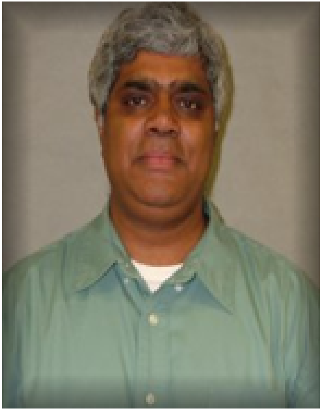
Automated Conjecturing for Proof Discovery

Craig Larson
(joint work with Nico Van Cleemput)

Virginia Commonwealth University
Ghent University

CombinaTexas
Texas A&M University
8 May 2016

Kiran Chilakamarri—On Conjectures



Goal

To tell you about a new idea for using our conjecture-generating program:

Goal

To tell you about a new idea for using our conjecture-generating program:

generating sketches of proofs (or proof ideas);

Goal

To tell you about a new idea for using our conjecture-generating program:

generating sketches of proofs (or proof ideas);

and a new proof of the Friendship Theorem.

Our Program

Black box.

Our Program

Black box.

Main heuristic idea from Fajtlowicz's Graffiti.

Our Program

Black box.

Main heuristic idea from Fajtlowicz's Graffiti.

Ingredients: Objects, Invariants, Properties, Choice of Invariant or property of interest, choice of upper or lower bounds.

The Program

- Open-source

The Program

- Open-source
- Written for Sage

The Program

- Open-source
- Written for Sage
- Python

The Program

- Open-source
- Written for Sage
- Python
- Lots of standard mathematical packages: GAP, R, GLPK, CVXOPT, NumPy, SciPy, LaTeX, matlabplot

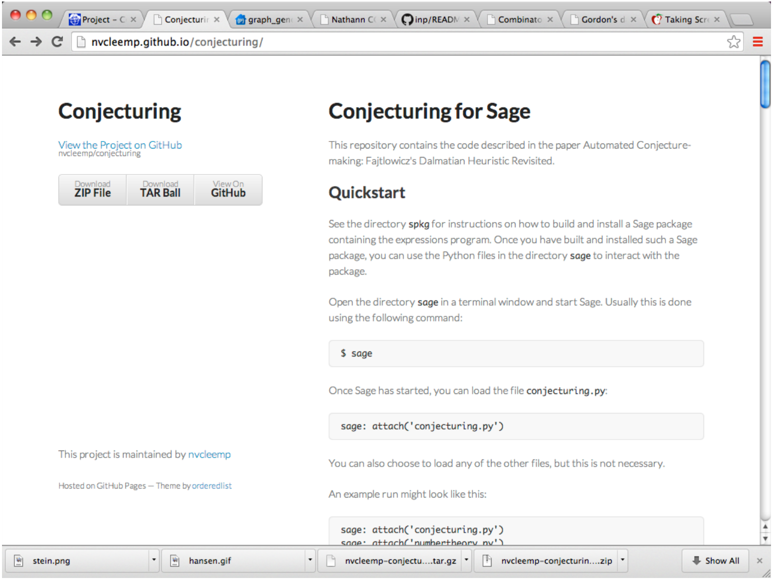
The Program

- Open-source
- Written for Sage
- Python
- Lots of standard mathematical packages: GAP, R, GLPK, CVXOPT, NumPy, SciPy, LaTeX, matlabplot
- Lots of Graph Theory: graphs, invariants, properties, constructors,...

The Program

- Open-source
- Written for Sage
- Python
- Lots of standard mathematical packages: GAP, R, GLPK, CVXOPT, NumPy, SciPy, LaTeX, matlabplot
- Lots of Graph Theory: graphs, invariants, properties, constructors,. . .
- Growing

The Program



nvcleemp.github.io/conjecturing/

Conjecturing

View the Project on GitHub
nvcleemp/conjecturing

Download ZIP File | Download TAR Ball | View On GitHub

This project is maintained by nvcleemp

Hosted on GitHub Pages — Theme by orderedlist

Conjecturing for Sage

This repository contains the code described in the paper Automated Conjecture-making: Fajtlowicz's Dalmatian Heuristic Revisited.

Quickstart

See the directory **spkg** for instructions on how to build and install a Sage package containing the expressions program. Once you have built and installed such a Sage package, you can use the Python files in the directory **sage** to interact with the package.

Open the directory **sage** in a terminal window and start Sage. Usually this is done using the following command:

```
$ sage
```

Once Sage has started, you can load the file `conjecturing.py`:

```
sage: attach('conjecturing.py')
```

You can also choose to load any of the other files, but this is not necessary.

An example run might look like this:

```
sage: attach('conjecturing.py')
sage: attach('numbertheory.py')
```

stein.png | hansen.gif | nvcleemp-conjectu....tar.gz | nvcleemp-conjecturin....zip | Show All

Properties

Example: `pairs_have_unique_common_neighbor`:

Every pair of vertices has exactly one common neighbor.

Properties

Example: `pairs_have_unique_common_neighbor`:

Every pair of vertices has exactly one common neighbor.

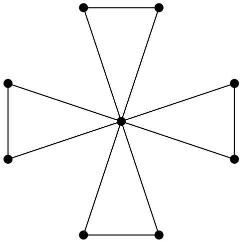


Figure: A flower F_4 with four petals.

Sufficient Condition Conjectures

164 graphs in the main database
87 properties
5 propositional operators: and, or, implies, not, xor

Sufficient Condition Conjectures

164 graphs in the main database
87 properties
5 propositional operators: and, or, implies, not, xor

```
#using ALL graph_objects
#sufficient condition conjectures for pairs_have_unique_common_neighbor


property = properties.index(pairs_have_unique_common_neighbor)

conjs = propertyBasedConjecture(graph_objects, properties, property,
                                sufficient = True, precomputed = precomputed)
for c in conjs:
    print c
```

```
> Generation process was stopped by the conjecturing heuristic.
> Found 5 unlabeled trees.
> Found 24895 labeled trees.
> Found 486 valid expressions.
((~(is_triangle_free))&(is_cycle))->(pairs_have_unique_common_neighbor)
((~(is_split))&(has_star_center))->(pairs_have_unique_common_neighbor)
```

What you get

What you get

Conjectures that are true for *all* input objects.

What you get

Conjectures that are true for *all* input objects.

Conjectures that say something not implied by any previously output conjecture.

Theory

Idea: You want conjectures that are an improvement on existing theory.

Theory

Idea: You want conjectures that are an improvement on existing theory.

```
property = pairs_have_unique_common_neighbor

theory = [is_k3]

propertyBasedConjecture(graph_objects, properties,
    property, theory = theory, sufficient = True,
    precomputed = precomputed)
```

Theory

Idea: You want conjectures that are an improvement on existing theory.

```
property = pairs_have_unique_common_neighbor

theory = [is_k3]

propertyBasedConjecture(graph_objects, properties,
    property, theory = theory, sufficient = True,
    precomputed = precomputed)
```

Conjecture:

```
(((is_eulerian)&(is_planar))&(is_gallai_tree))->
    (pairs_have_unique_common_neighbor)
```

Proof Sketch Generating Idea

- Prove: $P \implies Q$.
- Run necessary condition conjectures for P, using Q as the "theory".
- The generated conjectures must be "better" than Q for at least one graph conjectures.
- Get:

$$P \implies C_1$$

$$P \implies C_2$$

- By the truth test, each object x that has property P has properties C_1 and C_2.
- Thus x is in the intersection of the set of graphs having properties C_1 and C_2.
- If there is x in the graph database with x in $C_1 \cap C_2$ but $x \notin Q$ then program wouldn't stop—as conjectures could be improved.

Proof Sketch Generating Idea

- Prove: $P \implies Q$.
- Run necessary condition conjectures for P, using Q as the "theory".
- Lemma 1:
$$P \implies C_1$$
- Lemma 2:
$$P \implies C_2$$
- Lemma 3:
$$C_1 \cap C_2 \subseteq Q$$
- Then lemmas imply Theorem:
$$P \implies Q$$
- (semantic proof)

The Friendship Theorem

If every pair of people in a group have exactly one friend in common, then there is a person in the group that is friends with all of them.

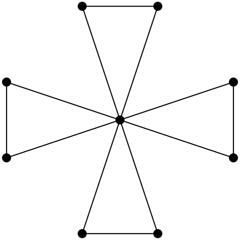
The Friendship Theorem

If every pair of people in a group have exactly one friend in common, then there is a person in the group that is friends with all of them.

If every pair of vertices in a graph have a unique common neighbor, then there is a vertex in the graph that is adjacent to all the other vertices (Erdős, Rényi, Sós, 1966).

The Friendship Theorem

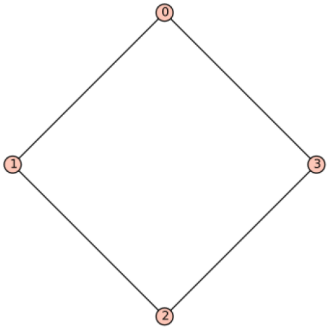
If every pair of vertices in a graph have a unique common neighbor, then there is a vertex in the graph that is adjacent to all the other vertices.



The Friendship Theorem

If every pair of vertices in a graph have a unique common neighbor, then there is a vertex in the graph that is adjacent to all the other vertices.

Useful Observation: Can't have any four-cycles.



Conjectured Lemmas

Investigate:

`(pairs_have_unique_common_neighbor)->(has_star_center)`

Conjectured Lemmas

Investigate:

```
(pairs_have_unique_common_neighbor)->(has_star_center)
```

Investigation:

```
property = pairs_have_unique_common_neighbor

theory = [has_star_center]
```

Conjectured Lemmas

Investigate:

```
(pairs_have_unique_common_neighbor)->(has_star_center)
```

Investigation:

```
property = pairs_have_unique_common_neighbor

theory = [has_star_center]
```

Conjectures:

```
(pairs_have_unique_common_neighbor)->(is_eulerian)
(pairs_have_unique_common_neighbor)->(is_circular_planar)
(pairs_have_unique_common_neighbor)->(is_gallai_tree)
```

Conjectured Lemma 1

```
(pairs_have_unique_common_neighbor)->(is_eulerian)
```

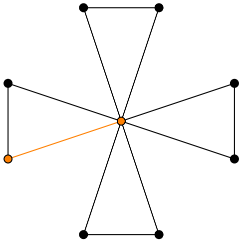
Euler's criterion: A connected graph is eulerian if and only if every degree is even.

Conjectured Lemma 1

```
(pairs_have_unique_common_neighbor)->(is_eulerian)
```

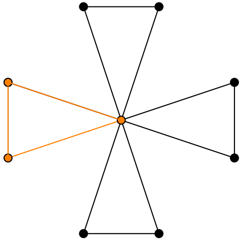
Conjectured Lemma 1

```
(pairs_have_unique_common_neighbor)->(is_eulerian)
```



Conjectured Lemma 1

`(pairs_have_unique_common_neighbor)->(is_eulerian)`



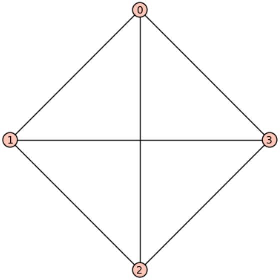
Conjectured Lemma 2

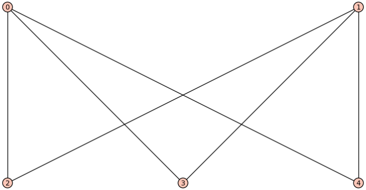
```
(pairs_have_unique_common_neighbor)->(is_circular_planar)
```

Theorem (Chartrand & Harary, 1967) A graph is outerplanar if and only if it does not contain a subdivision of K_4 or $K_{3,3}$.

Conjectured Lemma 2

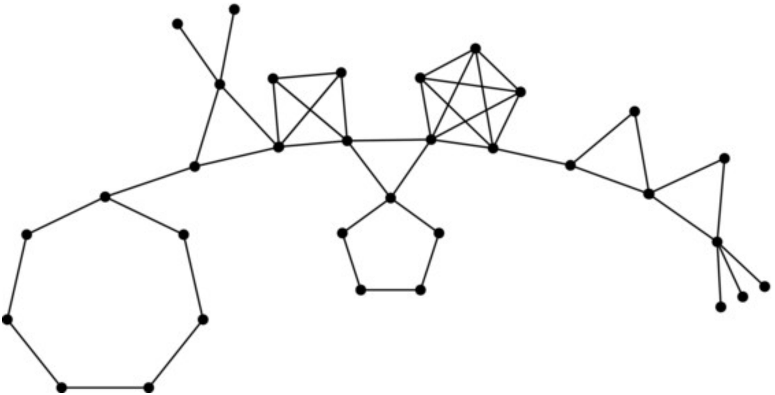
```
(pairs_have_unique_common_neighbor)->(is_circular_planar)
```





Conjectured Lemma 3

```
(pairs_have_unique_common_neighbor)->(is_gallai_tree)
```



Conjectured Lemma 3

Proof Main Ideas:

- Two-connected Gallai-trees are complete graphs or odd cycles.

Conjectured Lemma 3

Proof Main Ideas:

- Two-connected Gallai-trees are complete graphs or odd cycles.
- Outerplanar and Gallai-tree means the blocks are odd cycles.

Conjectured Lemma 3

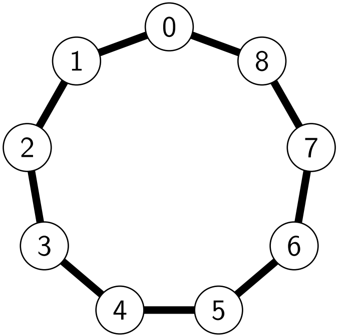
Proof Main Ideas:

- Two-connected Gallai-trees are complete graphs or odd cycles.
- Outerplanar and Gallai-tree means the blocks are odd cycles.
- If cycle has degree two vertices then the neighbors of these vertices must be adjacent.

Conjectured Lemma 3

Proof Main Ideas:

- Two-connected Gallai-trees are complete graphs or odd cycles.
- Outerplanar and Gallai-tree means the blocks are odd cycles.
- If cycle has degree two vertices then the neighbors of these vertices must be adjacent.



Conjectured Lemma 3

Proof Main Ideas:

- Assume the graph is two-connected. It is outerplanar and eulerian.

Conjectured Lemma 3

Proof Main Ideas:

- Assume the graph is two-connected. It is outerplanar and eulerian.
- If the graph is *not* a cycle, there is a vertex of degree at least 4.

Conjectured Lemma 3

Proof Main Ideas:

- Assume the graph is two-connected. It is outerplanar and eulerian.
- If the graph is *not* a cycle, there is a vertex of degree at least 4.
- Then the unique neighbor condition yields a contradiction.

Conjectured Lemma 3

Proof Main Ideas:

- Assume the graph is two-connected. It is outerplanar and eulerian.
- If the graph is *not* a cycle, there is a vertex of degree at least 4.
- Then the unique neighbor condition yields a contradiction.
- So the graph is a triangle.

Conjectured Lemma 3

Proof Main Ideas:

- Assume the graph is two-connected. It is outerplanar and eulerian.
- If the graph is *not* a cycle, there is a vertex of degree at least 4.
- Then the unique neighbor condition yields a contradiction.
- So the graph is a triangle.
- Apply induction.

Conjectured Lemma 3

Proof Main Ideas:

- Assume the graph is two-connected. It is outerplanar and eulerian.
- If the graph is *not* a cycle, there is a vertex of degree at least 4.
- Then the unique neighbor condition yields a contradiction.
- So the graph is a triangle.
- Apply induction.
- Assume the graph has a cut vertex v.

Conjectured Lemma 3

Proof Main Ideas:

- Assume the graph is two-connected. It is outerplanar and eulerian.
- If the graph is *not* a cycle, there is a vertex of degree at least 4.
- Then the unique neighbor condition yields a contradiction.
- So the graph is a triangle.
- Apply induction.
- Assume the graph has a cut vertex v.
- Each block is two-connected and must have the property that every pair of vertices has a unique common neighbor.

Conjectured Lemma 3

Proof Main Ideas:

- Assume the graph is two-connected. It is outerplanar and eulerian.
- If the graph is *not* a cycle, there is a vertex of degree at least 4.
- Then the unique neighbor condition yields a contradiction.
- So the graph is a triangle.
- Apply induction.
- Assume the graph has a cut vertex v.
- Each block is two-connected and must have the property that every pair of vertices has a unique common neighbor.
- So each block is a triangle.

Conjectured Lemma 3

Proof Main Ideas:

- Assume the graph is two-connected. It is outerplanar and eulerian.
- If the graph is *not* a cycle, there is a vertex of degree at least 4.
- Then the unique neighbor condition yields a contradiction.
- So the graph is a triangle.
- Apply induction.
- Assume the graph has a cut vertex v.
- Each block is two-connected and must have the property that every pair of vertices has a unique common neighbor.
- So each block is a triangle.
- All components of $G - v$ are Gallai trees.

The Friendship Theorem

If every pair of vertices in a graph have a unique common neighbor, then there is a vertex in the graph that is adjacent to all the other vertices.

Proof Ideas:

- Assume all pairs have a unique common neighbor.

The Friendship Theorem

If every pair of vertices in a graph have a unique common neighbor, then there is a vertex in the graph that is adjacent to all the other vertices.

Proof Ideas:

- Assume all pairs have a unique common neighbor.
- Then the graph is eulerian, outerplanar, and a Gallai-tree.

The Friendship Theorem

If every pair of vertices in a graph have a unique common neighbor, then there is a vertex in the graph that is adjacent to all the other vertices.

Proof Ideas:

- Assume all pairs have a unique common neighbor.
- Then the graph is eulerian, outerplanar, and a Gallai-tree.
- If its two-connected, its a triangle.

The Friendship Theorem

If every pair of vertices in a graph have a unique common neighbor, then there is a vertex in the graph that is adjacent to all the other vertices.

Proof Ideas:

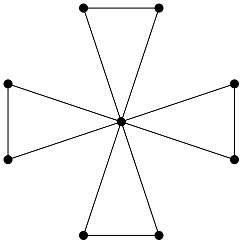
- Assume all pairs have a unique common neighbor.
- Then the graph is eulerian, outerplanar, and a Gallai-tree.
- If its two-connected, its a triangle.
- If it has two or more cut-vertices, diameter is at least 3, violating the common neighbor condition.

The Friendship Theorem

If every pair of vertices in a graph have a unique common neighbor, then there is a vertex in the graph that is adjacent to all the other vertices.

Proof Ideas:

- Assume all pairs have a unique common neighbor.
- Then the graph is eulerian, outerplanar, and a Gallai-tree.
- If its two-connected, its a triangle.
- If it has two or more cut-vertices, diameter is at least 3, violating the common neighbor condition.
- So it has at most one cut vertex, and all blocks are triangles.



Using Conjectures to Investigate Lemma 2

Investigate:

`(pairs_have_unique_common_neighbor)->(is_circular_planar)`

Investigation:

Using Conjectures to Investigate Lemma 2

Investigate:

```
(pairs_have_unique_common_neighbor)->(is_circular_planar)
```

Investigation:

```
property = pairs_have_unique_common_neighbor

theory = [is_circular_planar]
```

Using Conjectures to Investigate Lemma 2

Investigate:

```
(pairs_have_unique_common_neighbor)->(is_circular_planar)
```

Investigation:

```
property = pairs_have_unique_common_neighbor

theory = [is_circular_planar]
```

Conjectures:

```
(pairs_have_unique_common_neighbor)->
    ((is_regular)->(is_planar_transitive))
(pairs_have_unique_common_neighbor)->(is_interval)
(pairs_have_unique_common_neighbor)->(is_factor_critical)
(pairs_have_unique_common_neighbor)->(is_kite_free)
```

New Project—Graph Theory

164 graphs with precomputed data

New Project—Graph Theory

164 graphs with precomputed data

145 graphs with some missing data

New Project—Graph Theory

164 graphs with precomputed data

145 graphs with some missing data

87 properties

New Project—Graph Theory

164 graphs with precomputed data

145 graphs with some missing data

87 properties

78 invariants

New Project—Graph Theory

164 graphs with precomputed data

145 graphs with some missing data

87 properties

78 invariants

Open-source, on GitHub—anyone can use these definitions or add to them.

Human's Can't Make Better Conjectures

There are no simpler statements that are true and significant.

The Dream & Kiran

Code all published graph theory concepts and examples.

The Dream & Kiran

Code all published graph theory concepts and examples.

This will be a huge project.

The Dream & Kiran

Code all published graph theory concepts and examples.

This will be a huge project.

What you'd get:

The Dream & Kiran

Code all published graph theory concepts and examples.

This will be a huge project.

What you'd get:

- Conjectures that are true for all published examples.

The Dream & Kiran

Code all published graph theory concepts and examples.

This will be a huge project.

What you'd get:

- Conjectures that are true for all published examples.
- The simplest conjectures using published concepts.

The Dream & Kiran

Code all published graph theory concepts and examples.

This will be a huge project.

What you'd get:

- Conjectures that are true for all published examples.
- The simplest conjectures using published concepts.

Kiran would have liked that.

Thank You!

Automated Conjecturing in Sage:
http://nvcleemp.github.io/conjecturing/

C. E. Larson and N. Van Cleemput, Automated Conjecturing I: Fajtlowicz's Dalmatian Heuristic Revisited, Artificial Intelligence 231 (2016) 17-38.

clarson@vcu.edu