
Homework 3 - Translating Bril SSA to LLVM IR

CSIE 5054 - Advanced Compiler Design

National Taiwan University

Total Points: 8
Release Date: 2024/11/08 08:00
Due Date: 2024/11/29 23:59
TA Email: llvm@csie.ntu.edu.tw

Contents

1 Announcement	1
2 Preliminaries	1
3 Problem 1	2
4 Plagiarism and Academic Integrity	4

1 Announcement

1. The homework contains only 1 problem.
2. In case you encounter technical issues with Homework 3, kindly consult online resources initially, as most problems are likely related to your local environment. If challenges persist, reach out to our TAs following these guidelines:
 - Title your email [AC-HW3][Summary-Of-Your-Issue]. Please note that we will **NOT** receive emails in other formats as we have applied filters to our email system.
 - Provide detailed information about your computer, including the operating system.
 - Outline the methods you attempted previously, the resources you consulted, the steps you followed, and the results of your efforts.
 - Note that ambiguous requests, such as attaching screenshots without proper descriptions, will not be answered. **Such emailing will lower your priority.**
 - For guidance on how to formulate effective technical inquiries, please refer to How To Ask Questions The Smart Way. This resource can help you structure your questions to get quicker, more precise responses from the TAs.

2 Preliminaries

- Git
- GitHub
- Bril Tool-chain
- LLVM

3 Problem 1

In this homework, you will implement a translator that converts a Bril program in SSA form into LLVM IR. The goal is to produce a valid LLVM IR program that can be executed using `lli`, and whose output matches that of the original Bril program when executed with `brili`. You will build upon your SSA construction from Homework 2 and focus on the translation process to LLVM IR.

3.1 Task Details

3.1.1 Overview

1. **Set Up the Environment:** Ensure your development environment has LLVM 17 and Bril toolchain installed.
2. **Import SSA Implementation:** Copy your working SSA construction code from Homework 2.
3. **Implement LLVM Translation:** Complete the LLVM IR generation module.
4. **Validate Generated Code:** Ensure the LLVM IR is valid and produces correct output.

3.1.2 Steps

Below are the detailed steps to guide you through the homework:

1. **Accept the Assignment and Clone the Repository**

```
1 git clone --recursive <TBA>
2 cd <homework-directory>
```

2. **Review the Starter Code:** The repository contains the following structure:

```
homework-directory/
├── src/
│   ├── driver.py
│   ├── bril.py
│   ├── cfg.py
│   ├── dominance.py
│   ├── ssa_construct.py
│   ├── ssa_to_llvm.py
│   └── [other source files]
├── tests/
│   ├── loop.bril
│   └── [additional test cases]
├── bril/
├── install_bril.sh
├── run_test_case.sh
├── student_id.txt
└── README.md
```

3. **Copy Your SSA Implementation:** Copy your working SSA construction files from Homework 2 into the `src/` directory and verify they work correctly.
4. **Implement LLVM IR Translation in `ssa_to_llvm.py`:**
 - Handle basic Bril operations (`const`, `add`, `mul`, etc.)
 - Translate control flow structures (`br`, `jmp`)
 - Convert ϕ -functions to appropriate LLVM IR form
 - Manage variable declarations and types
 - Implement function definitions and calls

3.2 Testing Guidelines

- To generate and test LLVM IR:

```
1 # Generate LLVM IR
2 bril2json < ./tests/[test_file].bril | python3 ./src/driver.py > output.ll
3
4 # Run original Bril program
5 bril2json < ./tests/[test_file].bril | brili [arguments] > original.out
6
7 # Run generated LLVM IR
8 lli output.ll [arguments] > transformed.out
9
10 # Compare outputs
11 diff original.out transformed.out
```

If there is any difference between `original.out` and `transformed.out`, your LLVM IR translation is incorrect.

3.3 Submission Instructions

1. **Add your student ID to `student_id.txt`:** Open `student_id.txt` and replace the placeholder with your actual student ID.
2. **Ensure all your code changes are within the `src/` directory:** Only modify files inside the `src/` directory. **DO NOT** alter other parts of the repository unless explicitly instructed.
3. **Commit and Push Your Changes**

```
1 git add src/ student_id.txt
2 git commit -m "Completed Homework 3"
3 git push origin main
```

4. **Check the GitHub Actions Workflow**

5. **Confirm Grading Output**

3.4 Do and Don't

- You are allowed to modify any part of the starter code within the `src/` directory to suit your approach.
- Make sure you have a solid understanding of LLVM IR and its mapping from Bril IR before starting your implementation.
- Ensure your student ID is correctly entered in the `student_id.txt` file before submission.
- **DO NOT** modify anything outside the `src/` directory except `student_id.txt`. Any such changes will be considered cheating.
- Please note that we will be able to see through the GitHub Classroom backend if you have made changes to files that should not be modified.

3.5 Additional Resources

- LLVM Language Reference Manual
- Bril Documentation

4 Plagiarism and Academic Integrity

It is important to adhere to the university's academic integrity policy while completing this assignment. Please keep the following in mind:

1. **Do Not Share Your Code:** All submissions must be your own work. Sharing your code with others or obtaining code from others, including online resources, is considered plagiarism and will be treated as academic misconduct.
2. **Use of External Resources:** You are encouraged to textbooks, lecture notes, and official documentation to assist your understanding. However, directly copying code or solutions from external sources (e.g., GitHub, StackOverflow, or similar) without attribution is not allowed.
3. **Collaboration Guidelines:** You may discuss general concepts and strategies with classmates, but all coding and detailed design decisions must be completed **independently**.
4. **Consequences of Plagiarism:** Plagiarism, cheating, or any form of academic dishonesty will result in a zero on the assignment, and may lead to further disciplinary action as per university regulations.