Experimental Results

How to Write a Paper

What are Experiments Good For?

How should we organize the Experimental Results Section?

Research Questions

- How effective is the approach?
- How precise is the approach?
- How fast is the approach?
- What is the code size overhead imposed by the approach?
- How does the approach compares against the technique of Smith *et al* [Smith00]?
- What is the computational complexity of the approach in practice?

Organization – Example

- Goals
  - State research questions
- Setup
  - Hardware
  - Software
  - Benchmarks
  - Common methodology
- One subsection for each research question
  - Specific methodology
  - Discussion

Goals – Example

4. Experiments

The goal of this section is to show that the technique that we have introduced in this paper is effective and useful. To this end, we shall answer the following research questions:

- **RQ-A**: Is the new algorithm practical enough to be applied on large programs?
- **RQ-B**: What is the average speedup that the proposed technique provides to the programs that we generate?
- **RQ-C**: How does our method compare to the state-of-the-art approach of Cuddles et al?
- **RQ-D**: Which kind of programs can benefit more from the new strategy?

Setup – Examples

```
\subsection{}
\label{sub:setup}
```

4.1 Experimental Setup

```
\paragraph{Software}
```

Software. We have implemented all our algorithms on top of XX v 3.14, revision 16961, downloaded on April 1st of 2015. The operating system used was OS X 64-bits version 10.9.5. Every experiment uses gprof v 4.6.22, released on January 2nd 2019.

```
\paragraph{Hardware}
```

Hardware. Experiments were conducted on an Intel Core i7 with a clock of 1.4GHz, and 4 GB of RAM (DDR3) at 1,600MHz. Hiper threading was disabled to prevent time oscillations.

```
\paragraph{Benchmarks}
```

Benchmarks. We have used different benchmarks to answer each research question. In Section 4.2 we use three applications taken from different publicly available repositories. More details about these applications are provided at that section. in Section 4.3 we use SPEC CPU 2006, and in Section 4.4 we use benchmarks of our own craft.

```
\paragraph{Methodology}
```

Methodology. Experiments involving time in Sections 4.1 and 4.3 report the average of 5 runs. In Section 4.2, we report averages of 32 runs. Results are considered statistically significant within a confidence interval of 99%. We never sample the same program in sequence. Instead, we run each program once, in random order, and repeat this step five times.

Answering Research Questions

```
\subsection{RQ-A: Time}
\label{sub:time}
```

4.2 RQ-A: Time

The goal of this section is to evaluate the running time of our algorithm. In addition to the methodological details explained in Section 4.1, we also report time measurements for SCKarma v1.21. We use SCKarma to give the reader some perspective on the running time of our own tool, given that the SC family of tools is well known in the compiler community.

```
\paragraph{Discussion–
Simple Benchmarks}
```

Discussion–Simple Benchmarks. Figure 23 shows the running time of our tool, and compares it with the running times produced by SCKarma. On the average (arithmetic mean of 16 programs), we obtain 34.40 seconds, whereas SCKarma yields 36.30. The p-value is 0.0023, thus, well within our confidence level of 0.01 (confidence interval of 99%).

```
\paragraph{Discussion–
SPEC CPU2017}
```

Discussion–SPEC CPU2017. Figure 24 shows ...

Planning

1. Define the Research Questions
2. Define the experiments necessary (and feasible) to address each research question
3. Write mock figures (captions only)

Planning

1. Define the Research Questions
2. Define the experiments necessary (and feasible) to address each research question
3. Write mock figures (captions only)

When should we have answers for each one of these items?

Mock Figures

Figure 5: Specific threshold of function terminators for programs in the SPEC CPU 2006 benchmark suite. Light-grey bars show specific thresholds found with the algorithm of Section 3, and dark-grey bars show values found dynamically. The shortest the difference between dark and light bars, the more precise our algorithm.