

Aufgaben-Blatt: Berechnung des kürzesten Weges

Definition: Ein Pfad-Relation PR ist eine Menge von 3-Tupeln der Form $\langle \langle x, y \rangle, p, l \rangle$ so dass gilt:

1. $\langle x, y \rangle$ ist ein Paar von Punkten.
2. p ist eine Liste von Punkten. Der erste Punkt der Liste ist x , der letzte Punkt ist y , in SETL2-Notation gilt also: $x = p(1)$ und $y = p(\#p)$.
Die Liste p wird interpretiert als ein *Pfad*, der von x nach y führt.
3. l ist eine positive natürliche Zahl. Diese Zahl gibt die Länge des Pfades p an.

Die *Komposition* einer Abstands-Funktion D mit einer Pfad-Relation PR kann in der Mengenlehre wie folgt definiert werden:

$$D \circ PR := \left\{ \langle \langle x, z \rangle, [x] + p, D(\langle x, y \rangle) + l \rangle \mid \langle x, y \rangle \in \text{dom}(D) \wedge \langle \langle y, z \rangle, p, l \rangle \in PR \right\}$$

Die Notation $[x] + p$ bezeichnet dabei den Pfad, den wir erhalten, wenn wir den Punkt x vorne an den Pfad p anfügen.

Hinweis: Versuchen Sie bei der Lösung der nachfolgenden Aufgaben möglichst mit Mengen-Konstruktionen und nicht mit Kontroll-Strukturen wie **for** oder **while** zu arbeiten.

Aufgabe 1: Implementieren Sie eine Prozedur **compose**, so dass der Aufruf **compose**(D, PR) für eine Abstands-Funktion D und eine Pfad-Relation PR die Komposition $D \circ PR$ berechnet.

Aufgabe 2: Implementieren Sie eine Prozedur **cyclic**, so dass der Aufruf **cyclic**(p) für einen Pfad p genau dann den Wert **true** zurück liefert, wenn der Pfad p zyklisch ist, das heißt dass die Liste p einen Punkt mehrfach enthält.

Hinweis: Konvertieren Sie die Liste p in eine Menge und überlegen Sie, wie sich die Anzahl der Elemente dieser Menge zu der Länge der Liste verhält.

Aufgabe 3: Implementieren Sie eine Prozedur **eliminateCycles**, so dass der Aufruf
eliminateCycles(PR)

als Ergebnis die Pfad-Relation berechnet, die Sie erhalten, wenn Sie aus der Pfad-Relation PR alle die 3-Tupel $\langle \langle x, y \rangle, p, l \rangle$ entfernen, für die der Pfad p einen Zyklus enthält.

Aufgabe 4: Implementieren Sie eine Prozedur **createPathRelation**, so dass der Aufruf
createPathRelation(D)

aus einer Abstands-Funktion D eine Pfad-Relation erzeugt.

Aufgabe 5: Ändern Sie die in der Vorlesung entwickelte Prozedur **closure** so ab, dass der Aufruf
closure(D)

aus einer gegebenen Abstands-Funktion D eine Pfad-Relation erzeugt, die alle zyklen-freien möglichen Verbindungen zwischen zwei Punkten enthält.

Hinweis: Die in der Vorlesung verwendete Prozedur **minimum** benötigen Sie dazu nicht.

Aufgabe 6: Entwickeln Sie eine Prozedur **minimize**, so dass der Aufruf **minimize**(PR) aus einer gegebenen Pfad-Relation PR alle die 3-Tupel $\langle \langle x, y \rangle, p, l \rangle$ entfernt, für die der Pfad p nicht die kürzest mögliche Verbindung zwischen x und y ist.