In the name of God

PL homework #5

PanteA Habibi - 9131010

chapter 5 - Concepts in Programming Languages - J.C.Mitchell

۱)

procedure Q(t); integer x; begin Q=x+t; end;

در حالت static درست است زیرا تایپ procedure در algol، proc است و مستقل از پارامترهاست.

جمع بولین و عدد صحیح خطای زمان اجرا تولید می کند پس این تابع نیز در زمان اجرا خطا دارد.

۳)

a-

fun f(x,y)= if(y=0) then x else if(x=0) then y else x+y

b-

در ML، eq(x,x) غلط است یعنی در واقع دو آرگومان همنام نمیتوان داشت و error برمیگرداند.

c-

ممکن است به هر دو پارامتر تابع یک موجودیت اختصاص یابد اما در خود تابع باید جدا در نظر گرفته شود. با if، else و else if حالت مورد نظر را چک میکنیم و نتیجه ی دلخواه را برمیگردانیم. مثلا در تابع eq اگر به صورت زیر عمل کنیم درست و بدون خطا خواهد بود.

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

d-

همانطور که در سوال گفته شده type توابع به درستی ارزیابی نمیشودکه آیا equal هستند یا نه. از همین رو چون توانایی این چک کردن را ندارد پس نمیتوان در pattern ها، variableهای یکسان داشت. در واقع بعضی اوقات اتفاق می افتد که نمیتواند type ای را پیدا کند.

۴)

a-

اگر ورودی برگ باشد ، f آن محاسبه می شود و اگر نود میانی باشد ، تابع روی دو فرزندش پیاده می شود.

fun maptree (f, leaf(x))=leaf(f(x)) | maptree(f,node(x,y))=node(maplist(f x),maplist(f y)) =

maptree(f,f(x),f(y))

b-

چون لزوما درخت از نوع `a نخواهد بود مثلا اگر به هر برگ 0.5 اضافه شود از int به real تبدیل می شود و چون از ورودی و خروجی مطلع نیست میتواند هر چیزی باشد.

('a→'b) →'a tree→'b tree

۶)

a-

fun f1(a,b)=c; fun f2(a)=b; fun f3(b)=c

fun curry(f1) = f3(f2(a))

fun uncurry(f3(f2(a))) = f1(a,b)

b-

uncurry(curry(f1)) = uncurry(f3(f2(a))) = f1(a,b)

۷)

a-

اگر x رشته باشد ، x.i در زمان محاسبه ی y مقدار معلومی ندارد و جواب خروجی مقداری غیر معلوم خواهد بود و در عین حال بدون اعلام هیچ خطایی انجام گرفته است . مقدار آن هم بر اساس آن عددی که از قبل در خانه ای که x.i به آن اشاره می کند وجود داشته است جایگزین میشود.

b-

ML constructors can be declared so that they must be applied to arguments when constructing elements of the data type. Constructors do not actually do anything to their arguments, other than to "tag" their arguments so that values constructed in different ways can be distinguished by pattern matching.

In this data-type declaration, tag_int, and tag_str are each constructors. each IntString constructor must be applied to arguments to construct a value of type IntString. We must apply tag_int to a int, tag_str to a string in order to produce a value of type IntString.

Except that in ML "unions" (which are defined by datatype), each value of the union is tagged by a constructor that tells which of the constituent types the value comes from.

So when m is a string, tag_int can not be applied to m to produce type of IntString based on pattern matching. So ML determines the bug and shows the error.