

System Programming and Operating Systems Lab

ASSIGNMENT 4

Name: Shrirang Mhalgi

Roll No: 322008

Batch: B 1

1 Date of Completion:

01/02/2019

2 Aim:

Write a program using YACC specifications to implement syntax analysis phase of compiler to recognize simple and compound sentences given in input file.

3 Objectives:

To implement basic compiler to recognize sentences.

4 Theory:

Yacc takes a concise description of a grammar and produces a C routine that can parse that grammar, a parser. The yacc parser automatically detects whenever a sequence of input tokens matches one of the rules in the grammar and also detects a syntax error whenever its input doesn't match any of the rules. A yacc parser is generally not as fast as a parser you could write by hand, but the ease in writing and modifying the parser is invariably worth any speed loss. The amount of time a program spends in a parser is rarely enough to be an issue anyway.

There are mainly three kinds of sentences in English: simple, complex and compound.

Simple sentence : A simple sentence consists of just one clause. Examples are given below.

- The dog barks.
- The kettle boils.
- Birds live in nests.
- The boys are singing.

In its simplest form, a simple sentence consists of a subject and a verb.

We can add more meaning to the sentence by including qualifiers, objects, complements etc.

- Sitting on a branch, the monkey gibbered.
- The little girl was carrying a basket on her head.

Compound sentence : A compound sentence is made up of two or more independent clauses.

- The boys sang and the girls danced.

This compound sentence consists of two simple clauses connected by the coordinating conjunction and.

Another example is given below.

- Men may come and men may go, but I go on forever.

This compound sentence consists of three independent clauses.

We make compound sentences by joining independent clauses with the help of coordinating conjunctions.

More examples of compound sentences are given below.

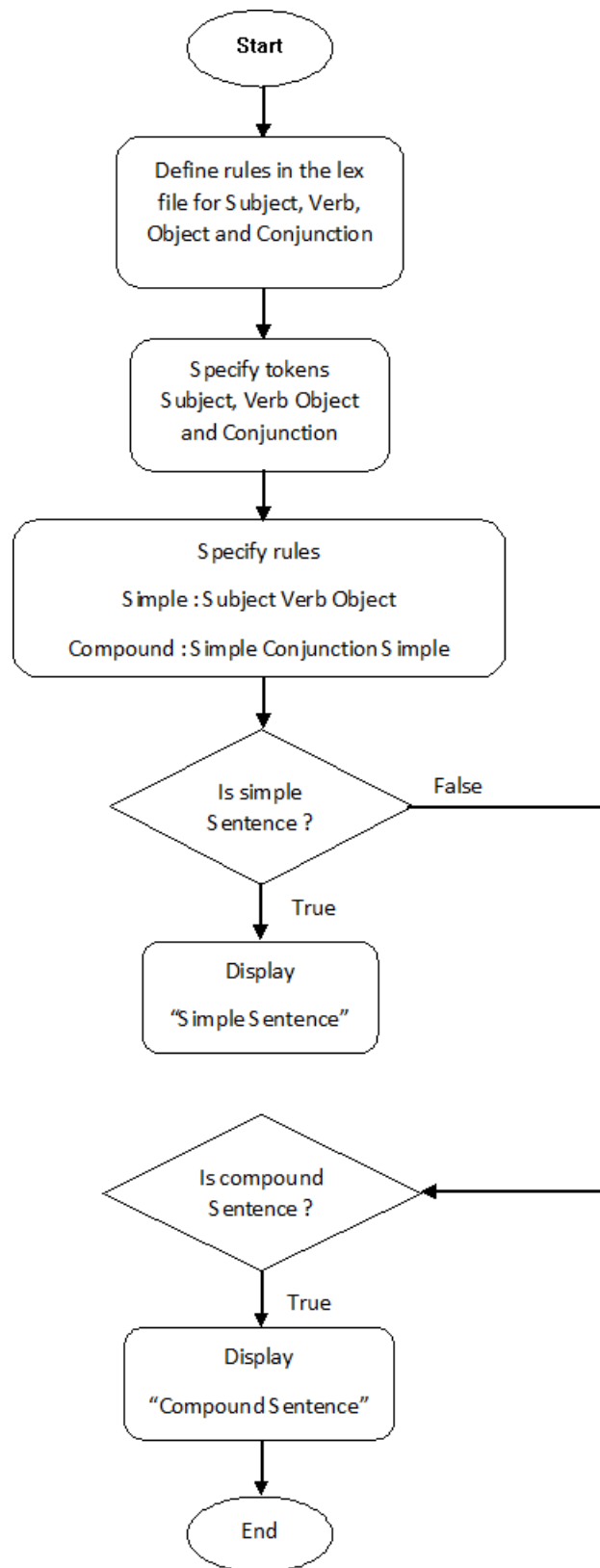
- You may watch TV or you can go out to play.
- You should either sit quietly or go out.
- The old man could neither see nor hear.
- He took the test several times but he couldn't pass.

This program tells whether the input given is compound sentence or simple one. It's easy, just check the scanned symbol with some conjunctives and if they are found, then it's compound sentence. Else a simple one. We use a flag variable in this code which is initially 0 indicating a simple sentence. If any conjunctives are matched, set the flag to 1. At the end, if flag=1 then compound sentence, else simple sentence.

5 Algorithm:

1. Start
2. Create a lex file.
3. Define the rules in the lex file your simple and compound sentences
4. Create a yacc file with same name as lex file.
5. Define tokens like Subject Verb Object and Conjunction.
6. Start with start:Simple/Compound as input can be simple or compound sentence.
7. Define label Simple and define its formation as Simple:Subject Verb Object.
8. If given sentence is as simple sentence the print result as " Simple Sentence" .
9. Define label Compound and define its formation as Compound:Simple Conjunction Simple.
10. If given sentence is as simple sentence the print result as " Compound Sentence" .
11. In main() call function yyparse()and yywrap()
12. End

6 Flowchart:



7 Code:

simcon.l :

```
%{
#include <stdio.h>
#include "y.tab.h"
}%
%%
I /
We /
He /
She /
You return Subject;

play /
sing /
dance return Verb ;

song /
soccer return Object ;

is /
and /
or /
not /
but return Conjunction ;
%%
```

simcon.y :

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

%token Subject Verb Object Conjunction

%%
start: Simple / Compound
Simple: Subject Verb Object {printf(" It is simple sentence! ");}
Compound: Simple Conjunction Simple {printf(" It is a Compound Sentence! ");}
;
%%

yyerror ()
{
    printf(" Error ");
}

int yywrap ()
```

```

{
    return 0;
}
main ()
{
    yyparse ( ) ;
}

```

8 Output:

```

sarthak@sarthak:~/Documents/SP05$ flex -l simcon.l
sarthak@sarthak:~/Documents/SP05$ yacc -vd simcom.y
sarthak@sarthak:~/Documents/SP05$ gcc lex.yy.c y.tab.c -lm -ll
sarthak@sarthak:~/Documents/SP05$ ./a.out
He sing song and She play soccer
    It is a Compound Sentence!
^Z
[8]+  Stopped                  ./a.out
sarthak@sarthak:~/Documents/SP05$ flex -l simcon.l
sarthak@sarthak:~/Documents/SP05$ yacc -vd simcom.y
sarthak@sarthak:~/Documents/SP05$ gcc lex.yy.c y.tab.c -lm -ll
sarthak@sarthak:~/Documents/SP05$ ./a.out
She sing song
    It is simple sentence!

```

9 Conclusion:

In this assignment we understood in detail how to implement a compiler to differentiate simple and compound sentences.