INF5150 Suggested solutions to exercises 2008-10-06

1. As usual we let $\mathcal{H}$ denote the set of all well-formed traces, and $\emptyset$ denote the empty set. \ is the symbol for set-minus, so $S_1 \setminus S_2$ denotes the set containing all elements that are in S_1 but not in S_2.

To compute the traces of Ex 1, Ex2 and Ex3 we need the definitions of seq, refuse, veto and assert:

Weak sequencing of trace sets:

$s1 \succsim s2$ denotes the set of all traces that may be constructed by selecting one trace t1 from s1 and one trace t2 from s2 and combining them in such a way that for each lifeline, the events from t1 comes before the events from t2.
Formally:

$$s1 \succsim s2 \stackrel{\text{def}}{=} \{h \in \mathcal{H} \mid \exists h1 \in s1, h2 \in s2 : \forall l \in \mathcal{L} : h{\restriction}l = h{\restriction}l \frown h{\restriction}2 \}$$

Note: if s1 or s2 is empty then $s1 \succsim s2$ is also empty

Weak sequencing of interaction obligations:

$$(p1,n1) \succsim (p2,n2) \stackrel{\text{def}}{=} (p1 \succsim p2 , (n1 \succsim p2) \cup (n1 \succsim n2) \cup (p1 \succsim n2))$$

seq:

$$[[d1 \text{ seq } d2]] \stackrel{\text{def}}{=} \{o1 \succsim o2 \mid o1 \in [[d1]] \wedge o2 \in [[d2]]\}$$

refuse:

$$[[\text{refuse } d]] \stackrel{\text{def}}{=} \{(\emptyset, p \cup n) \mid (p,n) \in [[d]]\}$$

veto:

$$[[\text{veto } d]] = \{(\{\langle\rangle\}, p \cup n) \mid (p \cup n) \in [[d]]\}$$

assert:

$$[[\text{assert } d]] \stackrel{\text{def}}{=} \{(p, n \cup (\mathcal{H} \setminus p)) \mid (p,n) \in [[d]]\}$$

Let $t1 = <!e,?e,!f,?f>$ $t2 = <!e,!f,?e,?f>$ $t3 = <!e,?e>$

$$[[Ex1]] = \{ (\{<!e,?e>\},\emptyset) \} \succsim \{ (\emptyset,\{<!f,?f>\}) \}$$
$$= \{(\{<!e,?e>\} \succsim \emptyset,\ (\emptyset \succsim \emptyset) \cup (\emptyset \succsim \{<!f,?f>\}) \cup (\{<!e,?e>\} \succsim \{<!f,?f>\}))\}$$
$$= \{ (\emptyset,\{t1,t2 \}) \}$$

$$[[Ex2]] = \{ (\{<!e,?e>\},\emptyset) \} \succsim \{ (\{\langle\rangle\},\{<!f,?f>\}) \}$$
$$= \{ (\{t3\},\{t1, t2\}) \}$$

$$[[Ex3]] = \{ (\{<!e,?e>\},\emptyset) \} \succsim \{ (\{<!f,?f>\}, \mathcal{H} \setminus \{<!f,?f>\}) \}$$
$= \{ (\{t1,t2\},n) \}$, where

$n = \{t \in \mathcal{H} \mid$ the first event on lifeline y is !e and the first event on lifeline x is ?e$\} \setminus$

{<!e,?e,!f,?f>,<!e,!f,?e,?f>}.
This means that the set *n* contains all traces where !e is the first event on lifeline y and ?e is the first event on lifeline x, except from the traces <!e,?e,!f,?f> and <!e,!f,?e,?f>.

To compute Ex4 we also need the definition of alt:

alt:

$[[d1 \text{ alt } d2]] \stackrel{def}{=} \{o1 \uplus o2 \mid o1 \in [[d1]] \wedge o2 \in [[d2]]\}$, where

$(p1,n1) \uplus (p2,n2) \stackrel{def}{=} (p1 \cup p2, n1 \cup n2)$

Let

t4 = <!a,?a,!b,?b>
t5 = <!a,!b,?a,?b>
t6 = <!c,?c >
t7 = <!a,?a,!b,?b,!e,?e>
t8 = <!a,!b,?a,?b,!e,?e>
t9 = <!a,?a,!b,?b,!e,?e,!f,?f>
t10 = <!a,!b,?a,?b,!e,?e,!f,?f >
t11 = <!a,?a,!b,?b,!e,!f,?e,?f>
t12 = <!a,!b,?a,?b,!e,!f,?e,?f >
t13 = <!c,?c,!e,?e>
t14 = <!c,?c,!e,?e,!f,?f>
t15 = <!c,?c,!e,!f,?e,?f>

$[[Ex4]] = \{(\{t4,t5\},\emptyset) \uplus (\emptyset,\{t6\})\}$
$\quad = \{(\{t4,t5\},\{t6\})\}$

$[[Ex5]] = [[Ex4 \text{ seq } Ex2]]$
$\quad = \{(\{t4,t5\} \succsim \{t3\}, (\{t6\} \succsim \{t3\}) \cup (\{t6\} \succsim \{t1,t2\}) \cup (\{t4,t5\} \succsim \{t1,t2\}))\}$
$\quad = \{(\{t7,t8\}, \{t9,t10,t11,t12,t13,t14,t15\})\}$

To compute Ex6 we need the definitions of xalt and constraints:

xalt:

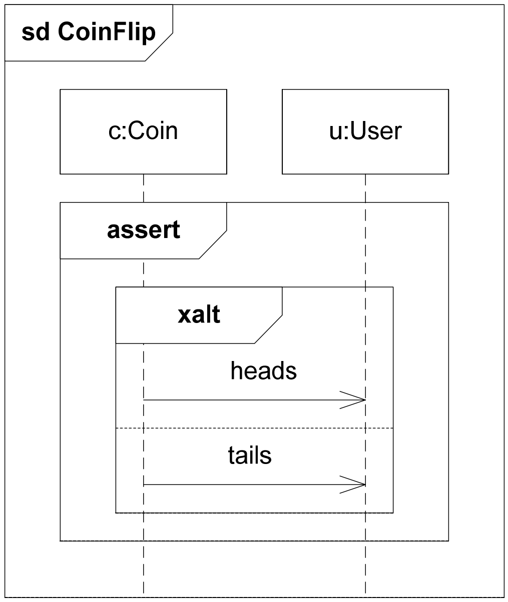
$[[d1 \text{ xalt } d2]] \stackrel{def}{=} [[d1]] \cup [[d2]]$

Constraints (i.e., guards):

$[[\text{ constr}(c) \text{ }]] \text{ def} = \{(\{\langle check(\sigma)\rangle \mid c(\sigma)\}, \{\langle check(\sigma)\rangle \mid \neg c(\sigma)\})\}$

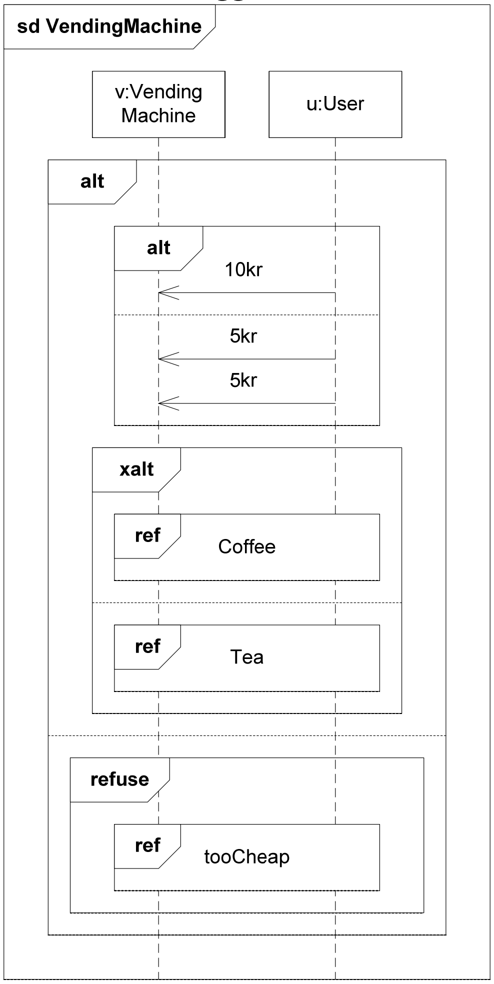
[[Ex6]]={({<chk(att=7),!a,?a>},{<chk(att≠7),!a,?a>})}∪
{({<chk(att≠7),!b,?b,!c,?c>},{<chk(att=7), !b,?b,!c,?c >})}
= { ({<chk(att=7),!a,?a>},{<chk(att≠7),!a,?a>}),
({<chk(att≠7),!b,?b,!c,?c>},{<chk(att=7), !b,?b,!c,?c >}) }

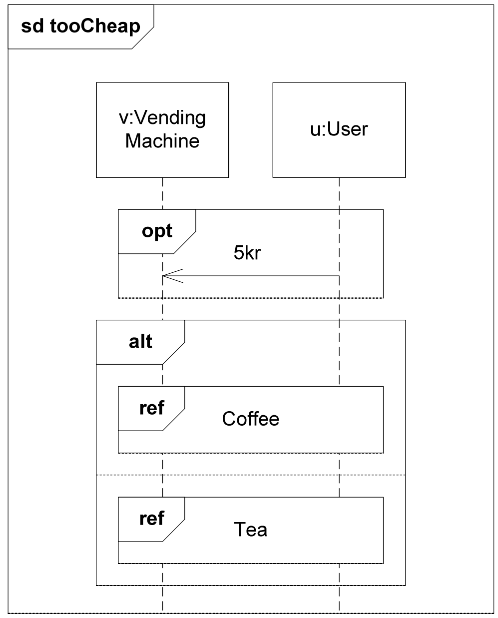
2.

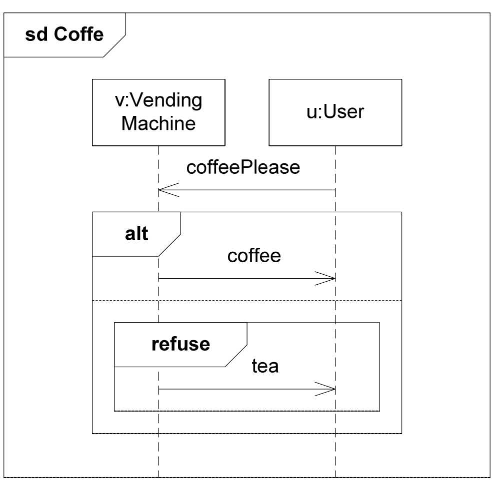


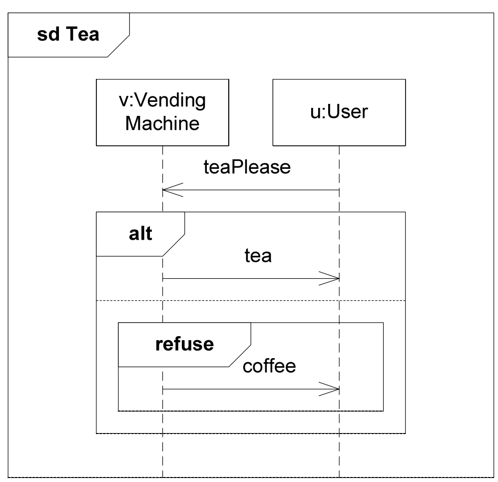
We choose to use limited refinement. This ensures that no new interaction obligations representing other outcomes are introduced.

3. Here is one suggestion:









Note that all traces from the “tooCheap” specification become negative in both interaction obligations in the VendingMachine specification.

We choose to use general refinement, in order to leave open the possibility that the final implementation also offers other drinks.

4. The mapping L is given by

L={p:PaymentHandler$\mapsto$v:VendingMachine, d:drinkPreparator$\mapsto$v:VendingMachine,

u:User$\mapsto$u:User }

Notice that we have broken a UML principle since it is not clear from the diagrams that d:drinkPreparator and p:PaymentHandler are parts of v:VendingMachine