Advanced Programming (I00032)
Testing simulation relation of iTask

Assignment 10

Preparation

In this exercise you implement the simulation relation $s \subseteq t$ that determines whether every event that makes sense in s also makes sense in t. Study the lecture slides. Take the following preparation steps:

- On Blackboard you find the complete operational semantics of iTasks that has been presented in the previous lecture (iTaskSemantics.zip).
- Create a new module that imports the semantic definition `iTask_semantics` and the `confSM` module of G∀st.
- Set the project to iTasks. In the project settings, add a path to `{Application}/iTasks-SDK/Examples/ESMVizTool`
  It contains the G∀st modules that compile with the latest iTask modules and generics.
- We advice you to use the `InputFun` option of `testConfSM` from module `confSM` to generate appropriate inputs for your task that acts as specification. For this to work, you need to alter line 473 of `confSM.icl` into:

```
InputFun f = {ts & inputs = f }
```

Assignment

With the above preparations, implement the simulation relation $s \subseteq t$ that has been presented in the lecture, using the `testConfSM` function from `confSM.icl`. The signature of your function should be:

```
task_conformance :: *World (Task' a) (Task' a) → *World | gEq {|*|} a & genShow {|*|} a
task_conformance world ta1 ta2
    = snd (testConfSM options
                    spec_task (ta1,initState)
                    IUT_task  (ta2,initState) (const (ta2,initState))
                    world
      )
where
    options = [InputFun (const proper_events)]
```

Deadline

The deadline for this exercise is December 9, 23:59h. Upload only the new module(s) that you have created for this assignment. We assume that you have not altered the files that you find in iTaskSemantics.zip.