Homework 5: Type Checker
COSE212, Fall 2015

Hakjoo Oh

Due: 12/5, 24:00

Problem 1 In this problem, your goal is to implement a type checker for the
PROC language:

```
type exp =
  | CONST of int
  | VAR of var
  | ADD of exp * exp
  | SUB of exp * exp
  | ISZERO of exp
  | IF of exp * exp * exp
  | LET of var * exp * exp
  | PROC of var * exp
  | CALL of exp * exp
and var = string
```

Types for the language are defined as follows:

```
type typ = TyInt | TyBool | TyFun of typ * typ | TyVar of tyvar
and tyvar = string
```

The type checker is implemented by the function:

$$\text{typeof : exp -> typ}$$

It takes a program and returns its type if the program is well-typed. When the
program is ill-typed, `typeof` should raise an exception `TypeError`.

Examples:

- The program

    ```
    PROC ("f",
     PROC ("x", SUB (CALL (VAR "f", CONST 3),
                     CALL (VAR "f", VAR "x"))))
    ```

    has type `TyFun (TyFun (TyInt, TyInt), TyFun (TyInt, TyInt))`.

- The program

  ```
  PROC ("f", CALL (VAR "f", CONST 11))
  ```

  has type `TyFun (TyFun (TyInt, TyVar "t"), TyVar "t")`, where `t` can be any type variable.

- The program

  ```
  LET ("x", CONST 1,
    IF (VAR "x", SUB (VAR "x", CONST 1), CONST 0))
  ```

  is ill-typed, so `typeof` should raise an exception `TypeError`.

As discussed in class, the function `typeof` is defined with two functions: one for generating type equations and the other for solving the equations. Your job is to complete the implementation of these two functions:

$$\text{gen\_equations} \ : \ \text{TEnv.t -> exp -> typ -> typ\_eqn}$$
$$\text{solve} \ : \ \text{typ\_eqn -> Subst.t}$$

Download `hw5.ml` from the course web-page, implement `gen_equations` and `solve`, and submit the file via Blackboard. In `hw5.ml`, modules for type environments (`TEnv`) and substitutions (`Subst`), as well as the operations of applying substitutions to types (`Subst.apply`) and extending substitutions (`Subst.extend`), are provided. Use these modules when implementing the type checker.