

Lösungsvorschlag zur Kurzprüfung zu Formale objektorientierte Software-Entwicklung

Aufgabe 1

1. Die Klassensorten sowie die daraus abgeleiteten Kollektionssorten sind gegeben durch $C \cup \overline{C}$ mit

$$\begin{aligned} C &= \{\text{Entry, File, Folder}\} \quad \text{und} \\ \overline{C} &= \{\overline{\sigma}(\gamma) \mid \sigma \in S \cup \{\text{Collection}\}, \gamma \in C\} \quad \text{wobei} \\ S &= \{\text{Set, Sequence, Bag}\} \end{aligned}$$

Die Untersortenbeziehung \leq auf der Menge der Klassensorten und den daraus abgeleiteten Kollektionssorten ergibt sich mit den Bezeichnungen wie eben wie folgt:
Sei

$$\leq_C = \{(\text{File, Entry}), (\text{Folder, Entry})\}$$

Dann ist \leq die transitive Hülle von

$$\begin{aligned} &\{(\overline{\gamma}, \overline{\gamma}) \mid \gamma \in C \cup \overline{C}\} \cup \\ &\{(\sigma(\gamma), \sigma(\gamma')) \mid \sigma \in S, (\gamma, \gamma') \in \leq_C\} \cup \\ &\{(\sigma(\gamma), \text{Collection}(\gamma)) \mid \sigma \in S\} \end{aligned}$$

2. Mit den Voraussetzungen der Aufgabenstellung ergibt sich:

- (a) `Folder.allInstances()->union(File.allInstances())`
hat den kleinsten Typ `Set(Entry)`.
- (b) `g.contents->iterate(e : Entry; s : Integer = 0 |
if e.oclIsTypeOf(File) then s+e.size else s+e.size() endif)`
ist nicht wohlgeformt, da `Entry` kein Attribut `size` hat.
- (c) `g.contents.size() > 0 and g.size()/g.contents.size() < 0`
ist nicht wohlgeformt, da `g.contents : Set(Entry)` und `_.size()` auf `Set(Entry)` nicht definiert ist.

- (d) `g.contents@pre->exists(e : Entry | e <> null)`
hat den kleinsten Typ `Boolean`.
- (e) `g.contents->select(e : Entry | e.oclIsTypeOf(File))`
hat den kleinsten Typ `Set(Entry)`.
- (f) `g.size() = g`
hat den kleinsten Typ `Boolean`.

Aufgabe 2

Die semantische Funktion für die strikte OCL-Multiplikation ist gegeben durch:

$$\llbracket _ * _ \rrbracket : \llbracket \text{Integer} \rrbracket \times \llbracket \text{Integer} \rrbracket \rightarrow \llbracket \text{Integer} \rrbracket$$

$$\llbracket _ * _ \rrbracket(x, y) = \begin{cases} \perp, & x = \perp \text{ oder } y = \perp \\ xy, & \text{sonst} \end{cases}$$

Aufgabe 3

Als „Wahrheitstabelle“ ergibt sich

| x | y | L $\llbracket _ \text{or} _ \rrbracket(x, y)$ | A $\llbracket \text{not} \rrbracket(x)$ | B $\llbracket \text{not} \rrbracket(y)$ | C $\llbracket _ \text{and} _ \rrbracket(A, B)$ | R $\llbracket \text{not} \rrbracket(C)$ |
|--------------|--------------|--|--|--|---|--|
| \perp | \perp | \perp | \perp | \perp | \perp | \perp |
| <i>true</i> | \perp | <i>true</i> | <i>false</i> | \perp | <i>false</i> | <i>true</i> |
| <i>false</i> | \perp | \perp | <i>true</i> | \perp | \perp | \perp |
| \perp | <i>true</i> | <i>true</i> | \perp | <i>false</i> | <i>false</i> | <i>true</i> |
| <i>true</i> | <i>true</i> | <i>true</i> | <i>false</i> | <i>false</i> | <i>false</i> | <i>true</i> |
| <i>false</i> | <i>true</i> | <i>true</i> | <i>true</i> | <i>false</i> | <i>false</i> | <i>true</i> |
| \perp | <i>false</i> | \perp | \perp | <i>true</i> | \perp | \perp |
| <i>true</i> | <i>false</i> | <i>true</i> | <i>false</i> | <i>true</i> | <i>false</i> | <i>true</i> |
| <i>false</i> | <i>false</i> | <i>false</i> | <i>true</i> | <i>true</i> | <i>true</i> | <i>false</i> |

Durch Vergleich der mit L und R bezeichneten Spalten ergibt sich, dass für alle $x, y \in \llbracket \text{Boolean} \rrbracket$ gilt:

$$\llbracket _ \text{or} _ \rrbracket(x, y) = \llbracket \text{not} \rrbracket(\llbracket _ \text{and} _ \rrbracket(\llbracket \text{not} \rrbracket(x), \llbracket \text{not} \rrbracket(y)))$$