Homework 2

CSE 4102 Project 2, Spring 2016

William Dickson

02/07/2016

Section: 001

Instructor: Jeffrey A. Meunier

## Introduction:

The intent of this assignment is the beginning developments of an evaluator for a functional programming language.

In order to be able to write interesting programs using this language, we need the ability to define names, or said another way, to bind values to names. In the evaluator for the language, these names will be managed in a data structure called an environment. The

environment is just a list of bindings, where a binding is a mapping of name to value.

In this assignment I implement the environment using two different methods.

## Output:

liam@liam-VirtualBox:[~/Documents/CSE4102/hw2]$ sml env1.sml

Standard ML of New Jersey v110.76 [built: Tue Oct 22 14:04:11 2013]

[opening env1.sml]

exception NameNotBound of string

type Env = (string \* int) list

val env\_new = fn : unit -> Env

val env\_bind = fn : Env -> string -> int -> Env

val env\_lookup = fn : Env -> string -> int

val e1 = [] : Env

val e2 = [("x",100)] : Env

val e3 = [("y",200),("x",100)] : Env

val it = 100 : int

val it = 200 : int

uncaught exception NameNotBound

raised at: env1.sml:16.48-16.65

/usr/lib/smlnj/bin/sml: Fatal error -- Uncaught exception NameNotBound with "z" raised at env1.sml:16.48-16.65

liam@liam-VirtualBox:[~/Documents/CSE4102/hw2]$ sml env2.sml

Standard ML of New Jersey v110.76 [built: Tue Oct 22 14:04:11 2013]

[opening env2.sml]

exception NameNotBound of string

type Env = string -> int

val env\_new = fn : unit -> Env

val env\_bind = fn : Env -> string -> int -> Env

val env\_lookup = fn : Env -> string -> int

val e1 = fn : Env

val e2 = fn : Env

val e3 = fn : Env

val it = 100 : int

val it = 200 : int

uncaught exception NameNotBound

raised at: env2.sml:12.50-12.67

/usr/lib/smlnj/bin/sml: Fatal error -- Uncaught exception NameNotBound with "z" raised at env2.sml:12.50-12.67

## Source Code:

**env1.sml**

(\*

\* Name: William Dickson

\* Class: CSE 4102

\* Instructor: Jeff Meunier

\* Due Date: Feb 9, 2016

\*)

exception NameNotBound of string;

type Env = (string \* int) list;

fun env\_new() : Env = nil;

fun env\_bind (env:Env) (name:string) (value:int) : Env = ((name, value)::env);

fun env\_lookup (nil:Env) (name:string) = raise NameNotBound name

| env\_lookup (env:Env) (name:string) =

let

val (nm:string, vl:int) = hd(env);

val envs:Env = tl(env);

in

if nm = name then

vl

else

env\_lookup envs name

end

;

(\* Test Cases \*)

val e1 = env\_new ();

val e2 = env\_bind e1 "x" 100;

val e3 = env\_bind e2 "y" 200;

env\_lookup e3 "x";

env\_lookup e3 "y";

env\_lookup e3 "z";

**env2.sml**

(\*

\* Name: William Dickson

\* Class: CSE 4102

\* Instructor: Jeff Meunier

\* Due Date: Feb 9, 2016

\*)

exception NameNotBound of string;

type Env = string -> int;

fun env\_new () : Env = fn (name:string) => raise NameNotBound name;

fun env\_bind (env:Env) (name:string) (value:int): Env =

fn (input:string) =>

if input = name then value

else env input

fun env\_lookup (env:Env) (name:string) = env name

(\* Test Cases \*)

val e1 = env\_new ();

val e2 = env\_bind e1 "x" 100;

val e3 = env\_bind e2 "y" 200;

env\_lookup e3 "x";

env\_lookup e3 "y";

env\_lookup e3 "z";