Computer Systems Modeling and Verification (USEEN1)

Dictionaries

Exercises taken from the *Python Programming Primer*[1] (and updated with type hints).

Textbook *Introduction to Python for Computational Science and Engineering* (2022), Chapter 3.

Corresponding lecture slides from *Computational Science and Engineering in Python*:

– Dictionaries

Tutorial. Define the following functions.

1. A function `list_to_dict[K, V](l: list[tuple[K, V]]) -> dict[K, V]` which converts a list of pairs into a dictionary.

2. A function `dict_to_list[K, V](d: dict[K, V]) -> list[tuple[K, V]]` which converts a dictionary into a list of pairs.

Exercices. Define the following functions.

1. A function `reverse_dict[K, V](d: dict[K, V]) -> dict[V, K]` that takes a dictionary `d` as the input argument and returns a dictionary `r`. If the dictionary `d` has a key `k` and an associated value `v`, then the dictionary `r` should have a key `v` and a value `k`. Not that this function is only expected to work for dictionaries where each value occurs at most once (although you do not need to check for this).

2. A function `count_chars(s: str) -> dict[str, int]` which takes a string `s` and returns a dictionary. The dictionary's keys are the set of characters that occur in string `s`. The value for each key is the number of times that this character occurs in the string `s`.

   Examples:

   ```
   >>> count_chars("x")
   {'x': 1}

   >>> count_chars("xxx")
   {'x': 3}

   >>> count_chars("xxxyz")
   {'x': 3, 'y': 1, 'z': 1}

   >>> count_chars("Hello␣World")
   {'H': 1, 'e': 1, 'l': 3, 'o': 2, '␣': 1, 'W': 1, 'r': 1, 'd': 1}
   ```

   Note that the order in which the key-value pairs are listed in the output dictionary is not important.

1. *Python Programming Primer*, Hans Fangohr *et al.* University of Southampton (2016)