Exercise 4

Due 26/5/2017

1 Introduction

This exercise uses Bison on top of Flex to check whether the solution set input file has correct syntax. The grammar is given in Table 1. In addition, *all* row vectors have to be of the same size (2, 3 or 4), denominators can not be zero, and denominators can not be negative.

S	::=	solutionSet
solutionSet	::=	{rowVec} + SP({rowVecList})
	::=	SP({rowVecList})
	::=	{rowVec}
rowVecList	::=	rowVec, rowVecList
	::=	rowVec
rowVec	::=	(num, num)
	::=	(num, num, num)
	::=	(num, num, num, num)
num	::=	[op] int
	::=	[op] int div int
int	::=	0
	::=	$[1-9][0-9]^*$
div	::=	/
	::=	$\backslash$
op	::=	$+$
	::=	$-$

Table 1: Context free grammar for the language of the solution set.

2 Bison

Bison is an LALR(1) parser generator, which receives as input a context free grammar, and implements a parser for that grammar in a single C file. An overall example for using Bison is inside the row operations parser.

3 Input

The input for this exercise is a single text file that contains the solution set.

4 Output

The output is a single text file that should contain a single word: either OK when the solution set has correct syntax, or FAIL otherwise.

5 Submission Guidelines

The code for this exercise resides as usual in subdirectory EX4. The file SolutionSet.y, and possibly SolutionSet.lex should be the only file(s) you change. Please submit your exercise in your GitHub repository under COMPILATION/EX4, and have a makefile there to build a runnable program called Lini. To avoid the pollution of EX4, please remove all *.o files once the target is built. The next paragraph describes the execution of Lini.

Execution parameters Lini recevies 2 input file names:

Input.txt
Output.txt