Dear Reviewers,

Thank you for your insightful reviews. We have tried to incorporate your suggestions. In particular, we have:

- Added about 2 pages of background on monads and effect systems.
- Removed claims about things we hope to achieve in the future (except in the Future Work section).
- Slightly expanded the explanation of the nanopass architecture in the introduction.
- Greatly expanded the relation to Eelco's work with appropriate citations in the Related Work section.
- The paper by Bach Poulsen and Van der Rest on higher order effects has been published, so we have now added an accessible citation of their work.

Some responses to particular points from the reviews:

> it would have been great to see the stack allocation pass

Unfortunately, it seems that our first submission of this paper erroneously stated we did not do stack allocation, but in fact we do. It is a bit obscured due to the fact that our language has no function calls, so there is no need for pushing and popping things from the stack, but we do allocate each variable in memory on the stack in the x86var handler.

> Why the use of higher-order effects? Why not: `[[let(e,f)]] = [[e]] >>= \x. [[f x]]`?

We explain below the implementation of the let handler that adding this extra step gives us the possibility of choosing a different evaluation order. However, we agree this does feel disappointing because we do not actually give such a different implementation.

> Is the use of PHOAS purely presentational or essential?

It is not essential, however most effect systems include a form of variable binding, which in our case conveniently works together with PHOAS. So changing to another binding mechanism would require changes in the effect system itself too. But we think that would not change the essence of our approach.

We have added this explanation in the revised paper.

> Have you run the generated machine code? If you did you should mention that!

Unfortunately not yet.