Informatik I: Einführung in die Programmierung

Prof. Dr. Peter Thiemann
Dr. Daniel Büscher, Hannes Saffrich
Wintersemester 2019

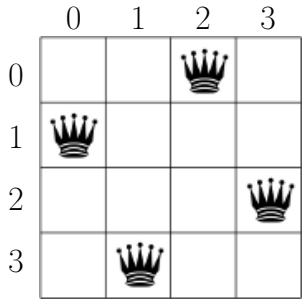
Universität Freiburg
Institut für Informatik

Übungsblatt 13 – Lösungen

Abgabe: Montag, 03.02.2020, 9:00 Uhr morgens

Aufgabe 13.1 (n-Damenproblem; Datei: `queens.py`; Punkte: 11)

Im n-Damenproblem sollen n Damen auf einem $n \times n$ großen Schachbrett so aufgestellt werden, dass keine zwei Damen auf derselben Reihe, Linie oder Diagonale stehen. Die folgende Abbildung zeigt eine gültige Lösung des 4-Damenproblems:



Schreiben Sie einen Generator `queens(n: int)`, welcher alle gültigen Belegungen des n-Damenproblems mittels *Backtracking* erzeugt. Jede Belegung soll dabei durch ein Tupel der Länge n repräsentiert werden, wobei der i-te Eintrag im Tupel die Zeilenposition der Dame aus Spalte i beschreibt. Die Damenbelegung der obigen Abbildung würde somit durch das Tupel `(1, 3, 0, 2)` beschrieben. Beispiel:

```
>>> for a in queens(4):
...     print(a)
...
(1, 3, 0, 2)
(2, 0, 3, 1)
>>> queens8 = queens(8)
>>> next(queens8)
(0, 4, 7, 5, 2, 6, 1, 3)
>>> next(queens8)
(0, 5, 7, 2, 6, 3, 1, 4)
```

Hinweis: 1-Tupel können mit `(elem,)` geschrieben werden und Tupel können analog zu Listen mit `+` konkateniert werden. Beispiel:

```
>>> (1, 2, 3) + (4,)
(1, 2, 3, 4)
```

Lösung:

```
 def queens(n: int):
    """Returns a generator of solutions for the `n`-queens problem.

    A solution is modeled as a `n`-tuple of integers, where the `i`-th entry of
    the tuple describes on which row the queen at column `i` should be placed.
    """
    return queens_recursive(tuple(), n)

def queens_recursive(t: tuple, n: int):
    """Helper function for `queens`."""
    if len(t) == n:
        yield t
    else:
        for row in range(n):
            t1 = t + (row,)
            if new_queen_is_valid(t1):
                yield from queens_recursive(t1, n)

def new_queen_is_valid(t: tuple) -> bool:
    """Returns `True` iff the queen on the last column of `t` is not in
    conflict with the queens on previous columns of `t`.
    """
    last_col = len(t)-1
    for col in range(last_col):
        if t[last_col] == t[col]:                    # horizontal conflict
            return False
        if t[last_col] == t[col] + (last_col - col): # diagonal conflict 1
            return False
        if t[last_col] == t[col] - (last_col - col): # diagonal conflict 2
            return False
    return True
```

Aufgabe 13.2 (Fgrep; Datei: `fgrep.py`; Punkte: 3+2+2)

In der Vorlesung wurde die Funktion `fgrep` vorgestellt, welche (wie das gleichnamige Unix-Kommando) Dateien auf bestimmte Zeichenketten hin durchsucht.

(a) Implementieren Sie einen Generator `fgrep(subject: str, filename: str)`, welcher die Datei mit Namen `filename` durchläuft und alle Tupel `(n, line)` generiert, für die gilt, dass die Zeile `line` die Zeichenkette `subject` enthält und `n` die Zeilennummer von `line` ist.

(b) Erweitern Sie die Funktion `fgrep` um ein Argument `v` vom Typ `bool`. Ist `v` True, so soll die Suche invertiert werden, d.h. es werden alle Tupel `(n, line)` generiert, für welche gilt, dass die Zeile `line` den String `subject` nicht enthält.

(c) Erweitern Sie die Funktion `fgrep` um ein weiteres Argument `i` vom Typ `bool`. Ist `i` True, so sollen alle Tupel `(n, line)` generiert werden, dessen Zeilen den String `subject` ohne Berücksichtigung der Groß- und Kleinschreibung enthalten (falls `v` False ist) bzw. nicht enthalten (falls `v` True ist).

Lösung:

```
"""
Author: Frank Schüssele
"""


def fgrep(subject: str, filename: str, v: bool, i: bool):
    """A generator that yields specific lines in the given file.

    Args:
        subject (str): The subject that should be checked for the lines.
        filename (str): The filename of the file to be checked.
        v (bool): If False, all lines that contain subject are yielded.
                  If True, all lines that do not contain subject are yielded.
        i (bool): If True, the case of subject is ignored.
    """
    with open(filename) as f:
        for n, line in enumerate(f):
            c = subject.lower() in line.lower() if i else subject in line
            if c != v:
                yield (n, line)
```

Aufgabe 13.3 (Erfahrungen; Datei: `erfahrungen.txt`; Punkte: 2)

Legen Sie im Unterverzeichnis `sheet13` eine Textdatei `erfahrungen.txt` an. Notieren Sie in dieser Datei kurz Ihre Erfahrungen beim Bearbeiten der Übungsaufgaben (Probleme, Bezug zur Vorlesung, Interessantes, benötigter Zeitaufwand, etc.).