

CS_411
CRITICAL SYSTEMS
Coursework 2 (5%)

Due: Friday, 22 March 2002, 12:00

Question 1

- (a) In Coursework 1, Question 1 you had to introduce some q and show in Agda

$q :: E$

Translate q and E into standard type theoretic (non-Agda-)syntax and derive using the rules introduced in the lecture the judgement $q : E$.

[8 marks]

- (b) In Coursework 1, Question 2 you had to introduce the type `MatMultType` of the matrix multiplication function. This was based on definitions of

$N :: \text{Set}$

and

```
Vec (A :: Set)
    (n :: N)
  :: Set
```

We can now introduce a new constant `MatMultType1` which doesn't make use of the definition of N and `Vec` and instead depends on parameters

$N :: \text{Set}$

and

$\text{Vec} :: \text{Set} \rightarrow N \rightarrow \text{Set}$,

ie. we have

```
MatMultType1 (N :: Set)
              (Vec :: Set -> N -> Set)
              :: Type
```

Define now in the standard type-theoretic (non-Agda-) syntax a constant `MatMultType1` and show using the rules introduced in the lecture

$$N : \text{Set}, \text{Vec} : \text{Set} \rightarrow N \rightarrow \text{Set} \Rightarrow \text{MatMultType1} : \text{Type}$$

(Note that N , and Vec are variables).

[12 marks]

Question 2

The goal of this exercise is to model a simple example of a railway interlocking system in Agda. Assume there is a railway line with one track only from A-Town to B-Town (you can use some local names for them if you want). In between the railway crosses a road, which is controlled by a barrier. There is one signal in A-Town, which controls the access to this railway line from A-Town to B-Town, and one in B-Town, which controls access to it from from B-Town.

We make several simplifying assumptions:

- We assume that there is a well-defined installation in A-Town and B-Town, which takes care of any trains entering these stations. So we don't have to control access to A-Town or B-Town.
- We don't assume that any train will move backwards.
- Further we ignore the time it takes for the barrier to open and close and the time it takes to get from A-town to B-town and back.

Therefore we need only to guarantee, that

- if the signal from A-Town to B-Town is green, the signal from B-Town to A-Town is red and the barrier is closed,
- if the signal from B-Town to A-Town is green, the signal from A-Town to B-Town is red and the barrier is closed

You should carry out the following steps:

- Define the set of states of a signal (red or green).
- Define the set of states of a barrier (up or down).
- Define the set of states of the whole system (which determines the states of the signals and the barrier).
- Define a predicate `correct`: depending on the signal states and the barrier it should be
 - the one element set, in case there is no collision
 - * This means that we have a proof of its correctness.
 - the empty set, in case there is a collision.
 - * This means that we cannot prove the correctness in this case.

- Define now a set of correct states of the system, which can be chosen by the operator:
 - Either no train is allowed on the line,
 - or there is one train allowed to move from A-Town to B-Town,
 - or there is one train allowed to move from B-Town to A-Town,
- Determine for each of these correct states the state of the signals and the barrier. (For instance if there is no train on the line, all signals are red and the barrier is up).
- Now show that for every correct state, we have that the system is safe, ie. that the correctness predicate holds for the states of the signals and of the barrier you have chosen.

If you want you can, once you have finished the above, model a more advanced situation. You will get extra marks for that. [30 marks]