Ludwig-Maximilians-Universität München
Institut für Informatik
Priv.-Doz. Dr. R. Hennicker
Dr. A. Knapp

SS 2003
Kurzprüfung 2

Lösungsvorschlag zur Kurzprüfung zu Formale objektorientierte Software-Entwicklung

Aufgabe 1

1. Mit $\Delta = \mathsf{BankAccounts}$ ist

$$\begin{aligned} &PreSat_I^{\Delta}(AddInterest) = \\ &\quad \{(\sigma, o.\mathsf{addInterest}()) \in State_\Delta \times Label_\Delta \mid \\ &\qquad o \in \mathsf{SavingsAccount}_\sigma \wedge o.\mathsf{interestRate}_\sigma \geq 0 \wedge o.\mathsf{balance}_\sigma \geq 0\} \end{aligned}$$

2. Da *AddInterest* wohlgeformt ist, gilt insbesondere

$$\mathrm{dom}([\![AddInterest]\!]_I^{\mathsf{BankAccounts}}) = PreSat_I^{\mathsf{BankAccounts}}(AddInterest)\ .$$

Aufgabe 2

Als Operationsspezifikation *AccountWithdraw* kann gewählt werden:

```
context Account::withdraw(a : Real)
pre: a >= 0 and balance-a >= -limit
post: balance = balance@pre-a
```

Die Vorbedingung ist etwa mit $\mathtt{a} = 0$, $\mathsf{balance} = 1$, $\mathsf{limit} = 0$ erfüllbar. Ist die Vorbedingung erfüllt, so etabliert die Zuweisung von $\mathtt{balance@pre} - \mathtt{a}$ an `balance` die Nachbedingung.

Aufgabe 3

1. Die Normalform für die Menge der beiden Operationsspezifikationen im Kontext der Klasse SavingsAccount ist:

```
context SavingsAccount::changeLimit(l : Real)
pre: (0 <= l and -l < balance) or (limit = 0)
post: ((0 <= l and -l < balance@pre) implies (limit = l)) and
      ((limit@pre = 0) implies (limit = limit@pre))
```

2. Die Menge der beiden Operationsspezifikationen ist bezüglich I nicht wohlgeformt, da etwa für $\mathsf{limit} = 0$ und Argumentwert $\mathtt{l} = 1$ zum Aufrufzeitpunkt beide Vorbedingungen erfüllt sind, die Nachbedingungen $\mathsf{limit} = 1$ und $\mathsf{limit} = \mathsf{limit@pre} = 0$ aber nicht gleichzeitig etabliert werden können.