Programming Languages Hw4

Arya Banaeizadeh 9431029

John Mitchell, Concepts in Programming Languages

5.1 Algol 60 Procedure Types

```
Procedure Q(t);

Integer t;

Begin

        Q := 1 + t;

End;
```

In algol 60 the type of procedure when passing to another procedure is not checked and therefore the code won't raise a static error. However, in runtime a type error occurs.

5.2 Algol 60 Pass-By-Name

P(A[1]) => i = A[1] => i = 2 => A[2] = 2 => i , A[1] , A[2] = 2

5.3 Nonlinear Pattern Matching

a)

```
fun f(p) =

        if #2(p) = 0 then #1(p)

        else if #1(p) = 0 then #2(p)

        else #1(p) + #2(p)
```

b)

we can't use eq(x,x) because of duplicate name variables.

c)

we are able to refer to the same variable in a function although we are not allowed to use duplicate variables when declaring function parameters:

```
fun eq(x,y)=if x=y then true else false
```

d)

in ML equality check on function types is not defined and therefore duplicate variable usage in function parameters raises an error

5.4 ML Map for Trees

a)

Fun maptree(f , leaf(y)) = leaf(f(y)) |

maptree (f, node(z,t)) = node(maptree(f,z),maptree(f,t))

b)

('a→'b)→'a tree→'b tree

any type can be given to f.

5.6 Currying

a)

Fun F1(a,b)=c , Fun F2(a)=F3 , Fun F3(b)=c

Fun Curry(F1)=F2

Fun G1(a,b)=z , Fun G2(a)=U3 , Fun G3(b)=c

Fun UnCurry(G2)=G1

b)

1. UnCurry input type and Curry output are equal
2. UnCurry output and Curry input are equal
3. Curry(UnCurry(f)) = (f -> g) -> g -> f = f
4. UnCurry(Curry(f)) = (g -> f) -> f -> g = g

5.8 Lazy Evaluation and Functions

a)

Fun merge (a,nil) = a |

Merge (nil,b) = b |

Merge (a,b) = cons Merge(a::head(b), tail(b))

b)

fun compose(f, g) = h x => f(g(x));

c)

f output should not be in the domain of g and g must be infinite