

Full Title*

Subtitle†

ANONYMOUS AUTHOR(S)

Write this last. State the problem. Say why it's an interesting problem. Say what your solution achieves. Say what follows from your solution.

Additional Key Words and Phrases: keyword1, keyword2, keyword3

1 INTRODUCTION

One page¹. This is a template for your (mine?, first?) research paper. The benefits of this template are:

- It has a bulleted list of contributions. This is the list you have to rewrite first. It will drive the entire paper.
- The repository, this template is in, has a Makefile. Using make will save you from setting up the build system for your paper. So you can start writing immediately. You'll need to install lhs2tex² though.³
- We present few examples of lhs2tex, a tool to make your (not only Haskell) code pretty type set. It can also be (ab)used for typing more than code blocks (Section 3).
- The template is prefilled with *Seven simple, actionable suggestions* by Jones [2015], that will make your papers better (Appendix A).
- We also mentions other seven actionable principles by Dreyer [2016]. Again, to make your papers better (Appendix B).

2 MAIN IDEAS

2–3 pages.

Figure 1 looks impressive...but sends readers to sleep and/or makes them feel stupid. Explain it as if you were speaking to someone using a whiteboard. Conveying the intuition is primary, not secondary. Once your readers have the intuition, they can follow the details (but not vice versa).

3 THE DETAILS

$$\begin{array}{c}
 \frac{}{x : A \vdash x : A} \text{VAR} \\
 \\
 \frac{\Gamma \vdash e : C}{\Gamma, x : A \vdash e : C} \text{WEAKEN} \quad \frac{\Gamma, x_1 : A, x_2 : A \vdash e : C}{\Gamma, x : A \vdash e : C} \text{CONTRACT} \quad \frac{\Gamma, \Delta \vdash e : C}{\Delta, \Gamma \vdash e : C} \text{EXCHANGE} \\
 \\
 \frac{\Gamma \vdash f : A \rightarrow B \quad \Delta \vdash x : A}{\Gamma, \Delta \vdash f x : B} \text{APP} \quad \frac{\Gamma, x : A \vdash e : B}{\Gamma \vdash \lambda x \rightarrow e : A \rightarrow B} \text{ABS}
 \end{array}$$

Fig. 1. Simply typed lambda calculus with explicit structural rules

*Title note

†Subtitle note

¹The page counts are for a denser two-column format. Scale appropriately

²<http://hackage.haskell.org/package/lhs2tex>

³The irony is that I setup this template, so I can avoid writing

More sections. 5 pages. This is where things like Figure 1 belong.

We use this section to demonstrate lhs2tex. You could use examples to introduce the problem, they are easy to write using lhs2tex.

```
zipWith :: (a → b → c) → [a] → [b] → [c]
zipWith k [] [] = []
zipWith k (x : xs) (y : ys) = k x y : zipWith k xs ys
zipWith k _ _ = error "lengths don't match"
```

But even if isn't related, you can (ab)use lhs2tex, to write maths like it was Haskell. For example we can define a category to consist of following data [Awodey 2010] (your paper doesn't need to be about category theory):

- Objects: A, B, C
- Arrows: f, g, h
- For each arrow f there are given objects: $\text{dom } f$ and $\text{cod } f$ called the *domain* and *codomain* of f .
- ...

These data are required to satisfy following laws

- Associativity:

$$f \circ (g \circ h) = (f \circ g) \circ h \quad (1)$$

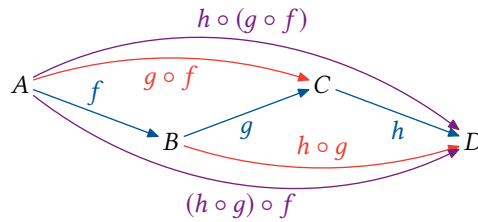
for all $f : A \rightarrow B, g : B \rightarrow C, h : C \rightarrow D$.

- Unit:

$$f \circ 1_A = f = 1_B \circ f \quad (2)$$

for all $f : A \rightarrow B$.

For further example, we can present associativity law as a diagram. The diagram below is made with a *MetaPost*⁴. You might consider using *diagrams* package⁵. Whatever tool you decide to use, reserve proper time to make your diagrams. "A picture is worth a thousand words" holds for information density for production time.



Next we'll test that *cleveref* works: Lemma 3.1 and Example 3.2.

LEMMA 3.1 (YONEDA). Let \mathcal{C} be locally small. For any object $C \in \mathcal{C}$ and functor $F \in \mathbf{Sets}^{\mathcal{C}^{\text{op}}}$ there is an isomorphism

$$\text{Hom}(y_C, F) \cong FC$$

which, moreover, is natural in both F and C .

PROOF. Omitted. □

Example 3.2. We can use Lemma 3.1 to derive Profunctor Optics [Boisseau and Gibbons 2018].

⁴<https://tug.org/metapost.html>

⁵<https://archives.haskell.org/projects.haskell.org/diagrams/>

4 RELATED WORK

1–2 pages. There are various resources.

- ACM SIGPLAN Author Information <http://www.sigplan.org/Resources/Author/> has a short *Writing* section.
- Simon Peyton Jones has a longer list on <https://www.microsoft.com/en-us/research/academic-program/write-great-research-paper/#!other-resources>.

5 CONCLUSIONS AND FURTHER WORK

Half a page.

REFERENCES

- Steve Awodey. 2010. *Category Theory* (2nd ed.). Oxford University Press, Inc., New York, NY, USA.
- Guillaume Boisseau and Jeremy Gibbons. 2018. What You Needa Know about Yoneda: Profunctor Optics and the Yoneda Lemma (Functional Pearl). *Proc. ACM Program. Lang.* 1, ICFP, Article 84 (Aug. 2018), 27 pages.
- Derek Dreyer. 2016. How to write papers so that people can read them. <https://www.mpi-sws.org/~dreyer/talks/talk-plmw16.pdf>. [Online; accessed 20-July-2018].
- Simon Peyton Jones. 2015. How to write a great research paper. <https://www.microsoft.com/en-us/research/academic-program/write-great-research-paper/>, <https://www.cis.upenn.edu/~sweirich/icfp-plmw15/slides/peyton-jones.pdf>. [Online; accessed 20-July-2018].

A SIMON’S SUGGESTIONS

We used some of *Seven simple, actionable suggestions* by Jones [2015], in this template. The all suggestions are:

- (1) **Don’t wait: write.** Writing papers is a primary mechanism for doing research (not just for reporting it)
- (2) **Identify your key idea.** Your goal s to convey a useful, re-usable, clear and sharp idea.
- (3) **Tell a story.** Imagine you are explaining at a whiteboard.
- (4) **Nail your contributions.** Do not leave the reader to guess what your contributions are!
- (5) **Related work: later.** Problems with too early related work: The reader knows nothing about the problem yet; so your description of various technical tradeoffs is absolutely incomprehensible. Describing alternative approaches gets between the reader and your idea
- (6) **Put your readers first (examples).** Introduce the problem, and your idea, using *examples* and only then present the general case.
- (7) **Listen to your readers.** Get your paper read by as many friendly guinea pigs as possible

There are various recordings of the presentation on YouTube, for example <https://www.youtube.com/watch?v=WP-FkUaOcOM>. I recommend watching the video, the presentation, questions and comments are all very insightful.

B DEREK’S PRINCIPLES

Dreyer [2016] gives seven concrete suggestions, which are different from Simon’s.

- (1) **Old to new.** Begin sentences with old info. End sentences with new info.
- (2) **One paragraph, one point.** A paragraph should have one main point, expressed in a single *point sentence*.
- (3) **Name your baby.** Give unique names to things and use them consistently.
- (4) **Just in time.** Give information precisely when it is needed, not before
- (5) **CGI model for abstract/intro.** *Context:* Set the stage, motivate the general topic. *Gap:* Explain your specific problem and why existing work does not adequately solve it. *Innovation:* State what you’ve done that is new, and explain how it helps fill the gap.

(6) **Have a “main ideas” section.** Use *concrete illustrative examples* and high-level intuition. Do *not* have to show the general solution.

(7) **Compare with related work at the end.** It goes at the end of the paper. Give real comparisons, not a “laundry list”!

There is a recording of the talk on YouTube: <https://www.youtube.com/watch?v=PM1Atui30qU>.