Type Independent Enumeration Based on Lambda Calculus

Rangeet Pan[1]

Iowa State University, Ames, IA, 50011, USA,
rangeet@iastate.edu

Abstract. **Keywords:** First class function, lambda calculus, type independent, enumerator, packing, unpacking, zip, unified typing.

1 Introduction

The abstract was a verbatim excerpt from Phil Koopman's advice on writing abstract [1]. The copyright for the text is held by Philip Koopman. More information about it is available in the reference. Below we will discuss Dr. Stirewalts's 5-paragraph rule for writing introductions.

Dr. Stirewalt's 5-paragraph rule for writing Introductions: Of the many tasks involved in writing a good conference paper, I find writing the introduction section to be the most difficult. This is unfortunate, as a poorly structured argument sets the wrong tone for what might otherwise be really good research. To help manage this painful process, I have developed a heuristic, called the *five-paragraph rule*, that is useful for organizing introductions. The heuristic prescribes that good introductions should contain a sequence of five major pieces, each of which should fit into a single paragraph in order to force the writer to communicate at the appropriate level of abstraction. The heuristic borrows ideas from *persuasive argument* and *structured analysis/structured design* (ala DeMarco/Yourdon), and it is reminscent of a similar structuring mechanism from freshman level courses in English composition. My success in publishing papers increased dramatically once I began to use this heuristic to structure my introductions.

The heuristic is: Design your introductions to comprise five paragraphs whose purpose and contents are as follows:

1. **Introductory paragraph**: Very briefly: What is the problem and why is it relevant to the audience attending *THIS CONFERENCE*? Moreover, why is the problem hard, and what is your solution? You must be brief here. This forces you to boil down your contribution to its bare essence and communicate it directly.
2. **Background paragraph**: Elaborate on why the problem is hard, critically examining prior work, trying to tease out one or two central shortcomings that your solution overcomes.
3. **Transition paragraph**: What keen insight did you apply to overcome the shortcomings of other approaches? Structure this paragraph like a syllogism: Whereas P and P => Q, infer Q.

4. **Details paragraph**: What technical challenges did you have to overcome and what kinds of validation did you perform?
5. **Assessment paragraph**: Assess your results and briefly state the broadly interesting conclusions that these results support. This may only take a couple of sentences. I usually then follow these sentences by an optional overview of the structure of the paper with interleaved section callouts.

2 Background

In this section, I will present necessary background. The objective of presenting the background here is to make the paper self-contained. After reading this section, even those readers who may not be initially familar with the underlying ideas behind my technique will be able to understand the rest of the paper.

3 Related Work

In this section, I will discuss related ideas that are most similar to mine. If the topic of my work has many recent related publications, I will categorize these and present each category in a separate section.

For each related idea (or category) I will present a discussion in the following form: (a) what is the idea (in brief)?, (b) what is nice/novel about this idea (category of ideas)?, and (c) what are the problems with this idea? or if there are no problems, how is my approach better compared to this idea?

4 Approach

5 Acknowledgments

The author of this template is thankful to all those wonderful people who worked on this template.

References

1. Koopman, P.: How to write an abstract. Technical report, Carnegie Mellon University (October 1997)
   http://www.ece.cmu.edu/∼koopman/essays/abstract.html.
2. Python Software Foundation: Enumerate (2019) `https://docs.python.org/3/library/functions.html?highlight=enumerate#enumerate`.
3. GeeksForGeeks: Packing and Unpacking (2019) `https://www.geeksforgeeks.org/packing-and-unpacking-arguments-in-python/`.
4. Pierce, B.C., Casinghino, C., Gaboardi, M., Greenberg, M., Hriţcu, C., Sjoberg, V., Yorgey, B.: Software foundations. Webpage: http://www. cis. upenn. edu/bcpierce/sf/current/index. html (2010)