Exercise 1

April 2, 2017

1 Introduction

This exercise implements an automatic checker for Linear Algebra assignments. Currently, first year students are given a linear system $Ax = b$, they apply the Gauss Jordan elimination algorithm, and then find the solution set. The purpose of our exercise is to make sure those first year students are doing their job right. For that, we will implement a tool called Lini.

2 Lini :: Example

Lini receives a linear system from the Professor, and a sequence of row operations + solution set from the student. Lini outputs a detailed summary of the check results in a pdf file.

Linear System (Given by the Professor) :: Example

$$\begin{bmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \\ 7 & 6 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \\ 7 \end{bmatrix}$$

Row Operations (Computed by the student) :: Example

$$\begin{aligned} R_1 &\leftarrow \tfrac{2}{5} R_1 \\ R_2 &\leftarrow R_2 - 3R_1 \\ R_1 &\leftarrow \tfrac{2}{5} R_1 \end{aligned}$$

Solution Set (Computed by the student) :: Example

$$\{(3, 2, 3)\} + SP(\{(1, 0, 1)\})$$

3 Lini :: Details

Input Format :: Linear System

The Matrix A and the vector b are accommodated in a single $m \times (n+1)$ matrix, whose rightmost column is b. A matrix entry is either an integer, or a fraction. The linear system resides in a text file, starting with a left bracket, and ending with a right bracket. Entries are separated with either blanks or tabs, and rows are separated with a single semicolon. Fractions are written with either slash or backslash. The entire system must be written as a single line in the text file:

$$[3\ 4\ -6; 7\ -2/9\ -100; 7\ 0\ 0; -6\ 4/3\ 10]$$

Input Format :: Row Operations

The row operations sequence reside in a text file. Each line contains a single row operation. Table 1 describes the context free grammar of row operations. In

S	::=	rowOperationList
rowOperationList	::=	rowOperation rowOperationList
	::=	rowOperation
rowOperation	::=	swapRows
	::=	mulRow
	::=	mulAndAddRow
swapRows	::=	R index leftRightArrow R index
mulRow	::=	R index leftArrow [−] num R index
mulAndAddRow	::=	R index leftArrow R index op [num] R index
index	::=	1 | 2 | 3 | 4
leftArrow	::=	$<-$
leftRightArrow	::=	$<->$
num	::=	int
	::=	int div int
int	::=	0
	::=	$[1-9][0-9]^*$
div	::=	/
	::=	\
op	::=	+
	::=	−

Table 1: Context free grammar for the language of row operations.

addition, the following limitations are imposed on row operations: (1) Rows can not be multiplied by zero. (2) Fractions are always with a non zero denominator. (3) The indices in rule mulRow have to be the same. (4) The first two indices in mulAndAddRow have to be the same.

Input Format :: Solution Set

The solution set resides in a text file. Whenever the system has no solutions, this text file contains a single word: EMPTYSET. When the system has a single solution, it is written as a set with a single row vector $\{(3, -\frac{2}{9}, 200)\}$. If the system has an infinite solution set, it should be written as a sum of two vector sets: $\{(2, -6, 100)\} + \text{SP}(\{(1, 0, 1), (6, 6, 0)\})$.

4 Programming Assignment

Write three lexers for the linear system, row operations and solution set. The exact context free grammar for the linear system and the solution set are missing intentionally. You should support linear systems with up to four rows and five columns (including the solution vector b). In the solution set, you should support row vectors with sizes $= 2, 3, 4$. The span set size is not limited, as vectors inside it can be dependant.

5 Setting Up the Working Environment

The formal environment of the course is Linux Ubuntu 14.04 LTS. Please install it natively on your machine, or use a VM of your choice. Once installed, open the terminal and git clone this repository:

```
https://github.com/OrenGitHub/COMPILATION_IDC_FOR_STUDENTS.git
```

Navigate to the Linux directory inside exercise 1 with

```
cd FOLDER_2_SOURCE_CODE/EX1/LINUX_GCC_MAKE/
```

and run make

6 Submission Guidelines

Open an account on GitHub. Then, visit the academic discount page to enable the free creation of private repositories. One team member should create a new private repository called COMPILATION, and then invite other team members and me as collaborators. My username is OrenGitHub. Please make sure the uppermost folder (COMPILATION) contains a text file "IDS.text" with the IDs of all team members (one ID per line). Throughout the semester, you will create sub-folders EX1, EX2, etc. inside COMPILATION. Please try to keep the directories structure the same as in the course repository. EX1 Should include a makefile building your source to a runnable program called Lini. Note, that currently Lini resides inside `FOLDER_06_Lini` so please have your makefile copy it from there before submitting. To avoid the pollution of EX1, please remove all *.o files once the target is built. The next paragraph describes the execution of Lini.

Execution parameters Lini recevies 6 input file names:

Matrix.txt
RowOperations.txt
SolutionSet.txt
MatrixTokens.txt
RowOperationsTokens.txt
SolutionSetTokens.txt