

Linear time algorithm for placing PHI nodes (Discussion v.0.0.0)

The problem that the paper discusses

What is the specific problem that the paper solves (PHI)

The paper provides a way to place phi nodes (for static single assignment form, SSA) in linear time as opposed to previous methods that require precomputing dominance frontiers.

Why is this problem important (PHI)

Phi nodes are used in SSA form that is beneficial to speed up dataflow analysis on basic blocks, this can be used to find bugs, security flaws and in general make code more robust rather than relying on someone to manually verify it.

Who will benefit immediately from the solution of this problem (PHI)

Dataflow analysis is the cornerstone for code analysis, speeding up the SSA form that makes dataflow analysis faster will help to speedup analysis time. Code analysis is used in coding environments to assist programmers.

What is the theory upon which the problem is defined? (PHI)

Since we are dealing with the internal representation of a program as nodes, inevitable we must deal with graph-theory and graph algorithms.

The context of the paper (X)

What is the general context of the paper? (PHI)

Compilers, static single assignment form used in dataflow analysis for code optimization, bugs and security flaw detections to name a few things.

Since when is this context source of research? (PHI)

It appears dataflow is attributed to Vyssotsky at Bell Labs 1960s. The main paper referenced is Cytron et al from 1989. Based on information from the ACM library can be found Ron Cytron, Jeanne Ferrante, Barry K. Rosen, Mark N. Wegman, and F. Kenneth Zadeck. 1991. Efficiently computing static single assignment form and the control dependence graph. ACM Trans. Program. Lang. Syst. 13, 4 (Oct. 1991), 451–490. [acm link](#) and then this paper appeared in 1995. Before SSA form, use-def chains were used in intermediate representation. *(See question: What is the most seminal paper in this field of research?)*

Is there any book that provides an in-depth overview of this problem? (PHI)

The dragon book mentions SSA form and PHI nodes and it also has a section about calculating dominator nodes as well as dominance frontiers. An algorithm for placing PHI nodes non trivially is not apparent.

What can we find about this context in wikipedia? (PHI)

There is a section on SSA form, but not on linear time placement of phi nodes.

The Solution (X)

What is the key idea used to solve the problem? (PHI)

Identification of key properties of a flow graph of a program. No need for precomputing dominance frontiers and as a consequence of the algorithm the paper presents it calculated the dominance frontier that can be used in other aspects of the phases of dataflow analysis in linear time according to the claims (see algorithm).

Why is this specific idea different from what had been done before? (PHI)

This paper employs the same principle but with a different approach ensuring a linear time instead of a potentially being quadratic.

Is there any algorithm involved in the solution? (PHI)

Yes, there are multiple cleverly algorithms, that use graph traversal and properties of based on graph theory to justify properties on the internal representation of the program to validate their claims.

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

The solution is exact accompanied with a proof of correctness and argument for complexity as compared to the authors argument that the Cytron et al. paper's implementation was complicated and partial description was provided.

The Organization of the Paper (X)

How was the abstract organized? (PHI)

“Context - Problem - Solution - Comparison - Results”

How was the introduction organized? (PHI)

Six paragraphs of:

1. Context/(Current techniques)
2. Problem
3. Solution
4. Notation/(Internal details)
5. Implementation/Results
6. Implications/(Sections overview)

This paper aims to argue for improvement of a step in the SSA form used in the seminal paper on intermediate representation, this is why a lot of detail is given in comparison to the seminal paper.

What is discussed in each section of the paper? (PHI)

The paper has eight sections, which are divided as follows:

1. Introduction
2. Background and Notation
3. DJ Graphs and their properties
4. Algorithm for placing phi nodes
5. Correctness and Complexity
6. Implementation and Experiments
7. Related work
8. Conclusion

What was left for the conclusion? (PHI)

Reiteration of the introduction and emphasis on the simplicity of technique used as well as implications, implementation, results and speedup. Ending with references to companion papers where solution is used in other dataflow problems.

The Written Style (X)

Can you give a title to each paragraph in the introduction? (PHI)

- 1) The phases of dataflow where placing of phi nodes is important;
- 2) Identification of redundancy in present technique;
- 3) Optimization in the algorithm for placing phi nodes;
- 4) Introduction to notation used in solution;
- 5) Implementation and comparison of both techniques;
- 6) Potential further usecases and overview of sections;

Can you find examples of sentence topics to every paragraph in the introduction? (PHI)

1. The algorithms for computing these intermediate representations have one common step- computing program points where data flow information must be “merged”, the so called **phi-nodes** *(2nd sentence)*
2. The time complexity of the original algorithm depends on the size of the dominance frontier. Although the size of the dominance frontier is **linear** for many programs (as was noted by Cytron et al.), ... *(1st sentence)*
3. In this paper, we present **a linear time algorithm for computing the desired set of phi-nodes for N**. without precomputing the dominance frontiers for all the nodes. *(1st sentence)*
4. The levels of the nodes in the dominator tree are used to order the computation of dominance frontiers of those nodes which are essential to **compute the final set of phi-nodes** in a bottom-up fashion. *(3rd sentence)*
5. Again our algorithm exhibited **a linear behavior** compared to the quadratic behavior of the original algorithm. *(6th sentence)*
6. The significance of the algorithm presented in this paper goes beyond to merely computing **phi-nodes** for SEGS or SSA form. *(1st sentence)*

Can you give examples of techniques used to connect different paragraphs? (PHI)

The discussion of an aspect of the paper topic in one paragraph flows into the second paragraph where the aspect is touched on in a different light but also the issue presented in the previous paragraph is elaborated on. For example:

1. Again our algorithm exhibited a linear behavior compared to the quadratic behavior of the original algorithm. *(Section 1, paragraph 5)*
2. The significance of the algorithm presented in this paper goes beyond to merely computing phi-nodes for SEGS or SSA form. *(Section 1, paragraph 6)*

Can you find examples of declarative, illustrative and enumerative paragraphs? (PHI)

- Declarative
 - (Section 1, Paragraph 1):
 - * (Sentence 1) Static Single Assignment (SSA) form [CFR+ 91], Sparse Evaluation Graphs (SEGS) [CCF91], and other related intermediate representations have been successfully used for efficient data flow analysis and program transformations. . . *(Continues on to provide support with reference to other works.)*
- Illustrative
 - (Section 1, Paragraph 2):
 - * (Sentence 2) As Cytron and Ferrante pointed out “Since one reason for introducing ~-nodes is to eliminate potentially quadratic

behavior when solving actual data flow problems, such worst case behavior during SEG or SSA construction could be problematic. (*Illustration with reference to seminal paper that supports this papers work.*)

- Enumerative
 - (Section 6, Paragraph 3):
 - * (*Enumeration of comparision of implementation in bit-vecotr and linked list.*)

The related works (X)

What is the purpose of the related works section in this paper? (PHI)

Points out that present techiniques present a quadratic time issue, identifies similar work in this endivure but highlights that while other papers provide same results but they do not compute for example dominance frontiers (or are more dense that slow down dataflow) that can be used in other phases of dataflow analysis. Identifies the similariteis with Cytron et al. paper and clearly identifies the difference they have made. Notify potential other work that can benefit from this speedup and calculating intermediate representations this paper uses.

What are the earliest papers about this problem? (PHI)

The paper specifically mentions Cyton et al. in 1989 as first propose of the algorithm, while other work on sparse analysis is also mentioned as 1982 and 1987.

What is the most seminal paper in this field of research? (PHI)

An Efficient Method of Computing Static Single Assignment Form 1989 (Cytron, ...) 310 citation that proposed the new algorithm that is referenced heavily in this paper.

The work for SSA form is mention in Cytron et al.:

- J. Ferrante, K. J. Ottenstein, and J. D. Warren. The program dependence graph and its use in optimization. ACM pans. on Programming Languages and Systems, 9(3):319-349, July 1987. (*1921 citation*)
- B. Alpern, M. N. Wegman, and F. K. Zadeck. Detecting equality of values in programs. Conf Rec. Fifteenth ACM Symp. on Principles of Programming Languages, January 1988.
- B. K. Rosen, M. N. Wegman, and F. K. Zadeck. Global value numbers and redundant computations. Conf. Rec. Fifteenth ACM Symp. on Principles of Programming Languages, January 1988.
- Taken from: The Way of the Computer Scientist for dataflow analysis
 - *Vyssotsky at Bell Labs 1960s for dataflow analysis*

– *Kildall 1973 paper, work by Hecht and Ullman for Iterative dataflow analysis*

Wikipedia also has many reference with Kendall at the top among many others.

Which good conferences have recently published papers about similar problems? (PHI)

- Sebastian Buchwald, Denis Lohner, and Sebastian Ullrich. 2016. Verified construction of static single assignment form. In Proceedings of the 25th International Conference on Compiler Construction (CC 2016). Association for Computing Machinery, New York, NY, USA, 67–76.
- Leonardo Filipe Rigon, Paulo Torrens, and Cristiano Vasconcellos. 2020. Inferring types and effects via static single assignment. Proceedings of the 35th Annual ACM Symposium on Applied Computing. Association for Computing Machinery, New York, NY, USA, 1314–1321.
- David Ittah, Thomas Häner, Vadym Kliuchnikov, and Torsten Hoefler. 2021. QIRO: A Static Single Assignment-based Quantum Program Representation for Optimization. ACM Transactions on Quantum Computing Just Accepted (August 2021).

Validation (X)

Which points do the authors try to prove with experiments? (PHI)

The speed of the algorithm for different programs as well as scaled on program size compared to the original algorithm. A proof is provided in the previous section.

Are the experiments rigorous enough? (PHI)

I think the proof coupled with the several programs in contrast to the original algorithm as well as provided curve with program size serves enough evidence. One needs to understand how the algorithm works and be satisfied with the proof to agree with them more.

Which visual resources have the authors used to present data? (PHI)

Tables as bar graphs as well as raw values for different programs, curves (on performance with program size). Both in comparison with the original algorithm.

Which statistical theory have the authors used in this paper? (PHI)

Raw performance figures with the calculated speedup and proofs as used.

Resources (X)

Do the authors use any particular type of notation? (PHI)

Simple standard psuedo-code.

Which examples have the authors used to present their ideas? (PHI)

The optimization of an algorithm (by Cyton et al.) that places phi nodes with dominance frontiers is talked out extensively in the introduction and is hashed out in the Background and notations section.

Which visual resources have the authors used to explain their points? (PHI)

Use of Mathematical notation for graph algorithm accompinied with example figures of the graphs before and after the computation. Psuedo-code as well as definitions of data structures used accompinied with example figure that shows the phi node insertion on the data structure after computation. Figure 5, 6, 7 and show the performance details.

**(Only in case there is on-line material available to support the paper)
Which material is publicly available? (PHI)**

N/A.