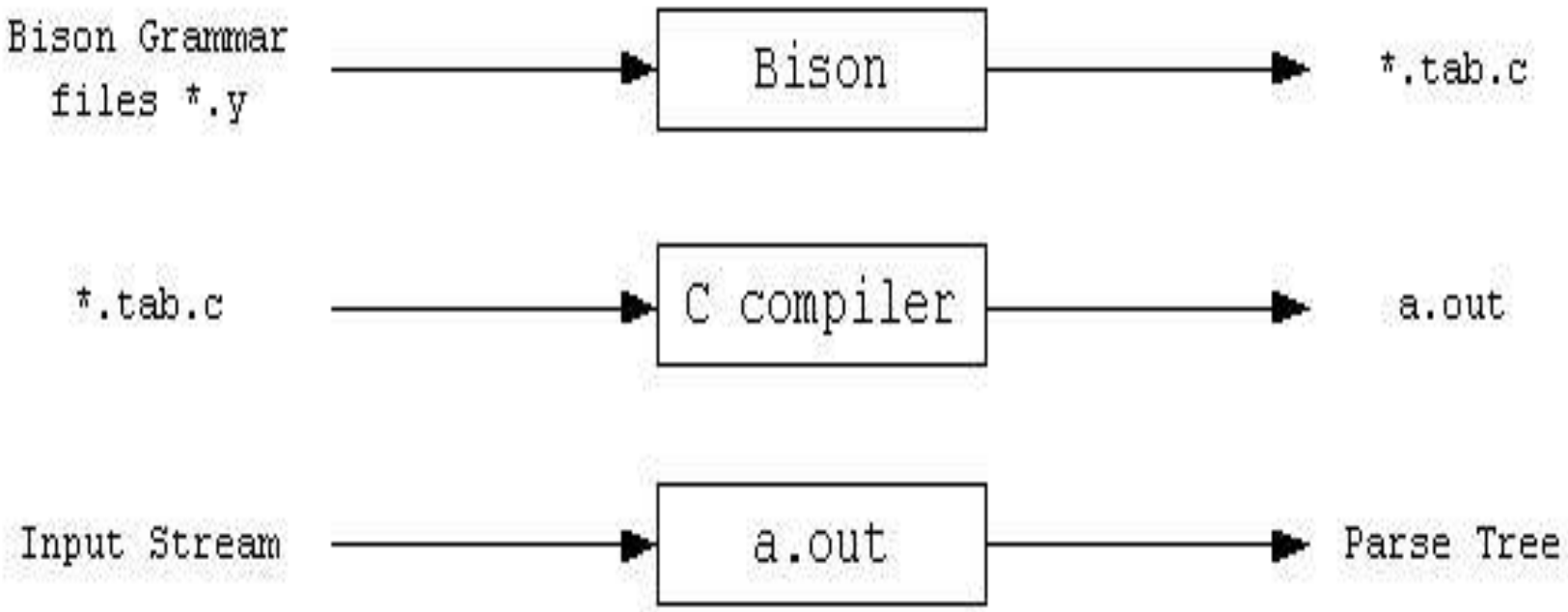
Simple Tutorial for Bison

Kshitij Agrawal
ka1042@nyu.edu

Tutorial to use Bison

- What is Bison?
- *Bison* is a general-purpose *parser generator* that converts a grammar description (Bison Grammar Files) for an LALR(1) context-free grammar into a C program to parse that grammar.
- The Bison parser is a bottom-up parser. It tries, by shifts and reductions, to reduce the entire input down to a single grouping whose symbol is the grammar's start-symbol.

Functionality



Steps to use Bison

- 1.Compile the example.y file using bison.
  bison –d example.y

- 2.This command generates two files:
  - example.tab.h example.tab.c.

- 3. On the other hand we also compile the lex file using flex.

Steps to use Bison(2)

- 4. Compile example.tab.c using
  g++ lex.yy.c example.tab.c -ll -o SA
- 5. This gives an executable SA.out
- 6. Test the executable using
  SA.out < “syntax.pas”
- 7. SA.out will call lex.yy.c for the tokens and make a tree.

Basic Layout

- Your *.y file should look like this:

%{ *C declarations* %}
declarations
%%
translation rules
%%
supporting C routines

Example

```
%{
#include <stdio.h>
#include <stdlib.h>
%}

%token <dvalue> DIGIT
%%
term : term '*' DIGIT {$$ = $1 * $3;}
        | term
%%

yylex() {
   int c;
   //Some computation
   return c;
}
```

Additional Reading

- Read Chp 4 Syntax Analysis 4.9