

**Rational2**: reduce only when toString is called

structure **Rational2** :> RATIONAL\_A =

struct

datatype rational = Whole of int | Frac of int\*int

exception BadFrac

fun make\_frac (x,y) =

if y=0

then raise BadFrac

else if y<0

then Frac(~x,~y)

else Frac(x,y)

fun add (r1, r2) =

case (r1,r2) of

(Whole(i), Whole(j)) => Whole(i+j)

| (Whole(i), Frac(j,k)) => Frac(j+k\*i,k)

| (Frac(j,k), Whole(i)) => Frac(j+k\*i,k)

| (Frac(a,b), Frac(c,d)) => Frac(a\*d + b\*c, b\*d)

fun toString r =

let fun gcd (x,y) =

if x=y

then x

else if x<y

then gcd(x,y-x)

else gcd(y,x)

fun reduce r =

case r of

Whole \_ => r

| Frac(x,y) =>

if x=0

then Whole 0

else let val d = gcd(abs x,y) in

if d=y

then Whole(x div d)

else Frac(x div d, y div d)

end

in

case reduce r of

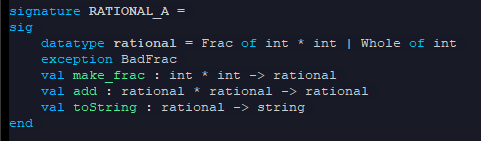
Whole i => Int.toString i

| Frac(a,b) => (Int.toString a) ^ "/" ^ (Int.toString b)

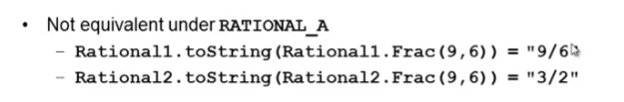
end

end

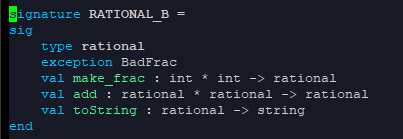
SIGNATURE A



* Type not abstract

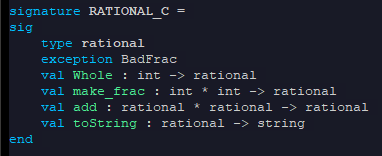


SIGNATURE B

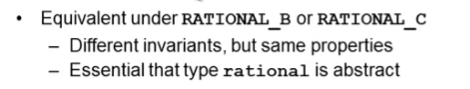


* Type now abstract

SIGNATURE C



* Type abstract
* Add a Whole constructor



* Because it exposed less
* make\_frac is the only option to create fractions. Frac cannot be used

