Introduction

In recent years, the magazine *Communications of the ACM* [http://cacm.acm.org/magazines/] has republished two computer-science research articles in each issue. It can be difficult for a computer scientist in one field to appreciate new research from another field because the work may use unusual terminology/notation or rely on other results that are not widely known. Therefore, each paper is preceded by a “Technical Perspective,” which is written by an expert (other than the authors) for non-experts. This short summary helps to place the paper within a larger research context, explaining the importance of the general research area and the contributions of this particular research paper.

For this assignment, you will take the role of the Technical Perspective author for a research paper. There are three stages, each of which is more challenging than it may appear:

- Choose an appropriate research paper
- Understand the paper and identify the paper’s contributions over previous research
- Write a 1–3 page Technical Perspective

Stage 1: Choosing a Research Paper

Due: November 18 (Wednesday) @ 11:59pm; earlier encouraged

At the end of this assignment is a list of research papers from which you can choose. This list has the following significant biases:

1. Concepts and technical machinery from class are *necessary* understand them.
2. Concepts and technical machinery from class course are *not sufficient* to understand them. That is, you will likely need to learn additional concepts on your own via additional reading.
3. Recent papers (the last few years) on topics of increasing importance (concurrency, scripting languages) are over-represented.

There are many other papers that could have been included even given these biases. Thus, while most students will probably select a paper from this list, you may choose a paper not on the list, *with instructor permission before the due date*. You can either find a paper on your own, by skimming the proceedings of programming-languages research conferences, or you can work with the instructor to find a paper on a topic that interests you. Note, however, that biases 1 and 2 above are essential — you need a paper that at least indirectly relies on formal semantics, type systems, or some other topic in the course. In short, pick a paper that you could not have understood without the course.

Choose a paper by reading paper abstracts and skimming or reading papers that sound interesting. Do not panic if much of a paper is impenetrable on the first reading — see Stage 2.

Duplicates: This is an individual assignment. While multiple students may choose the same paper, students that choose the same paper may not work together or discuss the paper. On the other hand, students that choose different papers may discuss the papers together and proofread/critique each other’s “Technical Perspective”s. Indeed, informally “pitching” your paper to another student may be a good way to organize your thoughts for your “Technical Perspective”.
**Requirements:** E-mail your paper choice (authors, title, and venue) to the instructor. (Remember, you must obtain instructor permission (before the due date) to choose a paper that is not on the list.)

**Grading:** There is no formal grade for Stage 1; however, failing to submit a paper choice or changing your paper choice later will incur a 10% penalty on the “Technical Perspective” assignment. This policy is to encourage you begin examining papers early and in sufficient detail that you will not regret your paper choice.

**Stage 2: Understanding Your Paper and Its Contributions**

**Due:** December 02 (Wednesday) @ 11:59pm; earlier encouraged

For Stage 2, you have two goals:

- Thoroughly understand the paper
- Understand the contribution(s) of the paper

While the course has given you a solid foundation in programming-language semantics, a gap remains between the classic concepts you have learned and the state-of-the-art. In short, you are unlikely to be able to read your paper front-to-back. To find appropriate background reading, consider several strategies:

- Your paper cites previous papers. Identify which of those are most likely to provide the background you need. Continue following references transitively until you find what you need.
- Search the web for tutorials and explanations.
- Ask the instructor questions about specific topics. Good questions would be, “What is an open class?” or “Do you know any tutorials on monads?” or “I understand Section 3.1 is about X but then the first sentence of Section 3.2 is completely opaque – can you help?”

**Requirements:** Submit a document to the Understanding Dropbox on MyCourses. You may submit whatever evidence you want, provided that:

- It is approximately one page, and definitely not more than two.
- It makes a convincing case that you have read the paper and understand the vast majority of it.

An outline of the paper and list of contributions is a natural approach. It is not necessary to use complete sentences. You might also list what other papers and references you found most useful.

**Grading:** There is no formal grade for Stage 2; however, failing to submit a document will incur a 10% penalty on the “Technical Perspective” assignment. This policy is to encourage you understand your paper early, with sufficient time left to write a good technical perspective.

**Stage 3: Write Your Technical Perspective**

**Due:** December 11 (Friday) @ 11:59pm

Your “Technical Perspective” must be between 1 and 3 pages long (details below). Writing concisely should be more difficult than writing a longer paper. Treasure your reader’s time, with each sentence being interesting and essential. Convey all the main ideas and contributions of the paper. Note that you will necessarily omit many (if not all) details of the paper’s technical development. At its core, your “Technical Perspective” must communicate the key result(s) of the paper, along with sufficient background and details to explain why the result is interesting. A “Technical Perspective” is almost entirely the “what” and the “why” of the paper, and almost none of the “how”. (But, you will need a firm grasp of the “how” in order to present the “what” and the “why”.)

The pretend audience is a senior studying computer science who has not taken the course. That is, you can assume your audience is a decent programmer with a good education, but you should be very wary of jargon or technology that would be known only to programming-languages experts. In contrast, the research paper you are writing about does make such assumptions, since it was written for a more expert audience. Hence, a “Technical Perspective” is providing real value by making the ideas in the work more accessible.
The actual audience is the instructor, who wants to see that the course has given you the ability (1) to learn more about programming-languages research and (2) to communicate what you learn to others.

Requirements: Submit a document to the Technical Perspective Dropbox on MyCourses. Your document must be typed, single-spaced, single-column, 12pt font, 1in margins, 1 page minimum length, 3 page maximum length, and submitted in PDF format. Your “Technical Perspective” must be in clear English prose, utilizing proper spelling and grammar

Recommended Research Papers


Recommended Conferences

- ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)
- ACM SIGPLAN International Conference on Functional Programming (ICFP)
- ACM SIGPLAN International Conference on Object-Oriented Programming Systems Languages and Applications (OOPSLA)
- ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)
- European Conference on Object-Oriented Programming (ECOOP)
- European Symposium on Programming (ESOP)