COMP 410
Fall 2023

Nondeterminism in Python

1.) Consider the following Prolog procedure:

```
myNumber(0).
myNumber(1).
myNumber(2).
```

Write an equivalent generator function in Python, with the name `myNumber`.

2.) Consider the following Prolog procedure, which uses myNumber:

```
makePair(pair(A, B)) :-
  myNumber(A),
  myNumber(B).
```

Write an equivalent generator function in Python, with the name `makePair`. It should generate Python pairs instead. For example, the Prolog `pair(1, 2)` is equivalent to the Python `(1, 2)`.

3.) Consider the following Prolog procedure, representing a generator for an AST:

```
gen(_, variable(x)).
gen(_, variable(y)).
gen(Depth, call(E1, E2)) :-
  Depth > 0,
  NewDepth is Depth - 1,
  gen(NewDepth, E1),
  gen(NewDepth, E2).
```

Assume we have the following Python code defined, representing the AST:

```
class Variable:
  def __init__(self, name):
    self.name = name

class Call:
  def __init__(self, e1, e2):
    self.e1 = e1
    self.e2 = e2
```

Write an equivalent generator in Python. You can assume the variable names are strings (e.g., "x", "y").