Christoph Josef Amrein

| | |
|---|---|
| Graduate Candidate | Christoph Josef Amrein |
| Examiner | Prof. Dr. Luc Bläser |
| Co-Examiner | Dr. Felix Friedrich, ETH Zürich, Zürich |
| Subject Area | Software and Systems |

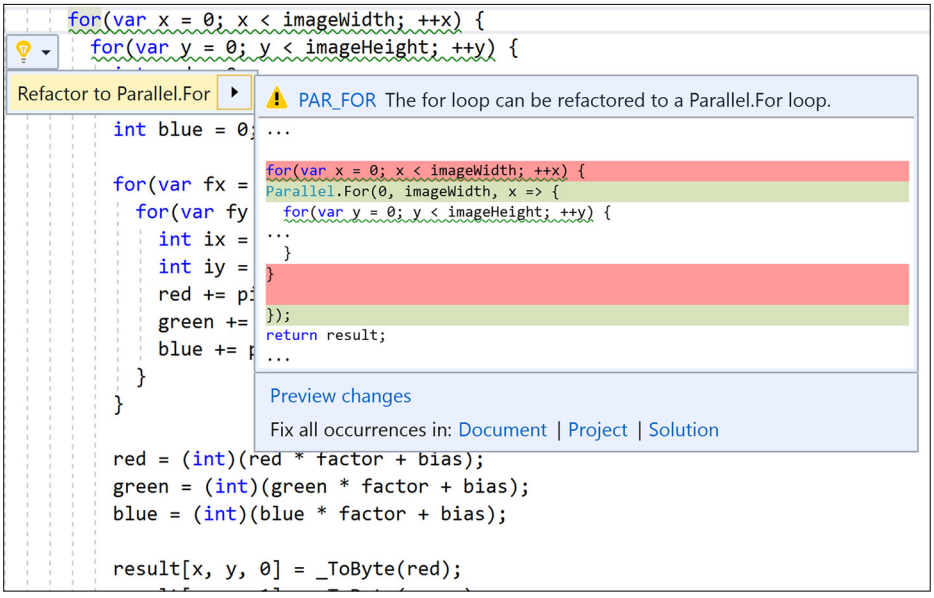
# Automatic Refactoring for Parallelization

```
for(int i = 0; i < upperBound; i++) {
  long current = factorials[1 + i * 2];
  long factorial = 1;

  for(int j = 1; j <= current; j++) {
    factorial = factorial * j;
  }

  factorials[1 + i * 2] = factorial;
}
```

Example code without conflicting array accesses.

```
int[] sum = new int[]
{
  1, 2, 3, 4, 5, 6
};

for(int i = 1; i < sum.Length; i++)
{
  sum[i] = sum[i - 1] + sum[i];
}
```

Example code with conflicting array accesses.




Visual Studio plugin suggesting the refactoring including a preview of the changes.

Introduction: In modern software, adaptions are necessary to fully leverage the parallelization potential. However, the parallelization introduces a new range of possible software faults. Thus, assisting utilities like static code analyzers are desirable. For instance, ones that inform engineers about the code fragments that can be safely adapted.

Objective: This thesis focuses on loops and introduces a conservative approach to verify that these can be parallelized. More specifically, it allows proofing that array accesses do not conflict between iterations. The procedure is a data flow analysis, which proofs the absence of conflicts by employing rules for a selection of binary expressions. Furthermore, its design allows it to profit from various code optimizations automatically.

Result: Experimental evaluations show that both, the prototype and the data flow analysis itself, do not incorrectly identify loops as parallelizable. Moreover, it pinpoints that the analysis can correctly identify most of the parallelizable loops, and only a negligible amount requires a more mature approach.