

## Assignment No:04

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

### stype1.l

```
%{  
/*  
 * We now build a lexical analyzer to be used by a higher-level parser.  
 */  
#include "y.tab.h" /* token codes from the parser */  
%}  
  
verb  
    is|am|are|was|were|be|being|been|do|does|did|will|would|should|can|could|have|has|had|go|play  
  
adverb    very|simply|gently|calmly|quietly  
  
preposition to|from|above|behind|below|between  
  
conjunction if|then|and|but|or  
  
adjective  their|my|your|his|her|its  
  
pronoun    I|you|he|she|they  
  
noun       [a-zA-Z]+  
  
%%  
  
[\\t ]+ ;  
  
{verb}      return VERB;  
  
{adverb}     return ADVERB;  
  
{preposition} return PREPOSITION;  
  
{conjunction} return CONJUNCTION;  
  
{adjective}  return ADJECTIVE;
```

```
{pronoun}    return PRONOUN;
```

```
{noun}       return NOUN;
```

```
"\n"         return NL;
```

```
%%
```

```
stype1.y
```

```
%{
```

```
#include<stdlib.h>
```

```
#include<stdio.h>
```

```
int yylex();
```

```
int yyerror();
```

```
%}
```

```
%token NOUN PRONOUN VERB ADVERB ADJECTIVE PREPOSITION CONJUNCTION  
NL
```

```
%%
```

```
sentence: simple_sentence NL {printf("Parsed a simple sentence.\n"); exit(0);}
```

```
    | compound_sentence NL {printf("Parsed a compound sentence.\n"); exit(0);}
```

```
;
```

```
simple_sentence: subject verb object
```

```
    | subject verb object prep_phrase
```

```
;
```

compound\_sentence: simple\_sentence CONJUNCTION simple\_sentence

| compound\_sentence CONJUNCTION simple\_sentence

;

subject: NOUN

| PRONOUN

| ADJECTIVE subject

;

verb: VERB

| ADVERB VERB

| verb VERB

;

object: NOUN

| ADJECTIVE object

;

prep\_phrase: PREPOSITION NOUN

;

%%

```
int main()
```

```
{
```

```
    yyparse();
```

```
}
```

```
int yyerror()
```

```
{
```

```
    printf("invalid");
```

```
}
```

```
int yywrap()
```

```
{
```

```
    return 1;
```

```
}
```

**Output :lex.stype1.l**

**yacc -d stype1.y**

**cc lex.yy.c y.tab.c**

**./a.out**

Explanation :

## Definitions Section

- This section includes the token definitions generated by Yacc (`y.tab.h`).

## Patterns Section

- Regular expressions for verbs, adverbs, prepositions, conjunctions, adjectives, pronouns, and nouns.

## Rules Section

```
{pronoun} return PRONOUN; {noun} return NOUN; "\n" return NL;
```

- Specifies what to return when each pattern is matched. For example, if a `verb` is found, return `VERB`.

## Token Definitions

```
c
```

Copy code

- Defines tokens that will be used in grammar rules.

## Grammar Rules Section

```
c
```

Copy code

```
printf "Parsed a simple sentence.\n" exit 0
printf "Parsed a compound sentence.\n" exit 0
```

- Defines the grammar rules for recognizing sentences:

- `sentence` can be a `simple_sentence` or `compound_sentence` followed by a newline (`NL`).
- `simple_sentence` consists of a `subject`, `verb`, and `object`, optionally followed by a `prep_phrase`.
- `compound_sentence` consists of `simple_sentence` connected by a `CONJUNCTION`.
- Definitions for `subject`, `verb`, `object`, and `prep_phrase`.

## Additional C Code Section

```
c
```

Copy code

```
int main int yyerror printf "invalid" int yywrap return 1
```

- `main()`: Calls `yyparse()` to start parsing.
- `yyerror()`: Prints "invalid" if a syntax error is encountered.
- `yywrap()`: Called at the end of input, returning 1 to indicate no more input.

## Summary

- **Lex file (stype1 . l):** Defines patterns for different parts of speech and generates tokens for the Yacc parser.
- **Yacc file (stype1 . y):** Uses these tokens to parse sentences according to defined grammar rules, recognizing simple and compound sentences and printing a message upon successful parsing.

When run, this program reads input, tokenizes it using the Lex scanner, and parses it using the Yacc parser to validate the structure of sentences based on the defined grammar.

also define what is verbs, adverbs, prepositions, conjunctions, adjectives, pronouns, and nouns.

ChatGPT

Sure! Here's a brief definition of each part of speech:

### Verbs

**Definition:** Verbs are words that describe actions, states, or occurrences. They are essential components of predicates in sentences.

**Examples:** is, am, are, was, were, be, being, been, do, does, did, will, would, should, can, could, have, has, had, go, play

### Adverbs

**Definition:** Adverbs modify verbs, adjectives, other adverbs, or whole sentences. They often describe how, when, where, or to what extent something happens.

**Examples:** very, simply, gently, calmly, quietly

### Prepositions

**Definition:** Prepositions are words that show relationships between nouns (or pronouns) and other words in a sentence. They often indicate direction, location, time, or method.

**Examples:** to, from, above, behind, below, between

### Conjunctions

**Definition:** Conjunctions are words that connect words, phrases, clauses, or sentences. They help to create more complex and cohesive structures in language.

**Examples:** if, then, and, but, or

### Adjectives

**Definition:** Adjectives are words that describe or modify nouns and pronouns, giving more information about their qualities or quantities.

**Examples:** their, my, your, his, her, its

## Pronouns

**Definition:** Pronouns are words that take the place of nouns or noun phrases. They are used to avoid repetition and simplify sentences.

**Examples:** I, you, he, she, they

## Nouns

**Definition:** Nouns are words that name people, places, things, or ideas. They serve as the subjects or objects in sentences.

**Examples:** In the provided lexical analyzer, any sequence of letters (both uppercase and lowercase) is treated as a noun. This can include words like "dog", "city", "happiness", etc.

## Application in the Program

In the provided Lex and Yacc files, these definitions are used to create patterns and rules for recognizing and parsing different parts of speech in sentences. Each pattern in the Lex file corresponds to a specific part of speech, and the Yacc file uses these parts to define grammar rules for valid sentence