

CSCI 4160 Project3

Due: see class calendar

Goal:

This assignment serves several purposes:

- to be familiar with Bison,
- to create the context free grammar for Tiger language, and
- to create the parser for Tiger language using Bison

Description:

This is a team project that build on the previous project. In this project you are required to use Bison to build a parser for the Tiger language. More specifically, write the context free grammar for the Tiger language. Please read Tiger language manual carefully before you start. The manual can be found in the D2L Content | Module 6: Class Repository | Tiger.

Team Project Requirement:

Please review the following documents in the class repository:

- How to access the team repository
Create an online team repository so that all members can access the project. The repository can be hosted on [GitHub](#), [Bitbucket](#), [Source Forge](#), or [GitLab](#).
- Requirements for team projects
D2L Content | Module 6: Class Repository | team requirements.doc

The team should submit the following documents while working on the project:

1. Time log report.

The time log report is used to monitor the progress of the project. Each team is required to submit 3 time log reports. Each report should specify the major works that has been done and estimated time needed to finish the project,

- The first time log report should be submitted within 2-3 days after the project is assigned. This report should specify the roles of team members, and estimated time needed from each team member.
- One time log reports to monitor the progress of the project (alpha version) and test cases.
- One report to monitor the progress of the final version of the project..

The time log report pattern file can be found at “project3\timelog” folder.

2. Test cases:

The test cases are actually a Tiger program used to test the project (alpha version). The pattern of test case file can be found at: “project3\Testing”

3. Test report

The test report summarizes the detected errors on alpha version of the project. The format of the file can be found at: “project3\Testing”

Important Dates and Deadlines:

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
21 Project assigned	22	23 1st time log report due	24	25	26	27
28 2nd time log report due	29	30 Project (alpha version) and test cases due	1(Oct.)	2 1st test report due	3 3rd time log report due	4
5 Final Project due	6	7	8	9	10	11
15	16	17				

Notes:

- All reports/files are due at 11:30pm on the due date.

What to do in this project?

Come up with context free grammar for Tiger language, and use Bison to generate the parser for it. Your grammar should have as few shift-reduce conflicts as possible, and no reduce-reduce conflicts. When you compile your parser file, Bison will output information about the number of shift-reduce conflicts as well as the number of reduce-reduce conflicts. If your grammar has a reduce-reduce conflict, please talk to me to see how to get rid of it.

Most Tiger CFG can be found in the brief manual located at “D2L Content | Module 6: Class Repository | Tiger | BriefManual.pdf”. However, some changes is required to make it work.

In the tiger.yy file, the precedence level and association of operators should be specified based on the Tiger manual.

Extra rules should be provided to recover from errors. You grammar should generate the same output on the sample file example.tig as the one provided by the instructor.

When defining CFG for Tiger language, please ignore the following features:

- Functions, i.e. function declaration, and function definition. (But you still need to recognize function calls since tiger has built-in functions)
- Records, i.e. record definition and record expression
- All other features that are only tied with functions and records.

Set up environment:

I strongly suggest teams to use the sample Visual Studio solution provided by me in the class repository.

1. Choose a tiger.ll from your team members (typically the one with highest grades). Correct all existing errors. Copy the tiger.ll to project3\BisonProject\MainDriver, and make the following change:

- a. Replace the line `#include "tokens.h"` by `#include "tiger.tab.hh"`. This is required because there is no `tokens.h`, and all token definition will be in `tiger.tab.hh` file, which is generated by Bison.
2. Put your grammars in `tiger.yy` file, which is the only file the team needs to work on.
3. To compile your project,
 - a. Compile `tiger.ll`. This step can be skipped if there is no change since the last compilation of `tiger.ll`.
 - b. Compile `tiger.yy`.
 - c. Build the MainDriver solution.
4. There are two ways to debug your grammar,
 - a. In the sample solution provided by the instructor, there is a file "tiger.output" under "FlexBison Tools"\Resource Files within Visual Studio environment. This file is generated everytime `tiger.yy` is parsed. It contains all state information of your grammars.
 - b. Reset `yydebug` to 1 at line 29 at `main.cpp` file. This will generate debug information on terminal when compiling a tiger file. This approach is probably hard to understand.

Instructor provided files in the class repository

The following files are provided by the instructor:

- Project3\Testing: This folder contains test reports.
- Project3\Timelog: This folder contain all time log reports.
- Project3\AlphaVersion: This folder contains the alpha version of the project. **Please put tiger.yy file only in this folder.**
- Project3\Rubric3.docx: the rubric file for this project.
- Project3\BisonProject: This folder contains all source files and required visual studio files
 - Skeleton source files provided in the sample project are listed below:
 - `tiger.ll`: an empty file. Should be replaced.
 - `lex.yy.c`: generated by Flex when compiling `tiger.ll`
 - `tiger.yy`: a skeleton file for tiger CFG.
 - `tiger.tab.hh` & `tiger.tab.cc`: generated by Bison when compiling `tiger.yy`.
 - `tiger.output`: debug file for CFG
 - `ErrorMsg.h`: contains the definition of error handler
 - `main.cpp`: the driver
 - `example.tig`: test program of tiger language
- Description3.pdf: this file
- Rubric3.doc: the rubric used to grade this assignment.
- example.txt. sample output when parsing example.tig file

How to submit

1. Once you have finished, submit the following files by the due date to D2L dropbox: Project3:
 - **Test case file: testcase.tig**
 - **Testing report: TestReport.docx**
 - **3 timelog reports: 1stReport.txt, 2ndReport.txt, and 3rdReport.txt**
 - **alpha version: alpha_tiger.ll**
 - **final version: Tiger.ll**