Recall $\lambda Th(\text{CommRing})$ is a typed $\lambda$-calculus — a programming language with a datatype "R" — a "commutative ring class.

This typed $\lambda$-calculus generates a CCC $C_{\lambda Th(\text{CommRing})}$.

Our puzzle last time was: what is a cartesian closed functor

$$F: C_{\lambda Th(\text{CommRing})} \to \text{Set}$$

There are tricky aspects to this puzzle... but let's start by reviewing what $\lambda Th(\text{CommRing})$ & $C_{\lambda Th(\text{CommRing})}$ are like:

<table>
<thead>
<tr>
<th>$\lambda Th(\text{CommRing})$</th>
<th>$C_{\lambda Th(\text{CommRing})}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>types:</strong></td>
<td><strong>objects:</strong></td>
</tr>
<tr>
<td>1</td>
<td>1 the terminal object</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>$R \times R$</td>
<td>$R \times R$</td>
</tr>
<tr>
<td>$R \times \text{hom}(R \times R, \text{hom}(R, R))$</td>
<td>$R \times \text{hom}(R \times R, \text{hom}(R, R))$</td>
</tr>
<tr>
<td>etc.</td>
<td>etc.</td>
</tr>
</tbody>
</table>

| **terms:**                   | **morphisms:**                    |
| variables of any type:       | A morphism $f: A \to B$ is an (equivalence class of) expressions $(x \in A, \phi(x))$ |
| $x, y, z, \ldots \in A$      | where $\phi(x) \in B$ is a term whose only free variable is $x$. (For example, we get $\Delta: A \to A \times A$ from $(x \in A, (x, x))$. |

Given terms $a \in A, b \in B$, we get $(a, b) \in A \times B$. 
Given a term $a \in A \times B$
we have terms
$\pi_1(a) \in A$, $\pi_2(a) \in B$.

Given terms $f \in \text{hom}(A,B)$
and $a \in A$, we have a
term $f(a) \in B$.

We also have lots of
equations between terms,
e.g.
$\forall x \exists y \langle \pi_1(c), \pi_2(c) \rangle$
where $c \in A \times B$ contains
free variables in $X$.

We also have

$0 \in R$

$\in \text{hom}(R,R)$

("negation")

(We left out negation before,
but we need it since we can't
say "\forall x \in R \exists y \in R \, x + y = 0".
in the typed $\lambda$-calculus — we
don't have "\exists".)

$\cdot \in \text{hom}(R \times R, R)$

$1 \in R$

We get the "addition" morphism
$+: R \times R \to R$

from
$\langle x \in R \times R, +(\pi_1(x), \pi_2(x)) \rangle$
or equivalently
$\langle x \in R \times R, +(x) \rangle$

How do we talk about $0$?
It should be some morphism
$0 : 1 \to R$

How do we get this?
$\langle x \in 1, 0 \rangle$

How about
$\cdot : R \times R \to R$?

This comes from
$\langle x \in R \times R, \cdot (x) \rangle$

Similarly
$- : R \to R$

comes from
$\langle x \in R, -(x) \rangle$.

We have lots more, such as
$\langle x \in R \times R, +(\cdot (\pi_1(x), \pi_2(x)), \pi_2(x)) \rangle$

— heuristically "$(y,z) \mapsto y^2 + z$."
We also have terms like
\[ x \in \text{hom}(R, R) \mapsto x(x(y)) \]
which is of type \( \text{hom}(\text{hom}(R, R), R) \),
but \( \lambda \text{th}(\text{comm\,Ring}) \) doesn't use
this "\( \lambda \)-abstraction" for its
definition.

Finally, we have equations,
including the commutative ring
axioms:
\[ x + y = y + x \]
\[ xy = yx \]
etc.

We get an equation between two
morphisms \( (x \in A, \varphi(x)) \) &
\( (y \in A, \psi(y)) \) when we have
\[ \varphi(x) = \psi(y) \]
\[ \forall x \in A \]

So \( \lambda \text{th}(\text{comm\,Ring}) \) is the
CCC whose objects are
generated by \( R \) and whose
morphisms are generated by
\[ +, 0, -, 1, \] with relations
given by comm. ring axioms.
In short \( \lambda \text{th}(\text{comm\,Ring}) \) is
the free CCC on a comm.
ring object.
Back to our puzzle:

What's a cartesian closed functor

\[ F : C_{\text{Comm Ring}} \to \text{Set} \]

Guess: it's just a commutative ring! (Namely \( F(R) \).)