

BU CAS CS 320 (FALL SEMESTER, 2006)
CONCEPTS OF PROGRAMMING LANGUAGES

Assignment 8

Out: Tuesday, 28 November 2006
Due: Thursday, 07 December 2006

Total: 140 points

Exercise 1 (30 points) The signature RATIONAL is given in Assignment 3. Please implement a functor that takes a structure of signature INTEGER to generate a structure of signature RATIONAL.

```
functor RationalFun (I: INTEGER) :> RATIONAL = struct
  (* your implementation *)
end
```

Exercise 2 (30 points) Please construct the typing derivations for the following terms:

- (10 pts) $\emptyset \vdash \mathbf{lam} x.\langle x + 1, x > 1 \rangle : \mathbf{int} \rightarrow (\mathbf{int} * \mathbf{bool})$, where $+$ and $>$ are assumed to have the constant types $(\mathbf{int}, \mathbf{int}) \Rightarrow \mathbf{int}$ and $(\mathbf{int}, \mathbf{int}) \Rightarrow \mathbf{bool}$, respectively.
- (20 pts) $\emptyset \vdash \mathbf{lam} f.\mathbf{lam} x.\mathbf{app}(\mathbf{app}(f, x), x) : (T_1 \rightarrow (T_1 \rightarrow T_2)) \rightarrow (T_1 \rightarrow T_2)$.

Exercise 3 (10 pts) Can the term $\mathbf{lam} x.\mathbf{app}(x, x)$ be assigned a simple type? If yes, please construct a typing derivation to justify your answer. If no, please briefly state your reasoning.

Exercise 4 (20 pts) Let $K = \mathbf{lam} x.\mathbf{lam} y.x$ and $S = \mathbf{lam} x.\mathbf{lam} y.\mathbf{lam} z.\mathbf{app}(\mathbf{app}(x, z), \mathbf{app}(y, z))$. Given a value v , please find another value v' so that you can construct a derivation for:

$$\mathbf{app}(\mathbf{app}(\mathbf{app}(S, K), K), v) \rightarrow^* v'$$

Exercise 5 (20 pts) Please construct a term t such that $\emptyset \vdash t : (T_1 \rightarrow T_2) \rightarrow ((T_2 \rightarrow T_3) \rightarrow (T_1 \rightarrow T_3))$ is derivable for any given types T_1, T_2, T_3 .

Exercise 6 (30 pts) Given the following datatypes for representing expressions and values, please implement a function *eval* that evaluates an expression to a value.

```
datatype exp =
  EXPbool of bool
| EXPint of int
| EXPprimfun of string * exp list
| EXPif of exp * exp * exp
| EXPtup of exp * exp
| EXPfst of exp
| EXPsnd of exp
```

```
datatype value =  
  VALbool of bool  
  | VALint of int  
  | VALtup of value * value
```

In particular, *eval* should be given the type $exp \rightarrow value$.