

Discussion Sections

The course has two sections that will be dedicated to the discussion of classic papers. The participation in each discussion is worth 5 points.

The Calendar

The discussions will happen in the classroom, at class time (obviously...). Each discussion will last throughout the entire class. For the dates, check the course's [work plan](#). The following papers will be discussed at the following days:

1. First paper: [A linear time algorithm for placing phi-nodes](#).
2. Second paper: [Divergence Analysis](#).

How does the discussion work?

Each discussion is worth five points to the student that participates. To get three of the points, the student must send to the instructor a summary of the paper, answering each of the questions posed below. The summary must be turned in right after the discussion. It can be typed, or handwritten. Students can talk to each other about the papers, but copies are not accepted. **The student must acknowledge the name of each colleague with whom he has discussed the paper in their report.** To get the other two points, the student must participate in the debate. The debate works as follows: the instructor randomly assigns topics to each student, right in the beginning of the class. Topics are discussed in the order given below. A student must provide concise answers to all the topics that he/she receives. Just reading the answer in the summary is fine. Other students can (and are encouraged to) add new points to those previously mentioned by the debater. The discussion is pretty informal: we will have no time for setting up slides. At most, students can use the whiteboard if they think it will make clear the point that they want to make. The list of topics below contains examples of answers that are acceptable.

1. The problem that the paper discusses
 1. What is the specific problem that the paper solves.
 - *The paper solves partial redundancy elimination with profiling information. Basically we want to move code away from redundant paths, using a profiler to assign a cost to each change that we perform in the program code.*
 2. Why is this problem important
 - *Partial redundancy elimination is important because it tends to speedup code. Moreover, by leaving this task to the compiler, the developer is free to focus on algorithms, instead of on micro-optimizations. A testimony of this important is the fact that PRE is taught in many compiler textbooks, and virtually every compiler implements some variation of it.*
 3. Who will benefit immediately from the solution of this problem.
 - *The main profiler is the compiler's back-end. This optimization may be implemented immediately in industrial strength compilers. In particular, this optimization was tailored to just-in-time compilers; hence, we believe that it can be used to improve the user experience that interpreters of dynamic languages such as Python and JavaScript provide.*
 4. What is the theory upon which the problem is defined?
 - *The paper is grounded by the theory of data-flow analyses. Indeed, throughout the paper the authors assume that the reader is familiar with concepts such as lattices, available expressions and very busy expressions. There is also a bit of graph theory in the paper, as the*

authors assume that the reader is familiar with the min-cut problem, and with maximal flow networks.

2. The context of the paper

1. What is the general context of the paper?

- *This is a compiler's paper. It deals with partial redundancy elimination that is a classic compiler optimization. In an ideal world, compilers should take care of micro-optimizations; hence, freeing developers of this details.*

2. Since when is this context source of research?

- *Partial Redundancy Elimination has been around for, at least, four decades. Profiling became a big part of the compiler's technology in the late eighties, early nineties, with the advent of the www, and the rise of the interpreted languages.*

3. Is there any book that provides an in-depth overview of this problem?

- *Yes. As an example, we can find about partial redundancy elimination in the Dragon's book, chapter 9. In that book, PRE is used to illustrate how different data-flow analyses can be combined together.*

4. What can we find about this context in wikipedia?

- *There is an entire article about partial redundancy elimination in the wikipedia, available at <http://www.XXX>. There is also an article about profiling at <http://www.XXX>*

3. The Solution

1. What is the key idea used to solve the problem?

- *The authors have developed a new algorithm based on the min-cut problem to solve partial redundancy elimination with profiling. They assign weights to the edges of the data-dependence graph of a program, and then find a min-cut in this graph. This cut determines the points where expressions should be moved in.*

2. Why is this specific idea different from what had been done before?

- *No particular paper seems to have addressed the problem of doing partial redundancy elimination in SSA form programs using profiling information. By joining these two ideas: SSA form and profiling, the authors proposed something new.*

3. Is there any algorithm involved in the solution?

- *There is the algorithm to solve the min-cut problem in directed graphs. This algorithm has polynomial time solution. It models the min-cut problem as a flow problem.*

4. Is the solution exact, or does it approximate an optimal? In the latter case, what would be the price of finding the optimal?

- *The solution is exact. Of course, the exact solution relies on profiling information. However, profiling is only good if it reflects the actual future runs of a program.*

4. The Organization of the Paper

1. How was the abstract organized?

- *The abstract is organized in the traditional "context - problem - solution - results" style. The context goes from XX to YY. The problem goes from XX to YY. The solution goes from XX to YY, and the results go from XX to YY.*

2. How was the introduction organized?

- *The introduction is organized in six paragraphs, that also follow the "context - problem - solution - results" style. The authors have also talked quite a lot about related works, explaining why their work deserves to be published. I believe they chose this strategy because they are attacking an old problem, and they need to convince the readers that this problem still offers room for innovation.*

3. What is discussed in each section of the paper?

- *The paper has six sections, which are divided as follows: - 1 Introduction - 2 Background - 3 Description of the solution - 4 Comparison with most similar approach - 5 Experiments - 6 Conclusion and future work.*

4. What was left for the conclusion?

- *In the conclusion the authors repeated most of the contents of the abstract, this time in the past. They re-emphasized the main contributions of the paper, and mentioned some future work that they would like to pursue.*

5. The Written Style

1. Can you give a title to each paragraph in the introduction?
 - *1) The need for speed in dynamic languages; 2) Why achieving this speed without dynamic information is so difficult; 3) The previous attempts to integrate PRE, SSA form and profiling; 4) Our key contribution: PRE + SSA + profiling 5) How the min-cut algorithm solves our problem; 6) Summary of experiments.*
2. Can you find examples of sentence topics to every paragraph in the introduction?
 - *In the first paragraph, we have the sentence topic in the first clause: "Partial Redundancy Elimination is one of the most important compiler optimizations". In the second paragraph*
3. Can you give examples of techniques used to connect different paragraphs?
 - *The authors sometimes mention a challenge in the last sentence of the paragraph, and talk about this challenge in the next paragraph. They also start some paragraphs but repeating the idea in the previous, but this time linking it with other concepts. For instance, in the third paragraph of the intro, we have "While there is a consensus that dynamic languages require better compilation technologies, the compiler writers seem to have been unable to apply the most successful techniques described in the literature onto these languages."*
4. Can you find examples of declarative, illustrative and enumerative paragraphs?
 - *Yes. The first paragraph of the introduction is clearly a declarative paragraph, because the authors start it with a thesis topic, e.g., "Partial Redundancy Elimination is one of the most important compiler optimizations", and then provide arguments that justify this thesis. The second paragraph of section 2 is illustrative. The authors are, in this case, explaining how classical PRE is unable to cope with a specific program. Finally, the first paragraph of the related works section (Section 4) is enumerative, as the authors are listing a number of previous attempts to solve the PRE problem in dynamic languages.*

6. The related works

1. What is the purpose of the related works section in this paper?
 - *The main purpose of the related works section is to compare some previous attempts to solve PRE with the proposed solution. The authors do not provide a general overview of the field. They also do not talk about the history of the problem. They start with the most recent solutions, and only talk about those that most closely compare to theirs.*
2. What are the earliest papers about this problem?
 - *The earliest paper about partial redundancy elimination seems to have been "Lazy Code Motion", by Knoop et al., but the authors do not talk about this paper in particular. I had to find it in the description of PRE, available in the wikipedia.*
3. What is the most seminal paper in this field of research?
 - *The most seminal paper seems to have been the first, "Lazy Code Motion". It has, as of March 13th of 2013, 615 citations in google scholar. Another paper that seems relatively important in this field is "A Variation of Knoop, Ruthing, and Steffen's Lazy Code Motion", which has 361 citations.*
4. Which good conferences have recently published papers about similar problems?
 - *Partial Redundancy Elimination is not exactly a hot topic of research. None of the good conferences that I know have published papers about this topic in the recent years. The references used in the paper also only*

contain old publications. The most recent is a paper by XX et al., "XXX", which was published in ASPLOS in 2005.

7. Validation

1. Which points do the authors try to prove with experiments?
 - *The authors try to show that their algorithm can speedup programs, when compared to previous approaches to partial redundancy elimination. They try to show two points: their algorithm yields better performance (Table 1 and Fig. 11), and their algorithm yields smaller code sizes (Fig 12).*
2. Are the experiments rigorous enough?
 - *I think the authors could have been more rigorous. They have not used confidence intervals in their results. They also have tried their approach in only one compiler, and have compared it against an algorithm that they have implemented. It would have been better, in my opinion, if they had tried to compare their algorithm against some PRE implementation available in an industrial quality compiler.*
3. Which visual resources have the authors used to present data?
 - *They use two figures with bar charts, so show speedup and code size reduction. This data is repeated in a table. The table provides results with one decimal digit of accuracy. I believe that they have used this redundancy to make sure that the reader can understand the amount of performance improvement that the new technique delivers.*
4. Which statistical theory have the authors used in this paper?
 - *The authors do not use much statistics in this paper. They have run each benchmark five times, and have thrashed the slowest and the fastest results. The final result is the average of the three left runs. They use arithmetic mean in all the results.*

8. Resources

1. Do the authors use any particular type of notation?
 - *They use pseudo-code to represent the algorithms. The style resembles a bit Algol. I found it a bit confusing to follow the pseudo-code.*
2. Which examples have the authors used to present their ideas?
 - *The paper uses only one example, that is introduced in the backgrounds section, and then is used throughout the rest of the manuscript, notably in Section 3. The example is a simple program that provides some opportunities of optimization. I do not think they could have made it any simpler. The example has the virtue of showing how the proposed approach yields better results in face of profiling information.*
3. Which visual resources have the authors used to explain their points?
 - *The authors use pseudo-code, and three elaborate figures. Fig 3 shows the dependence-graph of the running example. Fig 4 and Fig 6 illustrates different phases of the algorithm. Fig 4 shows the weights that the algorithm finds, given the profiling information, and Fig 6 shows the final solution of the problem.*
4. (Only in case there is on-line material available to support the paper) Which material is publicly available?
 - *The authors have made available a small prototype of their implementation. I tried to download, but it is not easy to install. Simply typing 'making' in the working directory yields a list of errors. They have left a small README.txt file that explains how to test their approach.*