CS323 Project 3 Report

Name: 陈宇恒 (Chen Yuheng)

Student ID: 11711111

I didn't implemented any bonus feature in this project. I did a simple optimization by reducing dummy labels.

Main Program

The main program is inherited from the program of project 2.

Based on the program of project 1, I added the **translation functions** to translate different kinds of nodes in the grammar tree. My translation functions are almost the same as what provided in the document.

My representation of TAC is **char string**, quite simple and straightforward.

With the support of the translation functions, by post-order traversing the grammar tree from the root node, I can get the TAC.

Optimization

Reducing the dummy labels

With the very naïve translation schemes, it's frequently that dummy labels exists.

For example:

```
1  ...
2  WRITE t7
3  LABEL label6 :
4  LABEL label3 :
5  t8 := #0
6  ...
```

The label `label6` and `label3` in line 3 and line S4 denote the same position of the code.

I wrote a python script `op.py` to reduce such dummy labels. This script read the TAC code and found dummy labels and last reduce them, with substituting the removed labels in the `GOTO` instructions.

This script is invoked after the main program finished, which is invoked by system call of Linux:

```
1  execlp("python3", "python3", "op.py", TAC_path, NULL);
```

By testing, this optimization can make $\sum \#executed\ instructions$ lesser for some SPL programs.