NOTES ON FUNCTIONS

ERIC MARTIN

1. Default values

Default values work as "as expected" for simple types:

```
>>> def h(a = 0):
        print(a)
        a += 1

>>> h()
0
>>> h(1)
1
>>> h()
0
```

But more complex types can cause surprises, as the default value is memorised after it has been initialised at the first function call:

```
>>> def f(a = []):
    a.append(1)
    print(a)

>>> f()
[1]
>>> f([0])
[0, 1]
>>> f()
[1, 1]
```

It can be worked around as follows:

```
>>> def g(a = None):
    if a == None:
        a = []
    a.append(1)
    print(a)

>>> g()
[1]
>>> g([0])
[0, 1]
>>> g()
[1]
```

Date: Session 1, 2015.

2. Documentation and testing

Docstrings provide a convenient way to document a function (or a module, or a class) and easily perform *unit testing*.

For instance, if the program `prolog_interpreter.py` is running, then we can execute:

```
>>> help(unify)
Help on function unify in module __main__:

unify(expression_1, expression_2, unification=None)
    Unifies two atoms or terms at most one of which contains variables
    (that is, capitalised terms, but contrary to standard Prolog underscores are not allowed).
    The third argument is used to possibly record the assignment of some terms to some variables,
    which has to be consistently extended; it is what is eventually returned.

    >>> unify('a','a')
    {}

    >>> unify('a', 'b')

    >>> unify('a','X')
    {'X': 'a'}

    >>> d = unify('f(a,b,b,a)', 'f(X,Y,Z,X)'); sorted([(x, d[x]) for x in d])
    [('X', 'a'), ('Y', 'b'), ('Z', 'b')]

    >>> d = unify('f(a,f(a,b,g(c,d),f(a,g(c,d))))', 'f(X,f(a,Y,U,f(X,g(V,d))))'); sorted([(x, d[x]) for x in d])
    [('U', 'g(c,d)'), ('V', 'c'), ('X', 'a'), ('Y', 'b')]

    >>> unify('f(a,f(a,b,g(c,d),f(a,g(c,d))))', 'f(X,f(a,Y,U,f(X,g(Y,d))))')
```

The tests can be run from the shell with the following command:

```
python –m doctest prolog_interpreter.py
```

As no test fails, the command exists silently, but the `-v` option provides a verbose output:

```
$ python –m doctest –v prolog_interpreter.py
Trying:
    program = {'a(X)' : ['b', 'c(X)', 'd(X)'],
        'b' : [],
        'c(X)' : ['b'],
        'd(X)' : ['b', 'e(X)', 'f'],
        'e(X)' : [],
        'f' : ['c(X)'],
        'g(X)' : ['a(X)', 'h']}
Expecting nothing
ok
Trying:
    answer_query('a(0)', program)
Expecting:
    True

...

Trying:
    d = unify('f(a,b,b,a)', 'f(X,Y,Z,X)'); sorted([(x, d[x]) for x in d])
Expecting:
    [('X', 'a'), ('Y', 'b'), ('Z', 'b')]
ok
Trying:
    d = unify('f(a,f(a,b,g(c,d),f(a,g(c,d))))', 'f(X,f(a,Y,U,f(X,g(V,d))))'); sorted([(x, d[x]) for x in d])
Expecting:
    [('U', 'g(c,d)'), ('V', 'c'), ('X', 'a'), ('Y', 'b')]
ok
Trying:
    unify('f(a,f(a,b,g(c,d),f(a,g(c,d))))', 'f(X,f(a,Y,U,f(X,g(Y,d))))')
Expecting nothing
ok
4 items had no tests:
    prolog_interpreter
    prolog_interpreter.get_program
    prolog_interpreter.unify_atoms
    prolog_interpreter.unify_variable_with
3 items passed all tests:
   3 tests in prolog_interpreter.answer_query
   2 tests in prolog_interpreter.parse
   6 tests in prolog_interpreter.unify
11 tests in 7 items.
11 passed and 0 failed.
Test passed.
```