

A4: Relationen und Abbildungen

Vorweg: In dem Verzeichnis dieser Aufgabe in meinem [pub](#) liegen die Vorlesungsskripte [relationen.pdf](#) und [abbildungen.pdf](#) von Frau Prof. Padberg, in denen Sie die mathematischen Definitionen nochmals nachschlagen können.

Relationen (zweistellige Relationen)

- Schreiben Sie die Klasse [RelationenGenerator](#) und eine Klassenmethode [generiere_relation\(A,B,k\)](#), die aus zwei übergebenen Mengen [A](#), [B](#) eine Relation der Mächtigkeit [k](#) erzeugt. Die Klassenmethode

[def RelationenGenertor.generiere_relation\(A,B,k\) ... end](#)

arbeitet wie folgt:

- Sie wählt zufällig aus der Menge [A](#) ein Element aus und ordnet diesem Element ein Element der Menge [B](#) zu, das wiederum zufällig gewählt wird. Verwenden Sie zwei Zufallsgeneratoren. Die Zuordnung soll als Instanz der Klasse [Tupel](#) repräsentiert werden. Jedes [Tupel](#) wird in die zu erzeugende Relation eingefügt. Wenn die [Relation](#) die Grösse [k](#) hat, gibt die Methode die Relation als Ergebnis zurück.
- Ein [Relationen](#)-Objekt ist eine Instanz der Klasse [Relation](#). Beim Einfügen von [Tupeln](#) muss darauf geachtet werden, dass keine Dubletten eingefügt werden.

Überlegen Sie, wie Sie die Relation in Ruby geeignet repräsentieren. Die Repräsentation sollte dabei der mathematischen Definition maximal entsprechen. Bereiten Sie die Klasse [Tupel](#) für das korrekte Einfügen in eine Relation geeignet vor.

Schreiben Sie die Methoden [add\(a_tupel\)](#) und [size\(\)](#) für die Klasse [Relation](#) und nutzen Sie diese Methoden für die Lösung in 1.)

- Machen Sie die Klasse [Relation](#) enumerierbar, implementieren Sie einen Iterator für die Klasse und implementieren Sie die nachfolgenden Methoden der Klasse mit Hilfe geeigneter Methoden des Moduls [Enumerable](#).
- Erzeugen Sie Relationen $R \subseteq A \times A$ und prüfen Sie für diese Relationen, ob sie die Eigenschaften reflexiv, symmetrisch, asymmetrisch, anti-symmetrisch und transitiv erfüllen. Schreiben Sie für die Überprüfung der Eigenschaften die Objektmethoden [reflexiv?\(\)](#), [symmetrisch?\(\)](#), [asymmetrisch?\(\)](#), [anti_symmetrisch?\(\)](#) und [transitiv?\(\)](#) der Klasse [Relation](#).
- Erzeugen Sie Relationen $R \subseteq A \times B$ und prüfen Sie für diese Relationen, ob sie die Eigenschaften rechts-, links- eindeutig, rechts- total erfüllen. Schreiben Sie für die Überprüfung der Eigenschaften die Objektmethoden [rechts_eindeutig?\(\)](#), [links_eindeutig?\(\)](#), [rechts_total?\(\)](#) und [links_total?\(\)](#) der Klasse [Relation](#).
- Schreiben Sie eine Objektmethode [verknuepfe\(andere_relation\)](#), die eine Relation $R_1 \subseteq A \times B$ mit einer Relation $R_2 \subseteq B \times C$ verknüpft und als Ergebnis eine neue Relation zurückgibt. Die Methode arbeitet nicht destruktiv.
- Berechnen Sie für eine Relation [R](#) den reflexiven, den symmetrischen und den transitiven Abschluss. Schreiben Sie dazu nicht destruktive Objektmethoden [reflexiver_abschluss\(\)](#), [symmetrischer_abschluss\(\)](#) und [transitiver_abschluss\(\)](#) der Klasse [Relation](#), die als Ergebnis eine Relation zurückliefern.
 - Schreiben Sie für alle diese Methoden eine Hilfsmethode [kopiere\(\)](#) in der Klasse [Relation](#), die eine neue Relation erzeugt und dabei die Instanzvariablen der Relation kopiert.
 - Lösen Sie die Berechnung des transitiven Abschlusses rekursiv.

8. Schreiben Sie die Objektmethode `invertiere()` der Klasse `Relation`, die zu einer Relation R die inverse Relation R^{-1} berechnet. Die Methode arbeitet nicht destruktiv.

9. Hinweise:

- Klassenmethoden können direkt auf der Klasse aufgerufen werden. Also durch `RelationGenerator.generiere_relation(A,B,k)`
Es wird vorausgesetzt, dass A,B, und k mit sinnvollen Werten initialisiert sind.
- Verwenden Sie wann immer möglich geeignete Methoden von `Enumerable`. Machen Sie die Klasse `Relation` enumerierbar.
- Externer Zugriff auf die Instanzvariable, in der die Tupel der Relation verwaltet werden, ist nicht erlaubt (keine sondierende Methode, kein Reader für diese Instanzvariable).

Abbildungen

- Schreiben Sie eine Klassenmethode `generiere_abbildung(A,B)` der Klasse `RelationenGenerator`, die aus zwei übergebenen Mengen A, B eine Abbildung erzeugt. Wählen Sie die Elemente des Wertebereichs zufällig aus.
`def RelationenGenerator.erzeuge_abbildung(A,B) ... end`
- Prüfen Sie, ob die Methode korrekt arbeitet, indem Sie die Minimal-Eigenschaften für Abbildungen auf der erzeugten Relation prüfen. Schreiben Sie dazu die Objekt-Methode `abbildung?()` der Klasse `Relation`.
- Prüfen Sie, ob die Abbildung, injektiv oder surjektiv ist. Schreiben Sie dazu die Objektmethoden `injektiv?()`, `surjektiv?()` für die Klasse `Relation`.
- Berechnen Sie die Komposition zweier Abbildungen in einer nicht destruktiven Objektmethode der Klasse `Relation`.
- Berechnen Sie für eine Abbildung das Urbild. Prüfen Sie dazu zunächst, ob die Relation eine Abbildung ist. Schreiben Sie die Objekt-Methode `urbild()` der Klasse `Relation`.
 - Die Methode arbeitet nicht destruktiv.
 - Sie benötigen für die Darstellung des Definitions- und Wertebereichs die Klasse `Potenzmenge`, diese ist im zip-Paket für Aufgabe 4 enthalten.
 - Prüfen Sie anschließend, ob das Urbild wieder eine Abbildung erzeugt.

Teil B

Schreiben Sie RUnit Tests für die Relationen-Methoden A 1.)-7.)