1 What is Logico?

Logico is a Python implementation of the robot named Logico that interperets categorical propositions and categorical syllogisms. This is exactly what the Python implementation does.

2 Simple Stuff and Limitations

- However, the Python implementation cannot describe truth. For example, it cannot tell you whether or not all cats are dogs; you must tell it that. In a sense, it is not telling you whether 3 is a square root, but telling you what 2+3 is instead. So the Logico Python implementation can tell validity, but not truth. It can tell when a syllogism is invalid.

Here is the method you would take to make a proposition in general:

```
>>> A=Propos(subject,predicate,lettertype,truth_value)
```

For instance,

```
>>> A=Propos("subject","dogs","A",True)
```

The quotation marks **are needed**. And a syllogism can be completed using this command (again, in general):

```
>>> S=Syllog(A,B,C)
```

where A, B, and C are already-defined propositions.

There are many operations you can peform from here, but most of them have this general syntax.

```
A.propos_method()
```

where A is an already-defined propositon. Notice the ()'s. Likewise, a syllogism method can be called like so:

```
S.syllog_method()
```

where S is an already-defined syllogism.

To find the list of all of the operations, type `help(Propos)` for the propositions and `help(Syllog)` for the syllogisms.

3 Help Text

```
Help on class Propos in module __main__:

class Propos(builtins.object)
 |  Class for a categorical proposition from the Aristotelian standpoint.
 |
 |  Methods defined here:
 |
 |  __init__(self, sub, pred, lettype, tVal)
 |      __init__(str,str,str,bool) -> Propos
 |      Propositon constructor. Takes the subject, predicate, categorical proposition type,
 |  and truth value.
 |
 |  __str__(self)
 |      Propos.__str__() -> str
 |      Class printer. Will put "F" in front of the proposition if it is false.
 |
 |  chgQuality(self)
 |      Propos.chgQuality() -> None
 |      Changes the quality of the proposition.
 |
 |  chgQuantity(self)
 |      Propos.chgQuantity() -> None
 |      Changes the quantifier of the proposition.
 |
 |  contradictory(self)
 |      Propos.chgQuality() -> None
 |      Performs contradiction on the propositon.
 |
 |  contrapose(self)
 |      Propos.contrapose() -> None
 |      Performs contradiction on the propositon.
 |      If proposition does not satisfy the conditions, returns "Illicit contraposition."
 |
 |  contrary(self)
 |      Propos.contrary() -> None
 |      Performs Aristotelian contrary on the propositon.
 |      If proposition does not satisfy the conditions, returns "Illicit contrary."
 |
 |  convert(self)
 |      Propos.convert() -> None
 |      Performs conversion on the propositon.
 |      If proposition does not satisfy the conditions, returns "Illicit conversion."
 |
 |  getEngName(self)
 |      Propos.getEngName() -> str
```

```
 |      Same as __str__, but ignores truth.
 |
 |  getQuality(self)
 |      Propos.getQuality() -> str
 |      Prints quality of proposition.
 |
 |  getQuantity(self)
 |      Propos.getQuantity() -> str
 |      Prints quantity of proposition.
 |
 |  getTermsDist(self)
 |      Propos.getTermsDist() -> list
 |      Lists the terms distributed by the propositon.
 |
 |  obvert(self)
 |      Propos.obvert() -> None
 |      Performs obversion on the propositon.
 |
 |  subalt(self)
 |      Propos.subalt() -> None
 |      Performs Aristotelian subalternation on the propositon.
 |      If proposition does not satisfy the conditions, returns "Illicit subalternation."
 |
 |  subcontrary(self)
 |      Propos.subcontrary() -> None
 |      Performs Aristotelian subcontrary on the propositon.
 |      If proposition does not satisfy the conditions, returns "Illicit subcontrary."
 |
 |  ----------------------------------------------------------------------
Help on class Syllog in module __main__:

class Syllog(builtins.object)
 |  A class for a categorical syllogism.
 |
 |  Methods defined here:
 |
 |  __init__(self, majorPrem, minorPrem, conclus)
 |      __init__(Propos,Propos,Propos) -> Syllog
 |      Class constructor. Takes three propositions as input.
 |
 |  __str__(self)
 |      Propos.__str__() -> str
 |      Class printer. Bar adjusts itself to length of the longest
 |      propositions. Also ignores putting F for false propositions.
 |
 |  getFig(self)
 |      Propos.getFig() -> int
 |      Gives the figure of the syllogism as an Int.
 |
 |  getMiddleTerm(self)
 |      Propos.getMiddleTerm() -> str
 |      Gives the middle term of the syllogism.
 |
 |  getMood(self)
 |      Propos.getMood() -> str
 |      Gives the mood of the syllogism.
 |
 |  isValid(self)
 |      Propos.getFig() -> bool
 |      Tests for validity. IT DOES NOT TEST FOR SOUNDNESS.
 |
 |  ----------------------------------------------------------------------
```

4 Downloading Python for your Computer

Please visit `https://www.python.org/downloads/`. Then click "Download Python 3.4.1."