LARSON—INFO 790–CLASSROOM WORKSHEET 04
Using CONJECTURING on CoCalc

1. Log in to CoCalc.
   (a) Start the Chrome browser.
   (b) Go to `https://cocalc.com`
   (c) Login (**your VCU email address** is probably your username).
   (d) You should see an existing Project for our class. Click on that.
   (e) Click "New", then type **790-c04** into the box, and click "Jupyter Notebook".
   (f) When your notebook opens look on the upper-right to make sure the SageMath kernel is running.

2. Load "conjecturing.py" by running `load("conjecturing.py")`. (Your "790-c04.ipynb" must be in your Home/root directory for this to work, and there should be an "expressions" and "conjecturing.py" file there too).

   Gravity Example Revisited

   In the paper you read last week, one of the examples was about using the CONJECTURING program for how Newton's Law of Gravitation ($F = g\frac{m_1 m_2}{r^2}$, where m_1 and m_2 are the masses of two bodies, r is the distance between them, g is the gravitational constant, and F is the force between the bodies). We can run this experiment ourselves.

3. You should have file "gravity_example.sage" in your CoCalc Project's Home/root directory from last week.

4. We will shortly run that script. Open that file first (by clicking on "gravity_example.sage" from your Home directory file list) and we'll talk about the code.

5. Find the call to the `Conjecturing` program. What are the *objects*? What is their *type*? What are the *methods* available for those objects?

6. How can we get an example of one of the objects? (Let `example = train_pairs[0]`).

7. In your worksheet, run `load("gravity_example.sage")`. You should get a number of conjectures. Do you see anything that looks like Newton's Law?

8. How can we get an example of one of the conjectures? (Let `conj = pair_above_conjs[0]`).

9. How can we evaluate conjecture `conj` for our *example*? (Run: `conj.evaluate(example)`). This will be True or False depending on whether the example is true for the conjectured bound.

   Later we'll see that these *invariant conjectures* (which are in fact properties) can be used as properties for property conjectures (they will give us new novel properties besides properties/booleans/classes that are given with a specific problem).

10. What is we want just the *value* the conjectured bound gives for a specific object? (Run: `conj[0].evaluate(example, returnBoundValue=True)`).

    Real Estate Example

    This example is also in the Brooks, et al. paper. The significance of this example is that here we will *use* invariant-conjectures—which are true or false for every input example—*as* properties themselves.

    Here we'll need an updated version of the "conjecturing.py" file (with functions added, for instance, to handle using conjectures as properties and printing them; there are under-the-hood details that must be addressed). So, for this, we'll need 4 files from your CoCalc Handouts folder.

11. Copy "conjecturing.py", "trainData,csv", "testData.csv", and "real_estate_python3.py" from your CoCalc Handouts folder to your Home/root directory.

12. Click on "real_estate_python3.py". Let's talk about the code there. How is it organized? What will happen?

13. In your worksheet, run `load("real_estate_python3.py")`. You should get a number of conjectures.