Computeralgebra-Praktikum

Universität Siegen Mohamed Barakat, Tommy Hofmann

Aufgabe 3 (Fortsetzung von Aufgabe 1)**.** Schreibe vier Methoden

- `is_well_defined`;
- `is_injective`;
- `is_surjective`;
- `is_bijective`.

Alle vier Methoden sollen `true` oder `false` zurückgeben. Die erste Prozedur testet ob die Datenstruktur einer Abbildung korrekt ist. Die zweite ob die Abbildung injektiv, die dritte ob die Abbildung surjektiv ist und die vierte ob die Abbildung bijektive ist.
Hinweis: `length`, `unique`, &&, `issubset`.
Hier ein paar Testbeispiele:

```
julia> f = (2, [1, 2], 1);

julia> is_well_defined(f)
false

julia> f = (2, [1, 3], 3);

julia> is_well_defined(f)
true

julia> is_injective(f)
true

julia> is_surjective(f)
false

julia> f = (3, [2, 1, 2], 2);

julia> is_injective(f)
false

julia> is_surjective(f)
true

julia> is_bijective((2, [2, 1], 2))
true
```

Aufgabe 4. Sei $R \in \{\mathbb{Q}, \mathbb{F}_5, \mathbb{Z}, \mathbb{Q}[x], \mathbb{F}_5[y]\}$. Programmiere eine Julia-Funktion

`fully_divide_column_trafo(A)`,

die bei Eingabe einer Spalte $A \in R^{n\times 1}$ eine Matrix $U \in \mathrm{GL}_n(R)$ zurückgibt, mit

$$UA = \begin{pmatrix} d \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

mit $d = 1$ falls $R = \mathbb{Q}$ bzw. $d = \mathrm{ggT}(A_{11}, \ldots, A_{n1})$ falls $R = \mathbb{Z}$ oder $R = \mathbb{Q}[x]$.
Hinweis: Benutze dafür die bereits programmierte Funktion `fully_divide_pair_trafo`, sowie die Befehle `nrows`, den Teilmatrixkonstruktor `A[1:n, 1:m]` und die Prozedur:

```
function fully_divide_pair_trafo_inflated(a, b, i, j, n)
  R = parent(a)
  U = identity_matrix(R, n)
  u = fully_divide_pair_trafo(a, b)

  U[i, i] = u[1, 1]
  U[i, j] = u[1, 2]
  U[j, i] = u[2, 1]
  U[j, j] = u[2, 2]

  return U
end
```

Vgl. Algorithmus 2.3.6 im LA I Skript:

`https://algebra.mathematik.uni-siegen.de/barakat/Lehre/WS20/LAI/Skript/LA_I.pdf`