52. (6 Punkte) Erweitern Sie die Spezifikation für den abstrakten Datentype *Menge* (von ganzen Zahlen) aus der Vorlesung mit den Operationen Einfügen, Entfernen, Elementtest, und Erstellen einer leeren Menge um das Erstellen eines *Iterators* mit den Operationen *next* und *hasNext* (wie im Java-Interface `Iterator`). Spezifizieren Sie diese Operationen formal.

53. (0 Punkte) Erweitern Sie die algebraische Spezifikation von Mengenoperationen aus Aufgabe 43 um die Funktionen *Vereinigung* und *Durchschnitt* (von je zwei Mengen).

54. (0 Punkte) Definieren Sie in Haskell einen algebraischen Datentyp `Menge a` für Mengen von Elementen des Typs `a`, die wahlweise durch Aufzählen der Elemente oder durch Angabe einer charakteristischen Eigenschaft definiert werden können. Schreiben Sie folgende Funktionen:

```
mengeausListe    :: [a] -> Menge a           -- {x1, x2, ..., xn}
mengefürdiegilt  :: (a -> Bool) -> Menge a   -- { x | f(x) }
mengeDurchschnitt:: Menge a -> Menge a -> Menge a
mengeVereinigung :: Menge a -> Menge a -> Menge a
mengeElementvon  :: a -> Menge a -> Bool     -- x ∈ M
```

55. (0 Punkte) Schreiben Sie eine Spezifikation für *Intervallarithmetik*.
Intervallarithmetik liefert verlässliche Ergebnisse, selbst wenn die Eingabedaten mit Messfehlern behaftet sind und zwischendurch Rechenfehler auftreten. Statt mit Zahlen rechnet man mit Intervallen $[a, b]$, die durch die arithmetischen Grundoperationen, $+$, $-$, $\times$ und $:$ miteinander verknüpft und miteinander verglichen werden können (durch $<$, $>$, $\leq$, usw.) Das Ergebnis ist wieder eine Intervall.

56. (14 Punkte) Graphenalgorithmen in der künstlichen Intelligenz. Die folgende Aufgabe findet sich in einer Sammlung von Rechenaufgaben mit dem Titel *Propositiones ad acuendos juvenes* (Aufgaben zur Schärfung des Geistes der Jugend), die wahrscheinlich um das Jahr 800 am Hof Karls des Großen entstanden ist und Alkuin von York zugeschrieben wird:
Die Aufgabe vom Wolf, der Ziege und dem Kohlkopf. Ein Mann musste einen Wolf, eine Ziege und einen Kohlkopf über einen Fluss übersetzen; er konnte aber nur ein Boot auftreiben, das gerade zwei von ihnen tragen konnte. Wie konnte er alles unversehrt hinüberbringen?[1]

    (a) Modellieren Sie diese Aufgabe durch einen Graphen. Die *Knoten* sollen den möglichen Zuständen des „Systems“ Mann–Ziege–Wolf–Kohlkopf–Boot–Fluss entsprechen, und die *Kanten* den erlaubten Übergängen: Zum Beispiel würde der Wolf die Ziege oder die Ziege den Kohlkopf fressen, wenn sie ohne Aufsichtig gelassen würden; der Mann, der das Boot rudert, kann nur einen Gegenstand oder ein Tier zusätzlich mitnehmen. Eine *Lösung* soll einem *Weg* in diesem Graphen entsprechen.

    (b) Schreiben Sie ein Programm, das diesen Graphen erstellt. Wie viele Knoten und wie viele Kanten hat der Graph? Schreiben Sie eine Programm, das mit Breitensuche einen Weg in diesem Graphen findet, der einer Lösung des Problems entspricht.

    (c) (0 Punkte) Ist die Lösung eindeutig? Wie kann man feststellen, ob es in einem Graphen nur einen einzigen Weg von s nach t gibt?

[1] PROPOSITIO DE LUPO ET CAPRA ET FASCICULO CAULI. Homo quidam debebat ultra fluvium transferre lupum et capram et fasciculum cauli, et non potuit aliam navem invenire, nisi quae duos tantum ex ipsis ferre valebat. Praeceptum itaque ei fuerat, ut omnia haec ultra omnino illaesa transferret. Dicat, qui potest, quomodo eos illaesos ultra transferre potuit. SOLUTIO. Simili namque tenore ducerem prius capram et dimitterem foris lupum et caulum. Tum deinde venirem lupumque ultra transferrem, lupoque foras misso rursus capram navi receptam ultra reducerem, capraque foras missa caulum transveherem ultra, atque iterum remigassem, capramque assumptam ultra duxissem. Sicque faciente facta erit remigatio salubris absque voragine lacerationis.