Cyber-Physical Computation

Last assignment

José Proença and Renato Neves

First task (managing shared resources with Uppaal)

Consider a small private airfield used by 2 planes, which can be either flying, parked, landing, or taking off. The landing field is a shared resource by both planes. Consider the following requirements:

- only 1 plane can use the field at a time;
- a Controller component receives requests to *land* or to *take off*, and replies with a *wait* signal when the field is not available;
- each plane sends requests to the Controller to *land* or to *take off*, and sends notifications when the field becomes *free*;
- the Controller has 5 time units to notify a plane to wait;
- after 5 time units from requesting access to the field and with no wait signal, the planes take another 5 time units to reach the field.
- each plane takes non-deterministically between 1-3 time units to take off, and between 4-6 time units to land and park.
- after taking off and after parking the planes notify the Controller with a *gone* signal.
- if a plane is told to wait, we assume it will take between 5-7 time units to reach the field.

Suggest a model for the planes and the controller. List 4 to 8 desired properties that the model should satisfy. Verify the properties via Uppaal.
Extra points: Extend your model to handle n planes at once.

Second task (small essay)

Write a small essay detailing the differences between modelling and verification (as you saw in Uppaal) and programming. If possible complement your explanations with a concrete running example.

Third task (program semantics and monads)

Consider the following *probabilistic* while-language.

$$\texttt{Prog}(X) \ni \texttt{x} := \texttt{t} \mid \texttt{p} +_p \texttt{q} \mid \texttt{p} \,;\, \texttt{q} \mid \texttt{if}\ \texttt{b}\ \texttt{then}\ \texttt{p}\ \texttt{else}\ \texttt{q} \mid \texttt{while}\ \texttt{b}\ \texttt{do}\ \{\ \texttt{p}\ \}$$

Notice that it contains a new language construct: namely $\texttt{p} +_p \texttt{q}$ which runs $\texttt{p}$ with probability p and $\texttt{q}$ with probability $1 - p$. Develop a new semantics for the extended language and implement it in Haskell using the probability monad.

Here are some suggestions to help you get started: use the code developed in previous lectures and also the library with the probability monad (available on the website). Regarding the semantics, start with the following rule for sequential composition and then try to derive the others.

$$\frac{\langle \mathtt{p}, \sigma\rangle \Downarrow \sum_i^n p_i \cdot \sigma_i \qquad \forall i \leq n.\, \langle \mathtt{q}, \sigma_i\rangle \Downarrow \mu_i}{\langle \mathtt{p}\ ;\ \mathtt{q}, \sigma\rangle \Downarrow \sum_i^n p_i \cdot \mu_i}\ \text{(seq)}$$

Extra points: Extend the semantics to handle a selection of your favorite effects, for example delays, log messages, or exceptions.

What to submit: A single report in PDF for all tasks **and** all the relevant source files. Send by email (nevrenato@gmail.com) a unique zip file "`cpc2223-N1_N2.zip`", where `N1` and `N2` are your student numbers. The subject of the email should be "`cpc2223 N1 N2`".

Deadline: 26th June 2023 @ 23h59