Pimpri Chinchwad Education Trust's

Pimpri Chinchwad College of Engineering

Major Project Synopsis

Department: Computer Engg. **Academic Year:** 2024 -2025 **Semester:** I

Year: B. Tech. Scheme A **Div: A** **Group ID: GA3** **Date: 7th August 2024**

Problem Statement: (Write here descriptive title)

Interactive tool for Static Single Assignment (SSA) Visualization for Compilers

Project Domain and SIG: Machine Intelligence and Optimization Algorithms

Whether it is inclined towards Either of following National thrust areas or others : HEALTH & Hygiene, AGRICULTURE, ENERGY SECTOR, TRANSPORT SECTOR, COMMUNICATION & NETWORKING, **QUALITY EDUCATION**, SECURITY and Public SAFETY , E-Commerce, E-GOVERNANCE **(Pl tick the relevant Thrust Areas)**

Team Members:

PRN No	Name of Student	Mail-id	Contact No.	Signature
121B1B006	Aditya Agre	aditya.agre21@pccoepune.org	7378555854	
121B1B013	Yash Athawale	yash.athawale21@pccoepune.org	8857853345	
121B1B017	Siddhi Bajpai	siddhi.bajpai21@pccoepune.org	7841841344	
121B1B032	Amay Chandravanshi	amay.chandravanshi21@pccoepune.org	8805274895	

Sponsorship if any - Company Details with address: NA

Name of External Guide: Dr. Uday Khedkar

Interdisciplinary Project with other department students (If yes write department and guide name of that department) : NA

Abstract:
Static Single Assignment (SSA) form is a crucial intermediate representation used in modern compilers to facilitate various optimization techniques. This paper presents an interactive tool designed for the visualization and understanding of SSA. The tool aims to aid both students and professionals in comprehending the transformations and benefits associated with SSA. Through this tool, users can gain deeper insights into SSA's role in compiler design, enhancing both learning and practical application in optimizing compilers. The interactive nature of the tool bridges the gap between theoretical concepts and practical implementation, making it an invaluable resource for compiler education and development.

Related Work:

1. **Compiler Explorer:**
   Compiler Explorer is an interactive online compiler which shows the assembly output of compiled C++, Rust, Go (and many more) code.
2. **Vlabs:**
   Virtual Labs (VLabs) by the Ministry of Human Resource Development (MHRD) in India is an initiative aimed at providing remote access to laboratory experiments for students and educators. This platform offers a wide range of interactive simulations and practical experiences in various scientific and engineering disciplines, enabling learners to perform experiments and understand complex concepts without the need for physical lab infrastructure.

Innovative concept and relevance of the topic:

Innovative Concept:

It offers a step-by-step visualization of code conversion into SSA form, complete with variable renaming, dominance frontiers, and phi-function insertion. The tool's interactivity allows users to engage with the material actively, modifying code and instantly observing the resulting SSA transformations. This hands-on, visual learning method significantly enhances comprehension and retention, making SSA concepts more accessible and less intimidating.

Relevance:

The relevance of this tool lies in its ability to address a critical need in both education and professional development within the field of compiler design. For students, understanding SSA form is essential for grasping advanced optimization techniques and compiler internals. Traditional teaching methods often fall short in conveying these complex ideas effectively. The interactive tool fills this gap by providing a dynamic and engaging learning experience.

Market potential and competitive advantage:

Market Potential:

1. Educational Institutions: Enhances compiler design courses at universities and coding bootcamps.
2. Professional Training: Aids companies in training and upskilling employees in compiler optimization.

Competitive Advantage:

1. Visualization: Clear, visual representations of SSA transformations and optimizations.
2. User-Friendly Interface: Accessible to both novices and experts.

Project Objectives: Industry/ Product/ Research/Societal (Min 3)

1. Develop a robust and user-friendly tool that facilitates the understanding of SSA form for students, educators, and professionals in the field of compiler design.
2. Democratize access to advanced compiler design education by providing a free or low-cost tool that can be used in various educational institutions, including those with limited resources.
3. Contribute to the academic and professional community by offering a tool that supports detailed analysis and research of SSA-based optimizations and transformations.

Technical Details (Platform and languages):
Web-based Application: The tool will be accessible through modern web browsers, ensuring cross-platform compatibility (Windows, macOS, Linux).

Cloud Infrastructure: Utilizes cloud services for scalable backend processing and data storage, ensuring performance and reliability.

Languages: TypeScript, Node.js, Python, C, MongoDB, HTML5, CSS3, Git

Technical Keywords (Ref ACM Keywords): (Ref web: www.acm.org/about/class/ccs98-html)
•Software and its engineering~Software notations and tools~Development frameworks and environments~Integrated and visual development environments

Relevant mathematical models associated with the Project

1. Control Flow Graphs (CFG)
2. Data Flow Analysis
3. Dominance Relations
4. Phi functions
5. Lattice Theory
6. SSA Construction Algorithms

Targets from project: (Discuss with guide and tick)

-Paid Consultancy project
-Sponsored Project
-Scopus/SCI Paper Publication
-Patent
-Project competition and awards

Plan of the conference/journal (Such as IEEE/Springer/Scopus Journal) where paper will publish or Patent/ Copyright of project.
Indian Patent of Software Tool

References: List of Conference/Journal Papers supporting project idea (at least 10 papers + white papers or web references)

1. Cytron, R., Ferrante, J., Rosen, B. K., Wegman, M. N., & Zadeck, F. K. (1991). *Efficiently computing static single assignment form and the control dependence graph*. ACM Transactions on Programming Languages and Systems (TOPLAS), 13(4), 451-490.
2. Appel, A. W. (1998). *SSA is functional programming*. ACM SIGPLAN Notices, 33(4), 17-20.
3. Aycock, J., & Horspool, R. N. (2000). *Simple generation of static single-assignment form*. In Compiler Construction (pp. 110-125). Springer, Berlin, Heidelberg.
4. Price, B. A., Baecker, R. M., & Small, I. S. (1993). *A principled taxonomy of software visualization*. Journal of Visual Languages & Computing, 4(3), 211-266.
5. Brown, M. H., & Sedgewick, R. (1984). *Techniques for algorithm animation*. IEEE Software, 1(1), 28-39.
6. Roman, G. C., Cox, K. C., Wilcox, D., & Plun, T. (1992). *Puzzle: A visual programming environment for parallel programming*. In Visual Languages, 1992. Proceedings., 1992 IEEE Workshop on (pp. 216-224). IEEE.
7. LLVM Project. *LLVM Language Reference Manual*. Available at: LLVM
8. Muchnick, S. S. (1997). *Advanced Compiler Design and Implementation*. Morgan Kaufmann Publishers.
9. Appel, A. W. (2004). *Modern Compiler Implementation in C/Java/ML*. Cambridge University Press.

10. Allen, F. E., & Kennedy, K. (2002). *Optimizing Compilers for Modern Architectures: A Dependence-based Approach*. Morgan Kaufmann.
11. Pizlo, F., & Vigna, G. (2005). *The Importance of SSA in Static Analysis and Program Transformation*. In Proceedings of the ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation (pp. 19-30).
12. Kearney, P., & Morrison, J. (1998). *Visualizing data structures and algorithms*. IEEE Software, 15(2), 56-63.
13. Stasko, J. T. (1997). *Tango: A framework and system for algorithm animation*. IEEE Computer Graphics and Applications, 17(2), 42-49.

Aditya Agre	121B1B006	
Yash Athawale	121B1B013	
Siddhi Bajpai	121B1B017	
Amay Chandravanshi	121B1B032	

Prof. Rahul Pitale
Project Guide