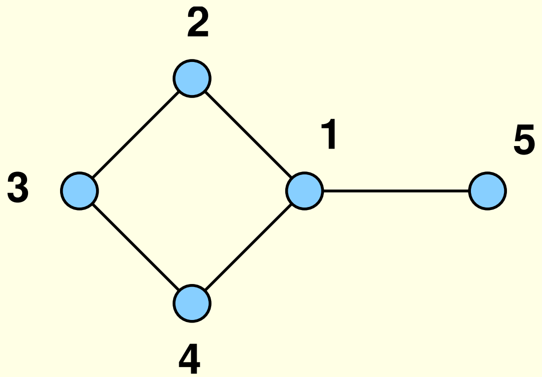
The difficulty is that isomorphic graphs appear.

Orderly Generation

Orderly generation is a method for complete isomorph rejection invented independently by Faradzev and Read (1978).

Each isomorphism class is represented by a unique "canonical" member of the class, which is usually the minimal member under some defined ordering. This is done in such a way that canonical objects can be made by extending smaller canonical objects.

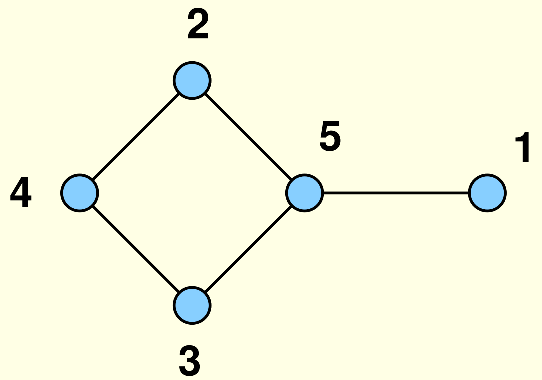
The generation process consists of extending canonical graphs and rejecting the resulting graphs if they are not canonical.



0	1	0	1	1
1	0	1	0	0
0	1	0	1	0
1	0	1	0	0
1	0	0	0	0

lower triangle

1 01 101 1000



0	0	0	0	1
0	0	0	1	1
0	0	0	1	1
0	1	1	0	0
1	1	1	0	0

0 00 011 1110

(canonical)

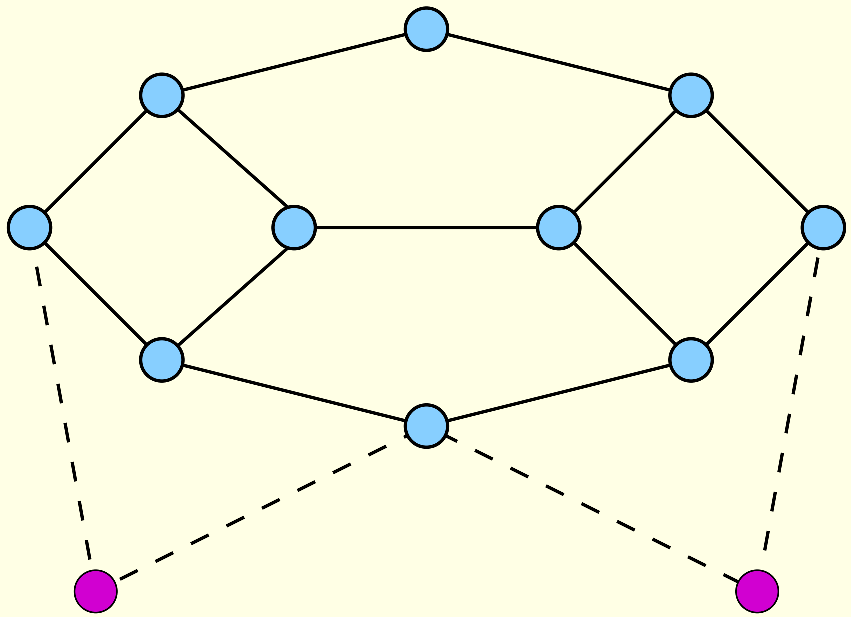
This definition of "canonical" has the property that removing the last vertex from a canonical graph gives a canonical graph.

The difficult part is the test for canonicity.

The need for the canonical form to have the hereditary property is a severe handicap for the canonicity test.

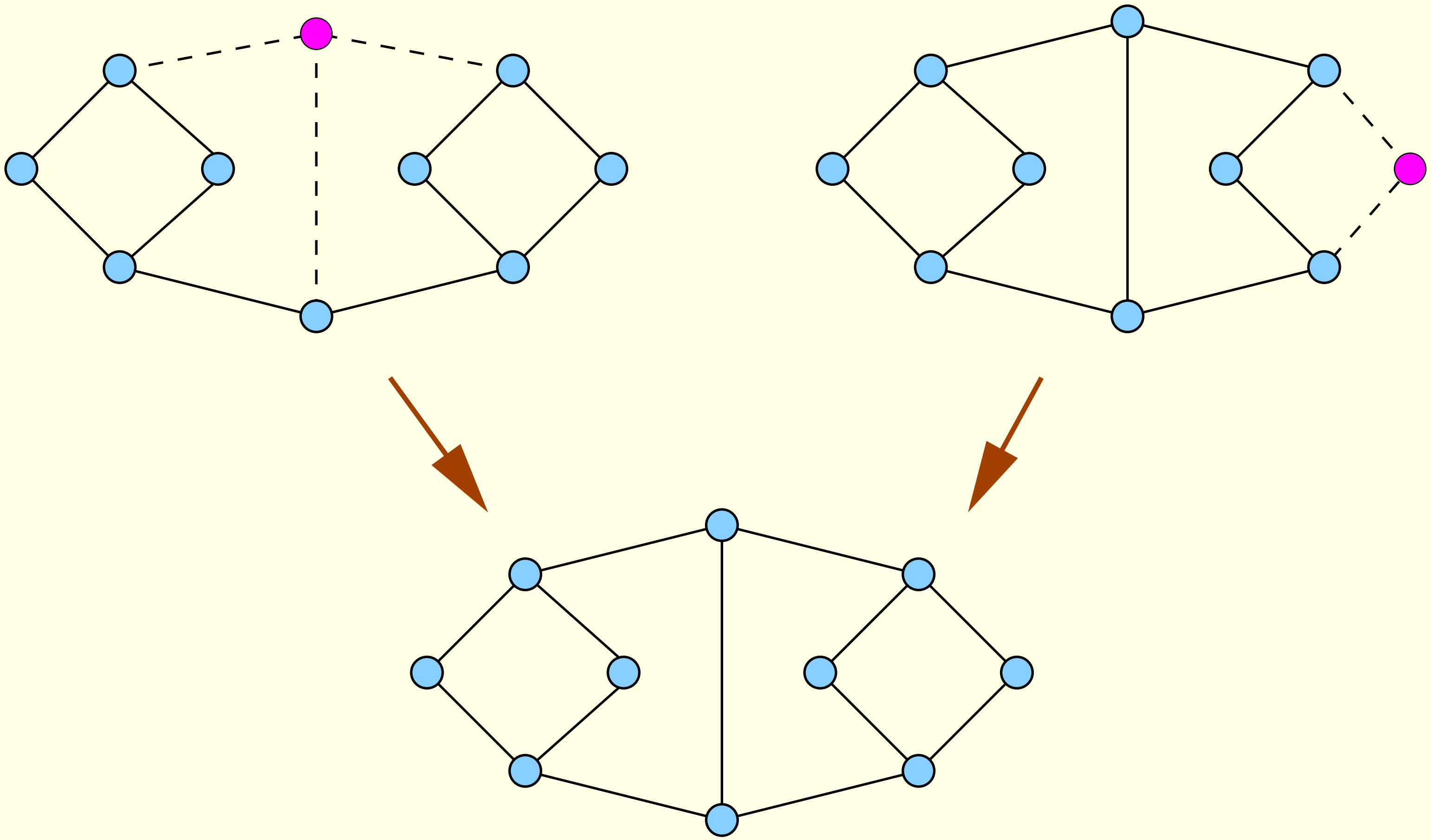
1st source of isomorphs: symmetry

Equivalent expansions result in isomorphic children.

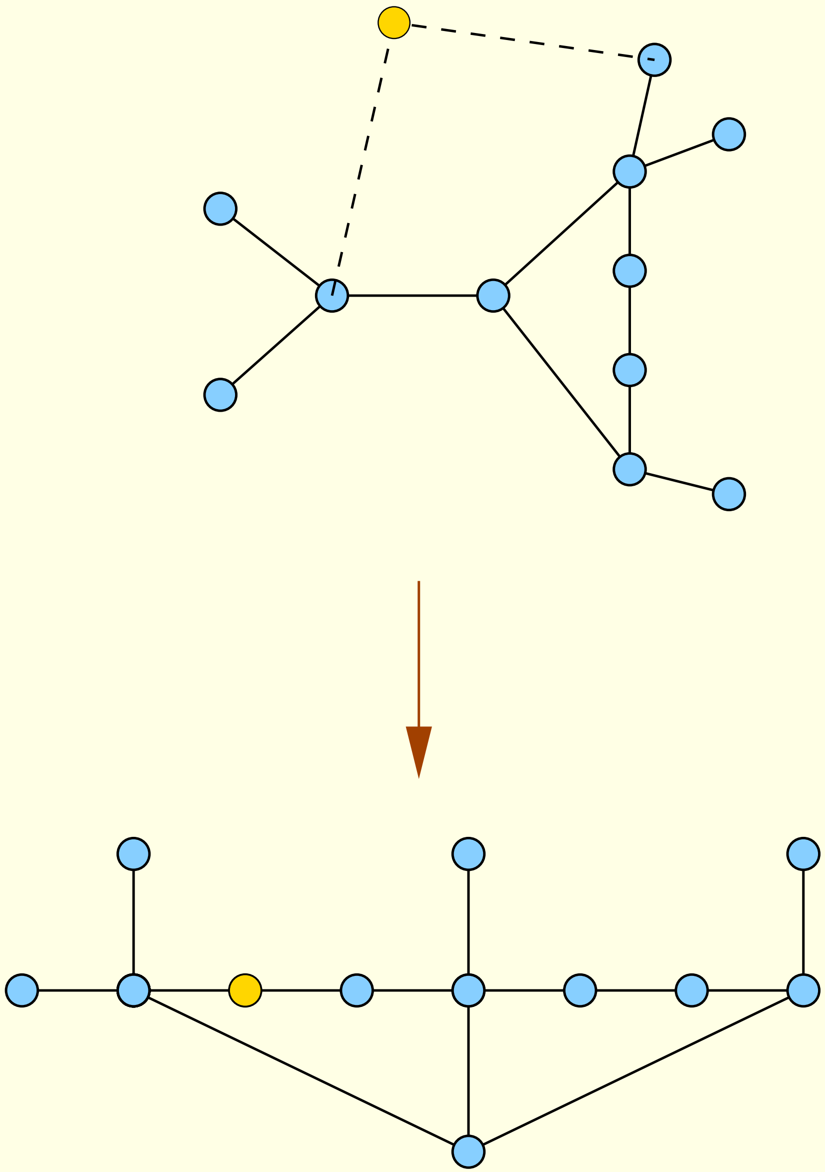


2nd source of isomorphs: different parents

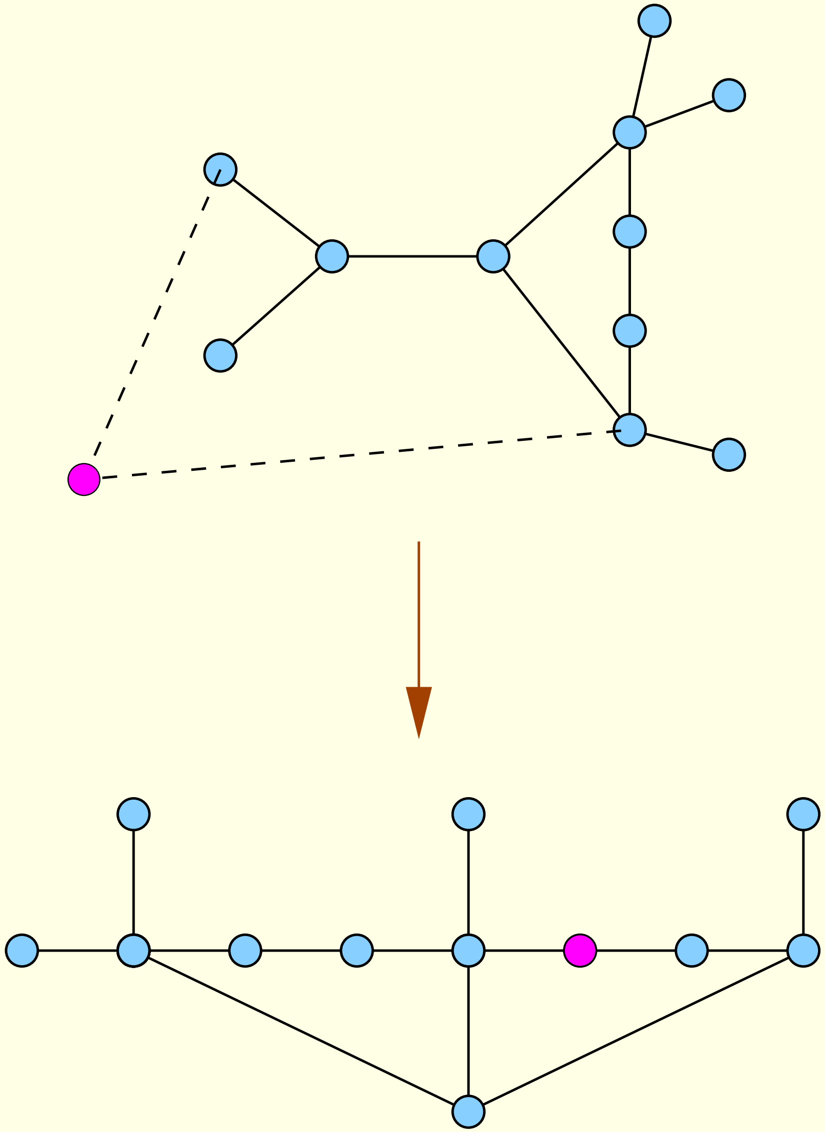
Slightly different parents can sometimes be extended to isomorphic children.



3rd source of isomorphs: pseudosimilarity



3rd source of isomorphs: pseudosimilarity



Generation by Canonical Construction Path

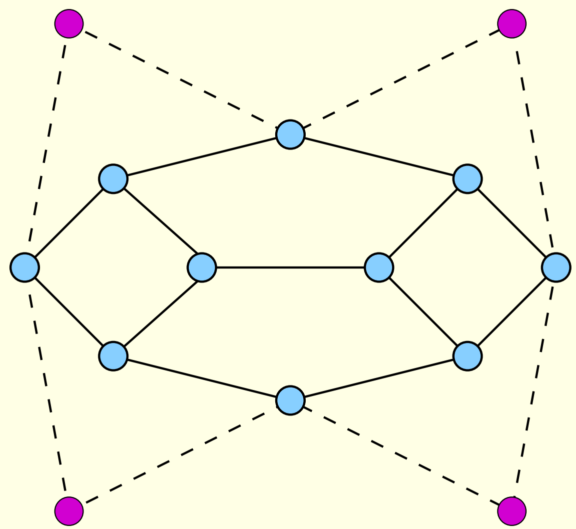
Also called canonical augmentation. McKay (1998)

Here we attempt to counter the three sources of isomorphs directly.

1st source: symmetry

Rule #1: Only make extensions inequivalent under the automorphism group of the smaller graph.

Perform at most one of these:



2nd and 3rd sources: different expansion

This includes construction from two different parents and construction from the same parent in two inequivalent ways.

For each reducible graph, define a canonical equivalence class of reductions. Here "canonical" means "independent of the labelling" and "equivalence class" means "equivalent under the automorphism group".

In the triangle-free graphs example, an equivalence class of reductions is an orbit of vertices.

A *canonical* orbit of vertices could be the orbit that contains the vertex labelled first by a canonical labelling program like `nauty`. (In practice, we use a layered sequence of invariants to choose an equivalence class without invoking `nauty` most of the time.)

2nd source: different expansion (continued)

Canonical orbit of reductions:

$$\mathcal{C} : \text{graph } G \to \text{orbit of reductions}$$

$$\mathcal{C}(G^\gamma) = \mathcal{C}(G)^\gamma \qquad (\gamma \in S_n)$$

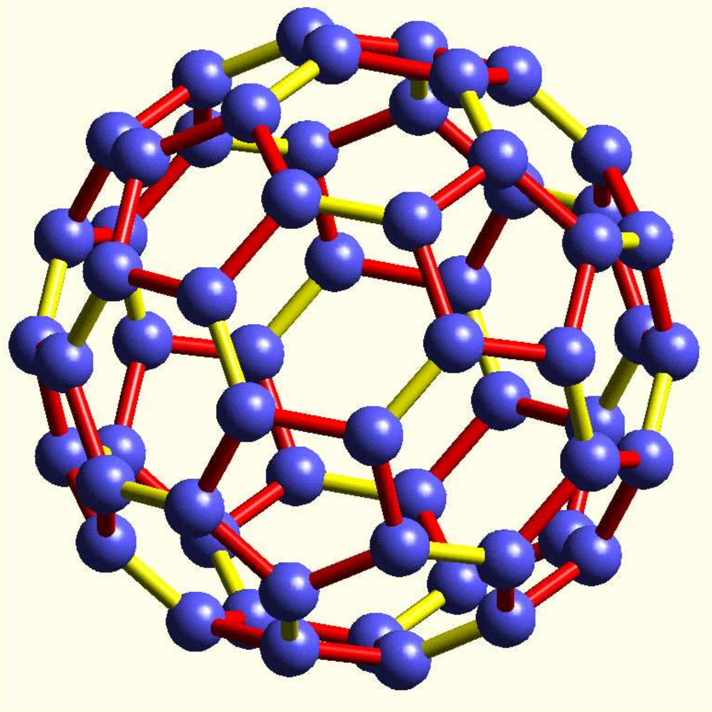
2nd source: different expansion (continued)

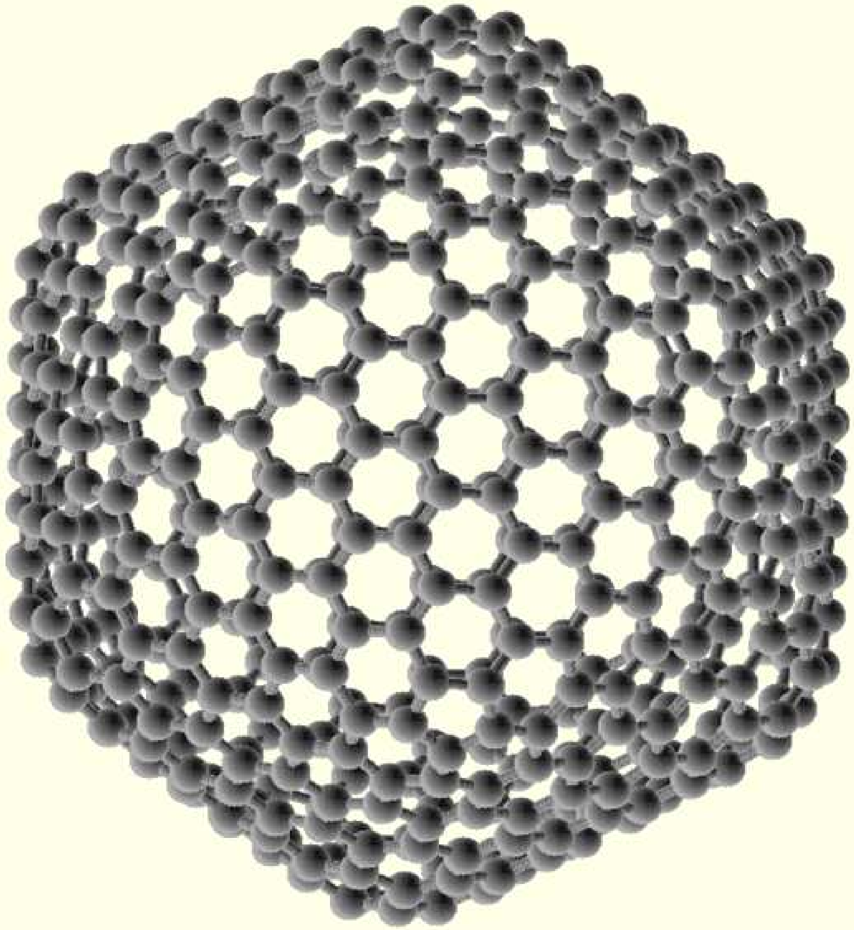
Rule #2: If object G is made using expansion ϕ, reject G unless $\phi^{-1} \in \mathcal{C}(G)$.

Theorem (McKay, 1989): If rules #1 and #2 are obeyed, and certain conditions hold, then all isomorphs are eliminated.

The "certain conditions" mostly involve the definition of symmetry.

Fullerenes





Fullerenes (combinatorially)

A fullerene is a simple 3-polytope with faces of size 5 and 6.

Alternatively:

A fullerene is a planar cubic graph with faces of size 5 and 6.

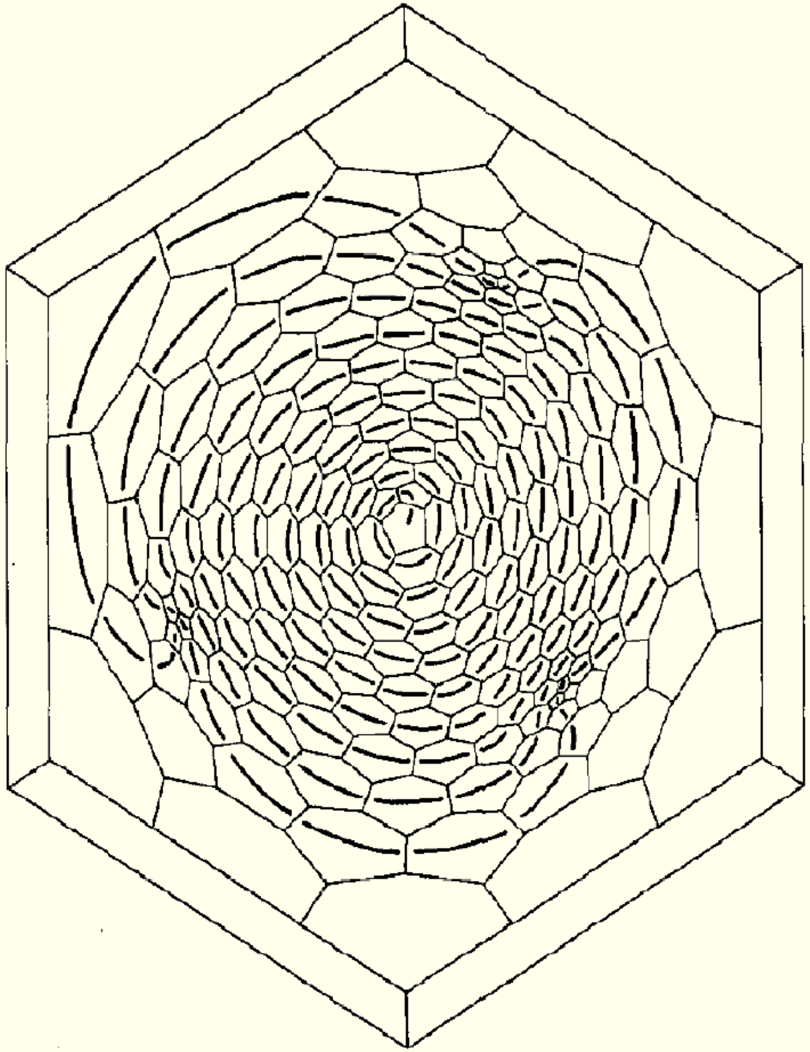
By means of Euler's polyhedral formula, it is routine to show that the number of pentagonal faces is exactly 12.

Isomorphism types of fullerenes

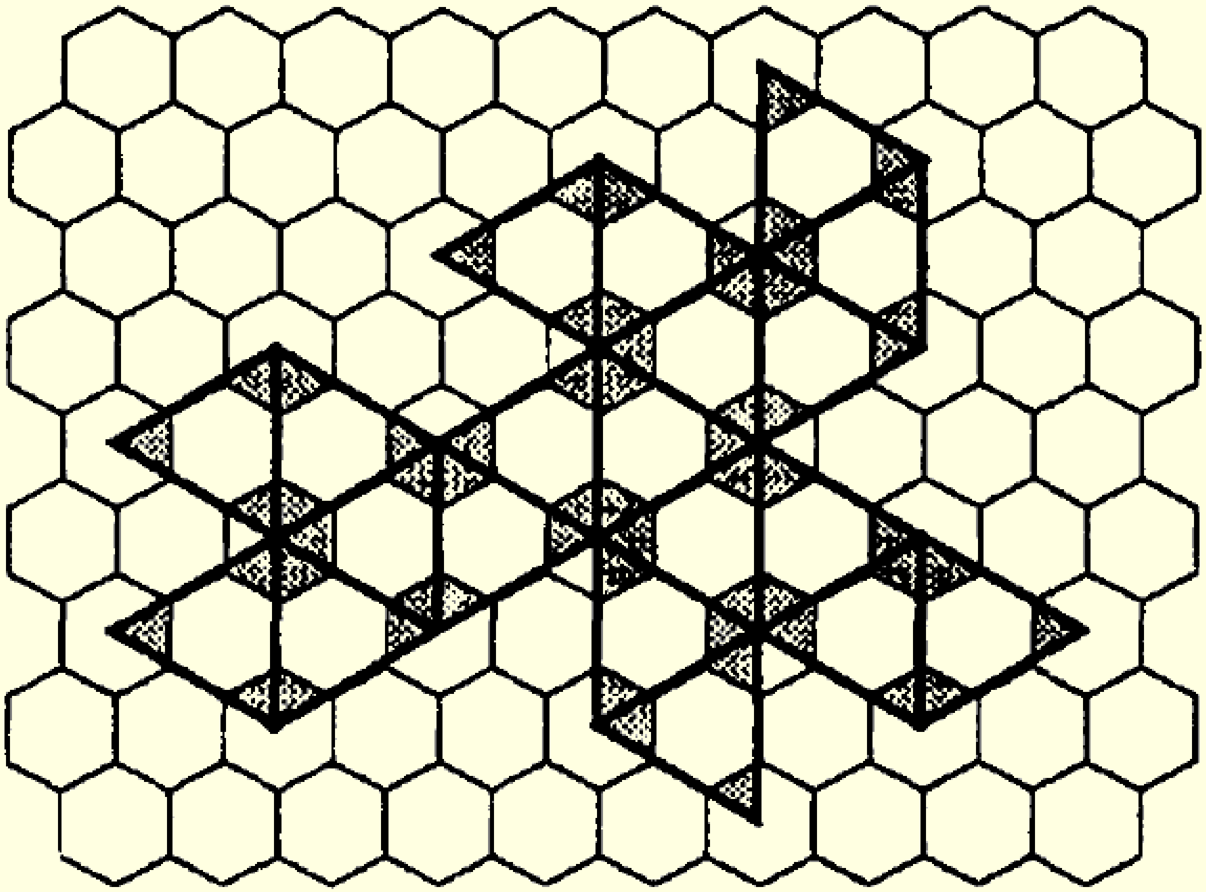
20	1
22	0
24	1
26	1
28	2
...	
60	1812
...	
100	285914
...	
200	214127742
...	
300	9332065811
...	
400	132247999328

Brief history of construction methods

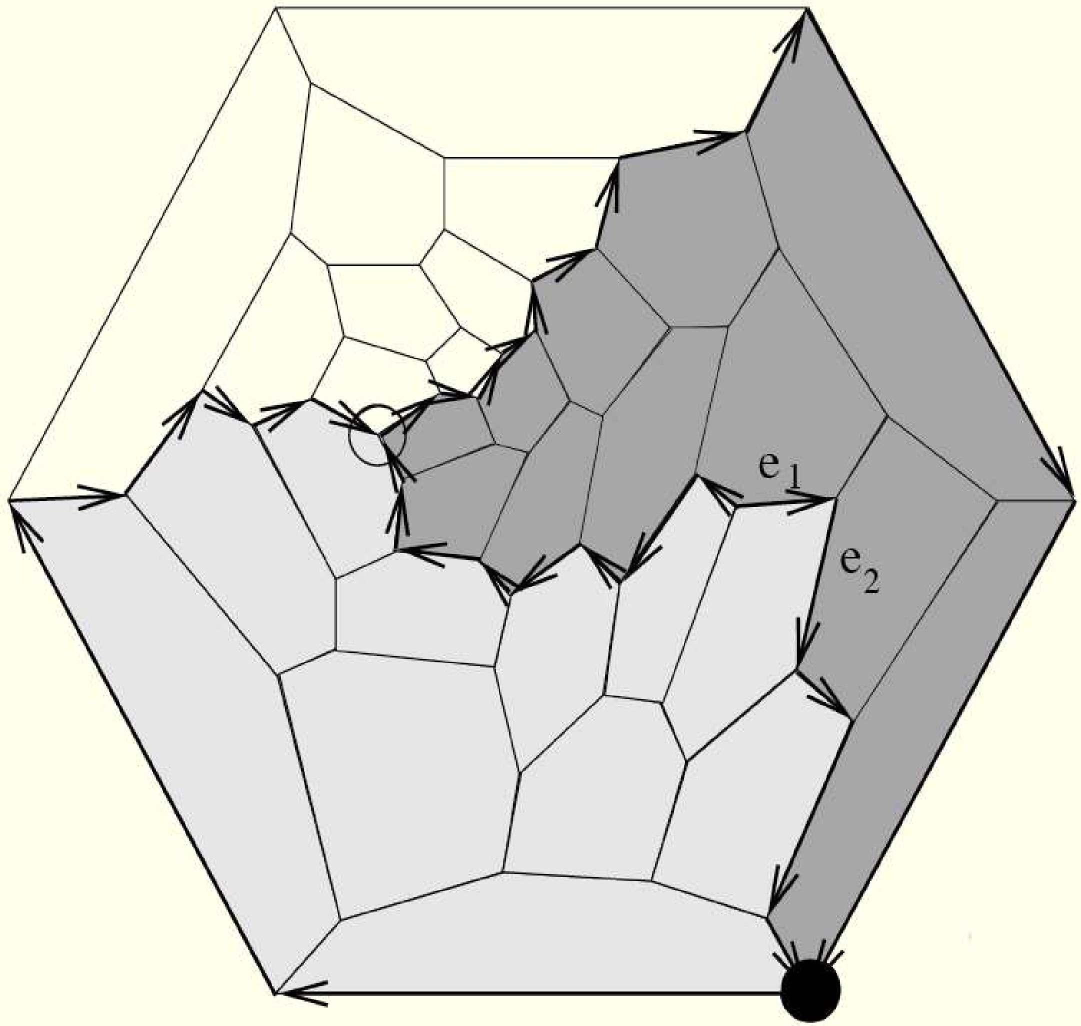
Manolopoulos et al. (1991) — spiral development



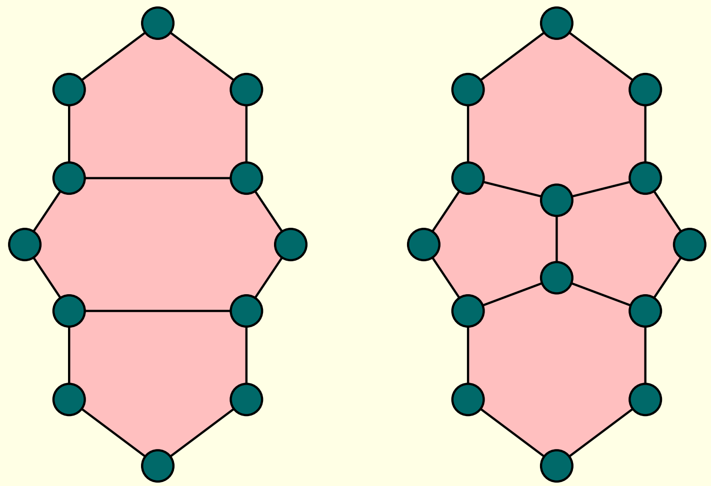
Yoshida and Osawa (1997) — folding nets



Brinkmann and Dress (1997) — zigzag (Petrie) paths

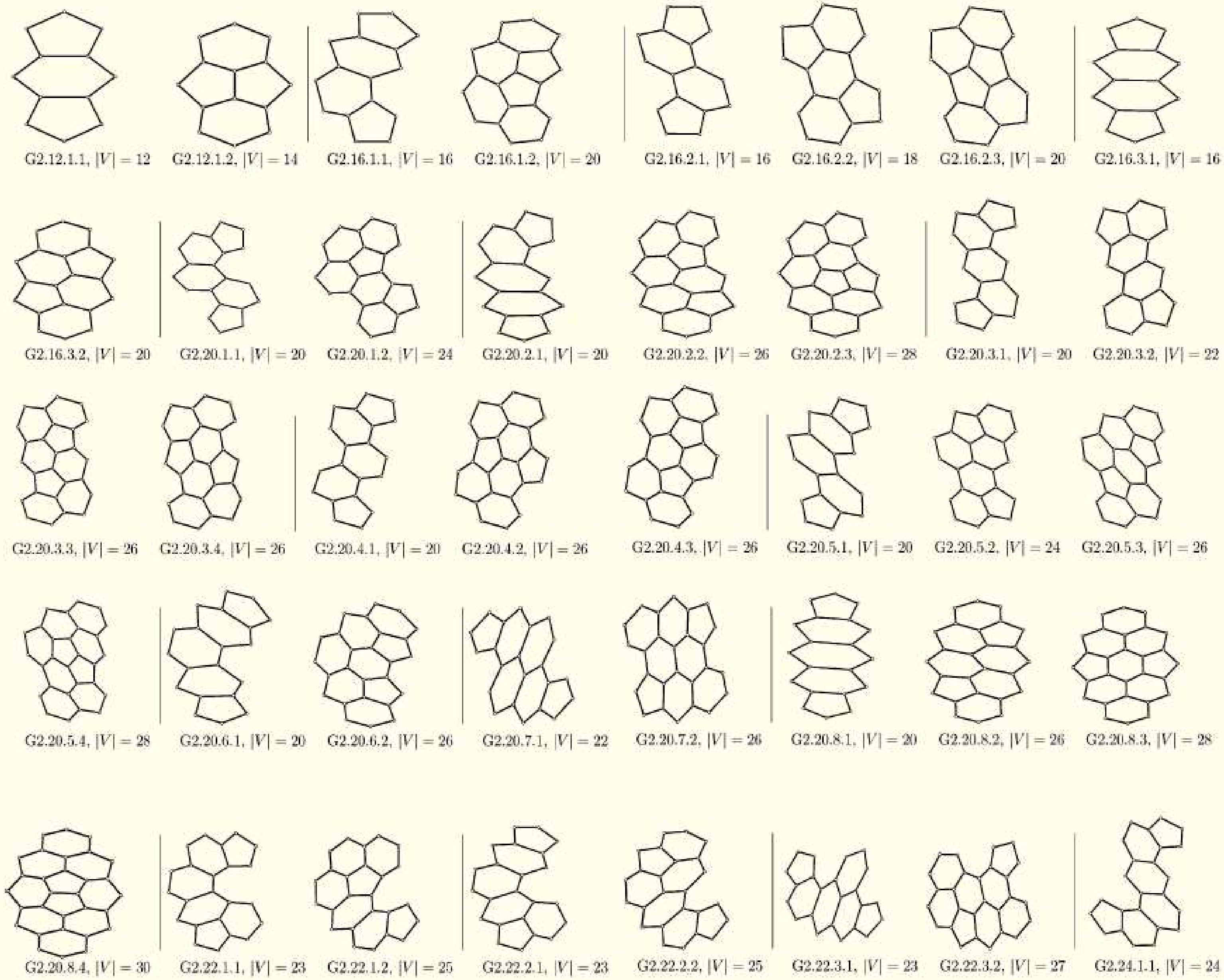


Fowler, Brinkmann, et al. (1993+) — growth patches



Endo–Kroto patch

Fowler, Brinkmann, et al. (1993+) — growth patches

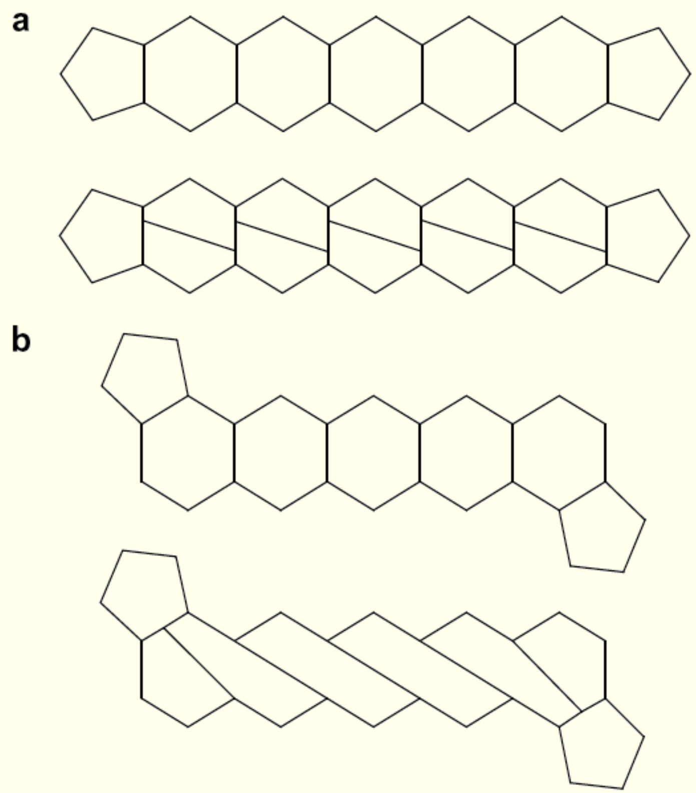


The problem with patches

- There are no growth patches with fewer than 2 pentagons (this is hard to prove: Brinkmann, Graver, Justus).
- The pentagons in a fullerene can be arbitrarily far apart.
- Therefore, a finite set of growth patches won't suffice.

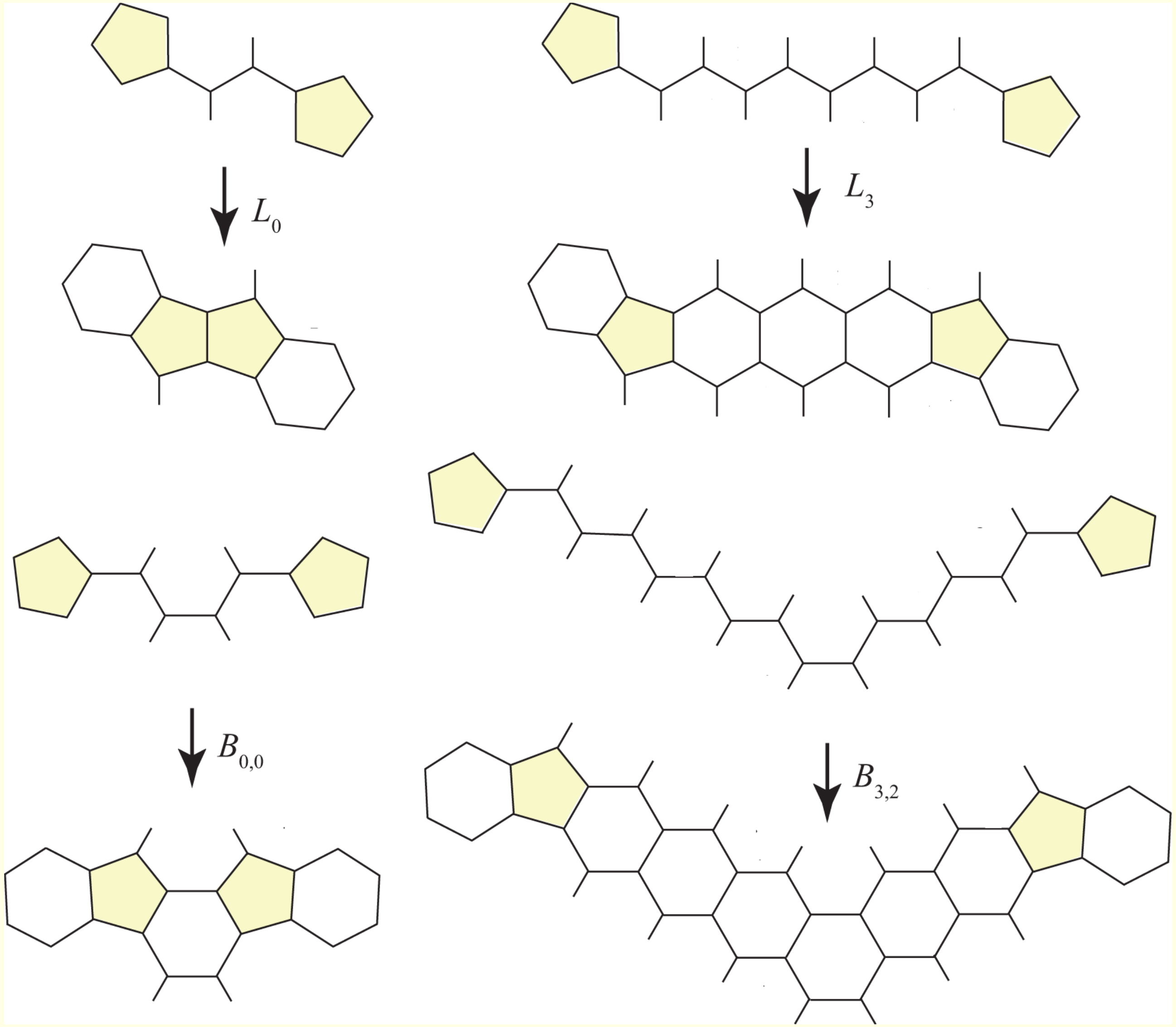
We seek instead some sufficient infinite family of patches.

Brinkmann, Franceus, Fowler & Graver (2006)

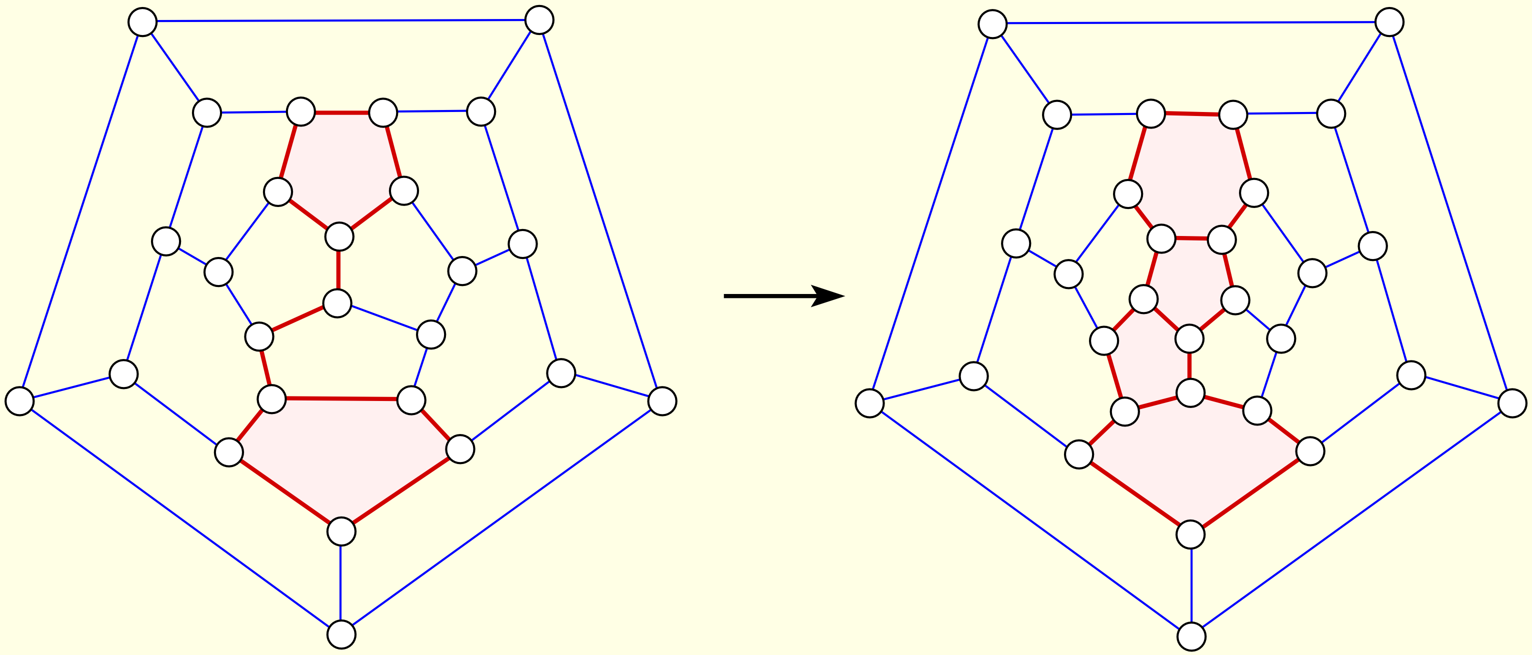


These suffice to at least 200 vertices but fail in general.

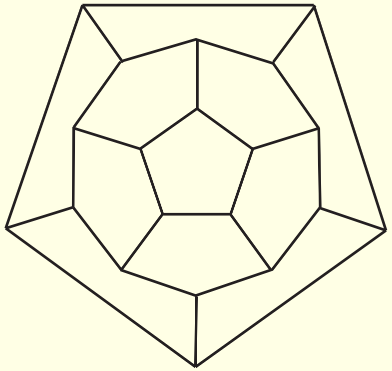
Hasheminezhad, Fleishner and McKay (2008)



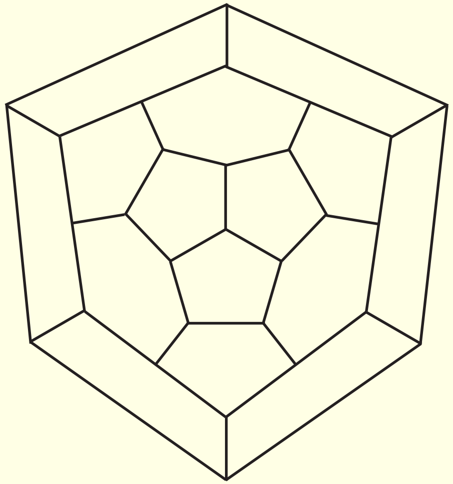
Hasheminezhad, Fleishner and McKay (2008)



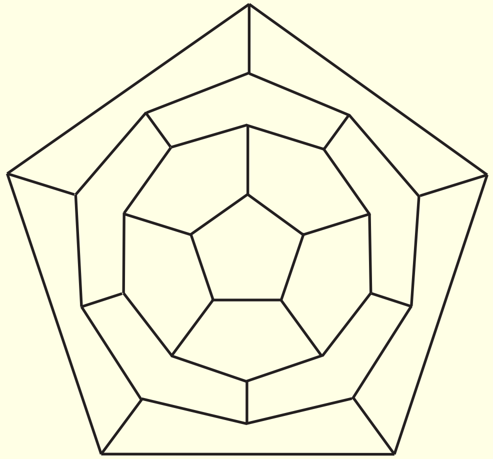
Irreducible graphs:



C_{20}



$C_{28}(T_d)$



$C_{30}(D_{5h})$

By adding additional rings of hexagons to $C_{30}(D_{5h})$, the (5,0)-nanotube type fullerenes are formed.

Implementation

Coauthors: Gunnar Brinkmann, Jan Goedgebeur

The canonical construction path method requires:

- Computation of all automorphisms.
- Computation of a "canonical reduction".

Both can be done in $O(n)$ time using DFS starting at each pentagon, with the embedding used to remove nondeterminism. (Remember there are exactly 12 pentagons.)

Efficiency

Theorem (McKay, 1989): The number of objects constructed is at most K times the number of objects accepted, where K is the average number of reductions per object.

The number of reductions per fullerene is bounded, since each involves a straight or single-bend path of hexagons between two pentagons.

Therefore the time per fullerene is $O(n)$.

In practice around 200,000 fullerenes per second.

IPR fullerenes (2015)

Coauthor: Jan Goedgebeur

Isolated Pentagon Rule: Fullerenes without adjacent pentagons are more likely to be chemically stable.

Therefore, it is of interest to generate only those without adjacent pentagons. For small sizes they are a minority, though they eventually become a majority for very large size.

Efficient methods:

1. IPR patches (Brinkmann and Dress)
2. Lookahead in fullerene generator
3. New method

IPR fullerenes (continued)

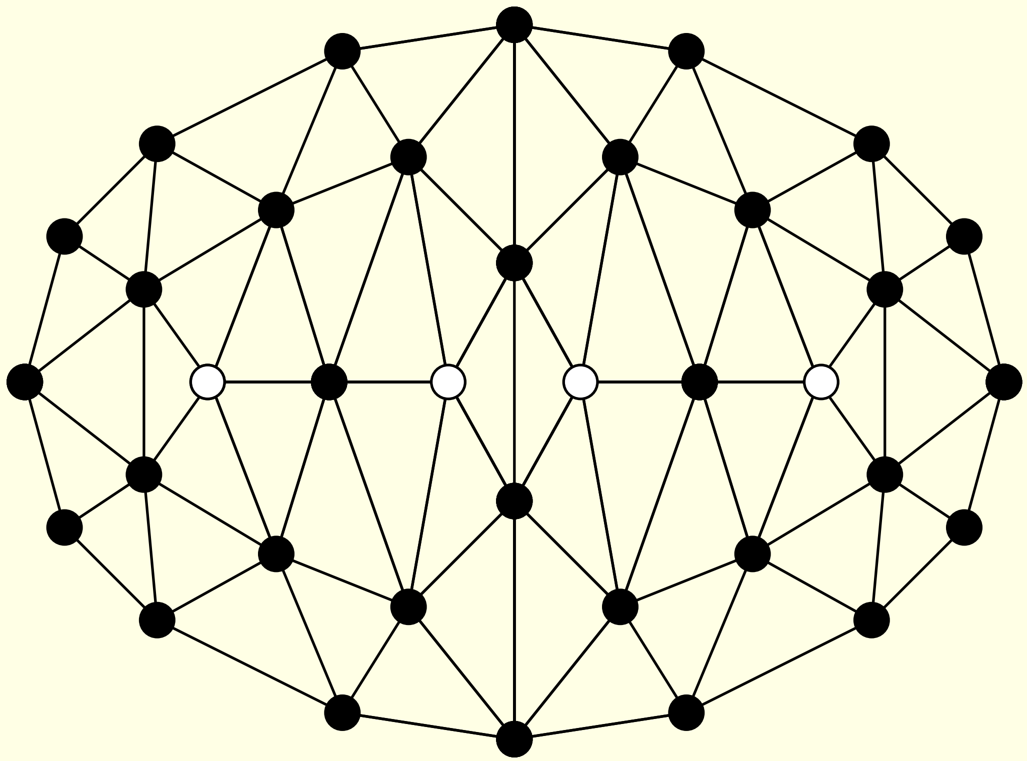
The new method uses the Hasheminezhad-McKay expansions but remains within the class of IPR-fullerenes.

The problem is in determining what the irreducible graphs are.

IPR fullerenes (continued)

To identify the irreducible graphs, we consider subgraphs called *k*-clusters. (The precise definition is technical.)

Here is an example of a 4-cluster, shown in the dual.
The white vertices have degree 5 in the complete dual-fullerene.



IPR fullerenes (continued)

For all fullerenes with only 1-clusters, there is a reduction between two clusters.

For all k-clusters with $2 \leq k \leq 5$, there is either a reduction within the cluster, or using a few vertices just outside the cluster.

All fullerenes with a k-cluster for $7 \leq k \leq 11$ are likewise reducible.

IPR fullerenes (continued)

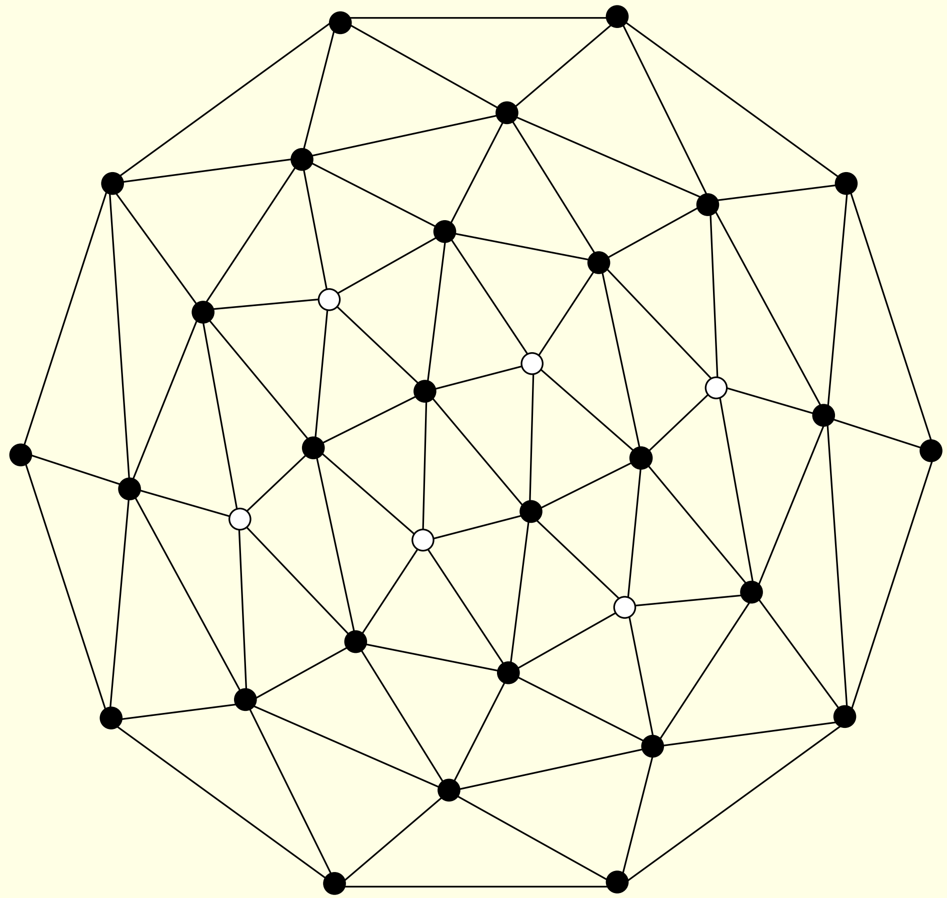
For all fullerenes with only 1-clusters, there is a reduction between two clusters.

For all k-clusters with $2 \leq k \leq 5$, there is either a reduction within the cluster, or using a few vertices just outside the cluster.

All fullerenes with a k-cluster for $7 \leq k \leq 11$ are likewise reducible.

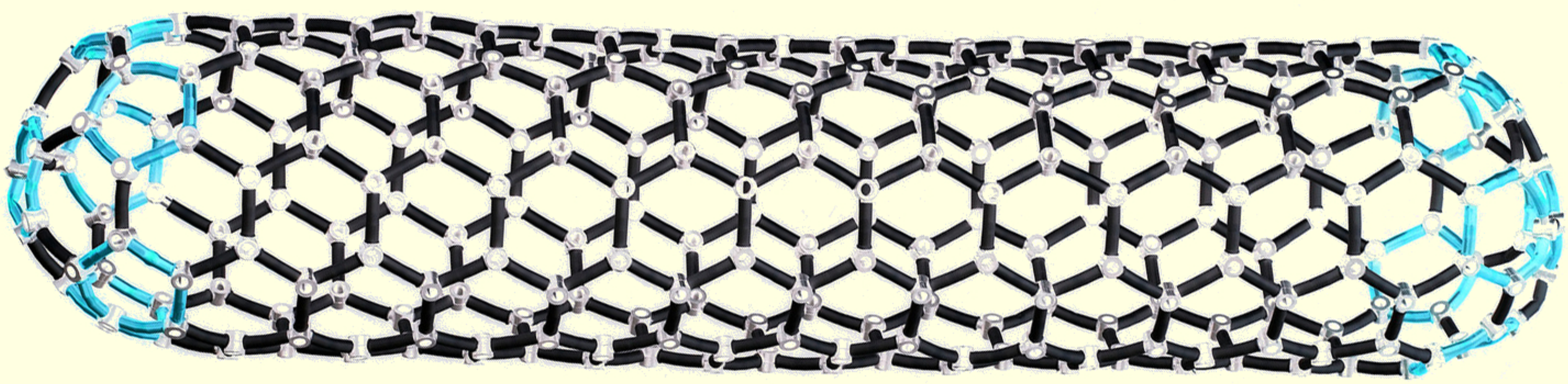
The irreducible fullerenes have either a single 12-cluster
or two 6-clusters.

This 6-cluster is irreducible.



IPR fullerenes (continued)

There are 4 infinite families of irreducible IPR fullerenes formed by making an arbitrarily long tube of hexagons and closing them up with a 6-cluster at each end. The two ends have to be the same 6-clusters.



And there are 36 sporadic irreducible fullerenes.

The program implemented by Jan outperforms all previous programs for IPR fullerenes, though the advantage decreases for very large sizes.