

Extra programming exercise for chapter 3

Øyvind Ryan (oyvindry@math.uio.no)

September 11, 2014

In Chapter 3 we encountered three different algorithms for converting numbers to different bases. We also looked at straightforward implementations of these in the lectures, but we have not made reusable implementations of them where the algorithms are defined as functions, and where the numbers to convert and the base for conversion are given as parameters. You will be asked to implement the following functions:

1. A function `digits(a, beta)`, which uses algorithm 3.7 to compute the digits of the number `a` in base `beta`.
2. A function `decimal_digits_fractional(a, beta, n)`, which uses algorithm 3.16 to compute the first `n` digits of the fractional number `a` in base `beta`.
3. A function `decimal_digits_rational(b, c, beta)` which uses algorithm 3.20 to compute the digits of the rational number `b/c` in base `beta`.

You should put these functions in a module called `baseconversion` (i.e. as functions in a file called `baseconversion.py`). A precise documentation for the three functions you shall define can be found in <http://folk.uio.no/oyvindry/matinf1100/python/chap3/baseconversion.html>.

Some parts here may require some more explanation. For instance, for the function `decimal_digits_rational`, algorithm 3.20 says nothing about when the algorithm should terminate. There are two things in this respect which your function should take into account:

1. When `b` in algorithm 3.20 is set to 0, the algorithm should terminate.
2. When we obtain a value for `b` which equals a previous value for `b`, we should terminate. The reason is that then we have obtained a repeating sequence. In order to detect that `b` has obtained a previous value, you need to store previous values of `b` in a list. If you call this list `bs`, you can check if a new value of `b` is in this list by writing `"if b in bs:"`. You can also find the position of `b` in `bs` by writing `bs.index(b)`. This is necessary in order to find the exact position for where the repeating sequence starts the first time.

When you have made your own implementation of these functions, you can compare with a suggested implementation which you can find in <http://folk.uio.no/oyvindry/matinf1100/python/chap3/baseconversion>.

py.

You can also test your functions by comparing with computations you have done by hand in the exercises in chapter 3. We have tested the suggested implementation of `baseconversion.py` on exercises 3.2.2, 3.3.3, and 3.3.4, these are found in

<http://folk.uio.no/oyvindry/matinf1100/python/chap3/>.

These tests also use a utility function called `string_from_digits`, which prints the digits in a presentable manner. You do not need to implement this function, it can be found in the suggested implementation of `baseconversion.py`.