

## Homework 4 of CS520 Theory of Programming Languages

### Deadline: 6:00pm on 4 June (Friday)

Submit your solutions in KLMS. (Reminder: We adopt a very strict policy for handling dishonest behaviours. If a student is found to copy answers from fellow students or other sources in his or her homework submission, she or he will get F.)

The numbers in the questions refer to exercise questions in the textbook of the course, i.e. “Theories of Programming Languages” by John C. Reynolds.

#### Question 1

Solve 11.4.

(40 marks)

#### Question 2

Solve 11.11. While solving this question, you might find it useful to use lists. Lists are explained in Section 11.4 of the textbook. If you do not want to read the textbook but still are interested in using lists, a quick solution is to view constructs for lists as syntactic sugars, as shown below:

$$\begin{aligned}\mathbf{nil} &\stackrel{\text{def}}{=} @\ 0\ 0 \\ e_0 :: e_1 &\stackrel{\text{def}}{=} @\ 1\ \langle e_0, e_1 \rangle \\ \mathbf{listcase}\ e\ \mathbf{of}\ (e_0, e_1) &\stackrel{\text{def}}{=} \mathbf{sumcase}\ e\ \mathbf{of}\ \left( (\lambda v. e_0), (\lambda v. (e_1\ v.0)\ v.1) \right) \quad \text{for a fresh variable } v\end{aligned}$$

Intuitively, **nil** denotes the empty list, and  $e_0 :: e_1$  denotes the list with  $e_0$  as its head and  $e_1$  as its tail. The last performs the pattern match. If  $e$  is the empty list, we evaluate  $e_0$ . Otherwise, we apply  $e_1$  to two parameters, one for the head of  $e$  and the other for the tail of  $e$ .

(20 marks)

#### Question 3

Solve 11.12.

(40 marks)