CS421 Project

Japanese to English Translator

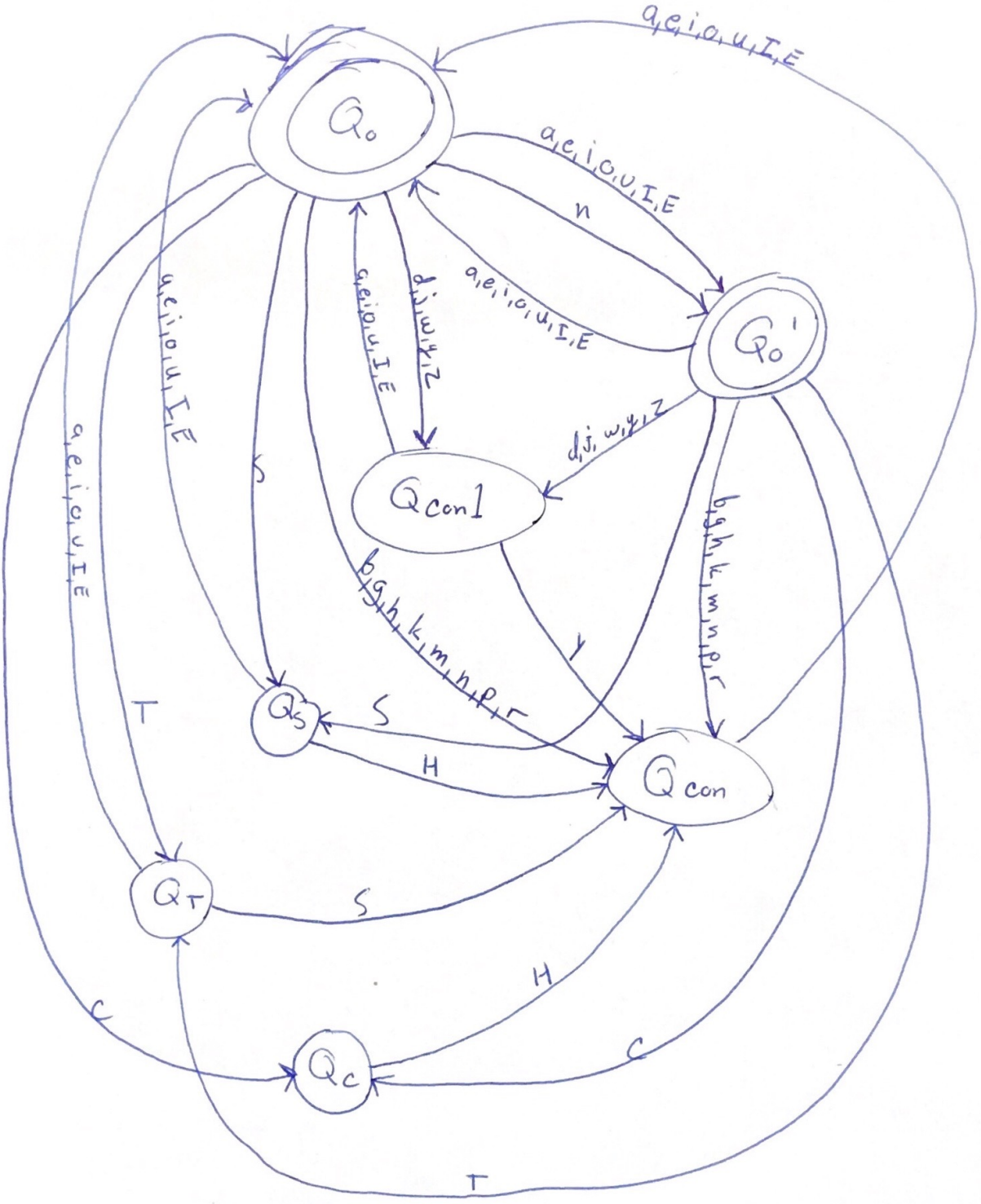
Group 5 Report

Created By:

Jack Sumners
Nick Garza
Eric Fink

State of the Program

- Translator is working perfectly for all test cases, and matches provided test outputs.
- All required parts have been included.
- Based on the example outputs for the translator and tracing messages for our parser, we have no errors or bugs to report!
- Extra credit features have **not** been implemented.


Q0
Q0'
Qcon1
Qcon
Q5
Q3
Q2
Qc
a,e,i,o,u,I,E
n
d,j,w,y,z
b,g,h,k,m,n,p,r
s
y
S
H
T
C

```
#include<iostream>
#include<fstream>
#include<string>
using namespace std;

//Q0 = 0
//Qcon1 = 1
//Q0' = 2
//Qcon = 3
//QS = 4
//QT = 5
//QC = 6

//=================================================
// File scanner.cpp written by: Group Number: 5
//=================================================

// --------- DFAs ---------------------------------

// ** MYTOKEN DFA to be replaced by the WORD DFA
// ** Done by: Jack Sumners
// ** RE:
bool mytoken(string s)
{
  //cout<<"Word: "<< s ;
  //cout<<"\n";

  int state =0;
  int charpos = 0;

  while (s[charpos] != '\0')
    {
     // cout<<" [  " << s[charpos] <<"  ] Start State: "<< state;


//** q0 STATE **
     if (state == 0 && (s[charpos] == 'b' || s[charpos] == 'g' || s[charpos] == 'h' || s[charpos] == 'k' || s[charpos] == 'm' || s[charpos] == 'n' || s[charpos] == 'p'
|| s[charpos] == 'r') )
       {state = 1;}
     else if(state == 0 && (s[charpos] == 'a' || s[charpos] == 'i' || s[charpos] == 'u' || s[charpos] == 'e' || s[charpos] == 'o' || s[charpos] == 'I' || s[charpos] ==
'E'))
       { state = 2;}
     else if(state == 0 && (s[charpos] == 'd' || s[charpos] == 'j' || s[charpos] == 'w' || s[charpos] == 'y' || s[charpos] == 'z'))
       { state = 3;}
     else if(state == 0 && s[charpos] == 's')
       { state = 4;}
        else if(state == 0 && s[charpos] == 't')
             { state = 5;}
        else if(state == 0 && s[charpos] == 'c')
          { state = 6;}
        else
          { //** Qcon1 STATE **
             if (state == 1 && (s[charpos] == 'a' || s[charpos] == 'i' || s[charpos] == 'u' || s[charpos] == 'e' || s[charpos] == 'o' || s[charpos] == 'I' || s[charpos]
== 'E'))
               {state = 2;}
               else if(state == 1 && s[charpos] == 'y')
               { state = 3;}
               else
                 { //** Q0' STATE **
           if(state == 2 && s[charpos] == 'n')
                        { state = 0; }
               else if (state == 2 && (s[charpos] == 'b' || s[charpos] == 'g' || s[charpos] == 'h' || s[charpos] == 'k' || s[charpos] == 'm' || s[charpos] == 'n' ||
s[charpos] == 'p' || s[charpos] == 'r') )
                        {state = 3;}
                  else if(state == 2 && (s[charpos] == 'a' || s[charpos] == 'i' || s[charpos] == 'u' || s[charpos] == 'e' || s[charpos] == 'o' || s[charpos] == 'I' ||
s[charpos] == 'E'))
                        { state = 2;}
```

```
                else if(state == 2 && (s[charpos] == 'd' || s[charpos] == 'j' || s[charpos] == 'w' || s[charpos] == 'y' || s[charpos] == 'z'))
                        { state = 3;}
                else if(state == 2 && s[charpos] == 's')
                        { state = 4;}
                   else if(state == 2 && s[charpos] == 't')
                        { state = 5;}
                   else if(state == 2 && s[charpos] == 'c')
                     { state = 6;}
                   else
                        { //** Qcon STATE **
                         if(state == 3 && (s[charpos] == 'a' || s[charpos] == 'i' || s[charpos] == 'u' || s[charpos] == 'e' || s[charpos] == 'o' || s[charpos] == 'I' ||
s[charpos] == 'E'))
                           { state = 2;}
                         else
                           { //** QS STATE **
                            if(state == 4 && (s[charpos] == 'a' || s[charpos] == 'i' || s[charpos] == 'u' || s[charpos] == 'e' || s[charpos] == 'o' || s[charpos] == 'I'
|| s[charpos] == 'E'))
                               { state = 2;}
                            else if(state == 4 && s[charpos] == 'h')
                               { state = 3;}
                              else
                               { //** QT STATE **
                                if(state == 5 && (s[charpos] == 'a' || s[charpos] == 'i' || s[charpos] == 'u' || s[charpos] == 'e' || s[charpos] == 'o' || s[charpos] ==
'I' || s[charpos] == 'E'))
                                  { state = 2;}
                                else if(state == 5 && s[charpos] == 's')
                                  { state = 3;}
                                else
                                    { //** QC STATE **
                                        if(state == 6 && s[charpos] == 'h')
                                      { state = 3;}
                                       else
                                      { //cout<<"no token found!"<<endl;
                                         return false;}
                                    }
                               }
                           }
                      }
                }
          }

            //cout<< " || end state: " <<state<<endl;

       charpos++;

    }//end of while

       // where did I end up????
  if (state == 0 || state == 2)
  {
    // cout <<"We did it"<<endl;
        return true;
  }
  return false;
}


// ** Add the PERIOD DFA here

bool periodtoken(string s)
{
       int state = 0;
       int charpos = 0;

       //checks each character of the word to test if is a expression of a single period "."
       while (s[charpos] != '\0')
       {
             if (state == 0 && s[charpos] == '.')
```

```
                state = 1;
        else
                return(false);

        ++charpos;
    }

    if (state == 1) return(true);
    else return(false);
}

// ** Done by: Nicholas Garza-Elsperger

// -----  Tables ---------------------------------------
// ** Update the tokentype to be WORD1, WORD2, PERIOD, ERROR, etc.
// Feel free to add a tokentype for the end-of-file marker.
enum tokentype { WORD1, WORD2, PERIOD, ERROR, VERB, VERBNEG, VERBPAST, VERBPASTNEG, IS, WAS, OBJECT, SUBJECT,
DESTINATION, PRONOUN, CONNECTOR };
//word1, word2, in above enum
string tokenName[30] = { "WORD1", "WORD2", "PERIOD", "ERROR", "VERB", "VERBNEG","VERBPAST","VERBPASTNEG","IS","WAS",
"OBJECT", "SUBJECT","DESTINATION", "PRONOUN", "CONNECTOR" };

// ** Need the reservedwords table to be set up here.
// ** Do not require any file input for this.
// ** a.out should work without any additional files.
struct reserved
{
  const char* string;
  tokentype tokenT;
}

//reserved table including string and tokentypes

reserved[] = {
  { "masu",VERB},
  {"masen",VERBNEG},
  {"mashita",VERBPAST},
  {"masendeshita",VERBPASTNEG},
  {"desu",IS},
  {"deshita",WAS},
  {"o", OBJECT},
  {"wa", SUBJECT},
  {"ni", DESTINATION},
  {"watashi",PRONOUN},
  {"anata",PRONOUN},
  {"kare",PRONOUN},
  {"kanojo", PRONOUN},
  {"sore", PRONOUN},
  {"mata",CONNECTOR},
  {"soshite", CONNECTOR},
  {"shikashi", CONNECTOR},
  { "dakara", CONNECTOR}
};

// ------------ Scaner and Driver -----------------------

ifstream fin;  // global stream for reading from the input file

// Scanner processes only one word each time it is called
// Gives back the token type and the word itself
// ** Done by: Eric Fink
int scanner(tokentype& a, string& w)
{
    // ** Grab the next word from the file via fin

    fin >> w;

  /*
```

```
 2. Call the token functions one after another (if-then-else)
    And generate a lexical error message if both DFAs failed.
    Let the token_type be ERROR in that case.
 3. Make sure WORDs are checked against the reservedwords list
    If not reserved, token_type is WORD1 or WORD2.

 4. Return the token type & string  (pass by reference)
 */



      int rowCount = sizeof reserved/ sizeof reserved[0];
      //cout<<"Row Count"<< rowCount <<endl;

      cout<<"\n";

      if(mytoken(w)) //checking step 2, part 1
        {
         for (int i = 0; i < rowCount; i++)
             {
              if (w == reserved[i].string) //checking step 3, part 1
                   {
                        //cout << "Word is reserved.\n";
                        //cout << "token_type: " << reserved[i][1] << endl;
                        a = reserved[i].tokenT;
                        return 0;
                   }
             }

         char lastLetter = ' ';
          lastLetter = w[(w.length()-1)];

          //checking step 3, part 2
         if (lastLetter == 'I' || lastLetter == 'E')
           {
            //cout << "token_type: WORD2\n";
            a = WORD2;
           }
         else
           {
            //cout << "token_type: WORD1\n";
            a = WORD1;
           }




      }
      else if(periodtoken(w)) //checking step 2, part 2
        {
         //cout<<"token_type: PERIOD\n";
         a = PERIOD;
        }
      else if(w == "eofm")
        { //do nothing
        }
      else
        {
         cout << "LEXICAL ERROR: "<< w <<" is not a valid token \n";
         a = ERROR;
        }


      return 0;
}//the end of scanner
```

```
// The temporary test driver to just call the scanner repeatedly
// This will go away after this assignment
// DO NOT CHANGE THIS!!!!!!
// Done by:  Rika
int main()
{
  tokentype thetype;
  string theword;
  string filename;

  cout << "Enter the input file name: ";
  cin >> filename;

  fin.open(filename.c_str());

  // the loop continues until eofm is returned.
   while (true)
    {
      scanner(thetype, theword);  // call the scanner
      if (theword == "eofm") break;  // stop now

      cout << "Type is:" << tokenName[thetype] << endl;
      cout << "Word is:" << theword << endl;
    }

   cout << "End of file is encountered." << endl;
   fin.close();

}// end
```

```
Script started on Mon 10 Dec 2018 07:14:41 PM PST
]0;sumne007@empress:~/CS421_Scanner_G5-master [?1034h[sumne007@empress CS421_Scanner_G5-master]$
g++ scanner.cpp
]0;sumne007@empress:~/CS421_Scanner_G5-master [sumne007@empress CS421_Scanner_G5-master]$ ./a.out
Enter the input file name: scannertest1

Type is:PRONOUN
Word is:watashi

Type is:SUBJECT
Word is:wa

Type is:WORD1
Word is:rika

Type is:IS
Word is:desu

Type is:PERIOD
Word is:.

Type is:PRONOUN
Word is:watashi

Type is:SUBJECT
Word is:wa

Type is:WORD1
Word is:sensei

Type is:IS
Word is:desu

Type is:PERIOD
Word is:.

Type is:PRONOUN
Word is:watashi

Type is:SUBJECT
Word is:wa

Type is:WORD1
Word is:ryouri

Type is:OBJECT
Word is:o

Type is:WORD2
Word is:yarI

Type is:VERB
Word is:masu
```

Type is:PERIOD
Word is:.

Type is:PRONOUN
Word is:watashi

Type is:SUBJECT
Word is:wa

Type is:WORD1
Word is:gohan

Type is:OBJECT
Word is:o

Type is:WORD1
Word is:seito

Type is:DESTINATION
Word is:ni

Type is:WORD2
Word is:agE

Type is:VERBPAST
Word is:mashita

Type is:PERIOD
Word is:.

Type is:CONNECTOR
Word is:shikashi

Type is:WORD1
Word is:seito

Type is:SUBJECT
Word is:wa

Type is:WORD2
Word is:yorokobI

Type is:VERBPASTNEG
Word is:masendeshita

Type is:PERIOD
Word is:.

Type is:CONNECTOR
Word is:dakara

Type is:PRONOUN
Word is:watashi

Type is:SUBJECT
Word is:wa

Type is:WORD1
Word is:kanashii

Type is:WAS
Word is:deshita

Type is:PERIOD
Word is:.

Type is:CONNECTOR
Word is:soshite

Type is:PRONOUN
Word is:watashi

Type is:SUBJECT
Word is:wa

Type is:WORD1
Word is:toire

Type is:DESTINATION
Word is:ni

Type is:WORD2
Word is:ikI

Type is:VERBPAST
Word is:mashita

Type is:PERIOD
Word is:.

Type is:PRONOUN
Word is:watashi

Type is:SUBJECT
Word is:wa

Type is:WORD2
Word is:nakI

Type is:VERBPAST
Word is:mashita

Type is:PERIOD
Word is:.

End of file is encountered.
]0;sumne007@empress:~/CS421_Scanner_G5-master [sumne007@empress CS421_Scanner_G5-master]$ exit

exit

Script done on Mon 10 Dec 2018 07:15:14 PM PST

```
Script started on Mon 10 Dec 2018 07:15:29 PM PST
]0;sumne007@empress:~/CS421_Scanner_G5-master [?1034h[sumne007@empress CS421_Scanner_G5-master]$
g++ scanner.cpp
]0;sumne007@empress:~/CS421_Scanner_G5-master [sumne007@empress CS421_Scanner_G5-master]$ ./a.out
Enter the input file name: scannertest2

Type is:WORD1
Word is:daigaku

LEXICAL ERROR: college is not a valid token
Type is:ERROR
Word is:college

Type is:WORD1
Word is:kurasu

LEXICAL ERROR: class is not a valid token
Type is:ERROR
Word is:class

Type is:WORD1
Word is:hon

LEXICAL ERROR: book is not a valid token
Type is:ERROR
Word is:book

Type is:WORD1
Word is:tesuto

LEXICAL ERROR: test is not a valid token
Type is:ERROR
Word is:test

Type is:WORD1
Word is:ie

LEXICAL ERROR: home* is not a valid token
Type is:ERROR
Word is:home*

Type is:WORD1
Word is:isu

LEXICAL ERROR: chair is not a valid token
Type is:ERROR
Word is:chair

Type is:WORD1
Word is:seito

LEXICAL ERROR: student is not a valid token
Type is:ERROR
```

Word is:student

Type is:WORD1
Word is:sensei

LEXICAL ERROR: teacher is not a valid token
Type is:ERROR
Word is:teacher

Type is:WORD1
Word is:tomodachi

LEXICAL ERROR: friend is not a valid token
Type is:ERROR
Word is:friend

Type is:WORD1
Word is:jidoosha

LEXICAL ERROR: car is not a valid token
Type is:ERROR
Word is:car

Type is:WORD1
Word is:gyuunyuu

LEXICAL ERROR: milk is not a valid token
Type is:ERROR
Word is:milk

Type is:WORD1
Word is:sukiyaki

Type is:WORD1
Word is:tenpura

Type is:WORD1
Word is:sushi

Type is:WORD1
Word is:biiru

LEXICAL ERROR: beer is not a valid token
Type is:ERROR
Word is:beer

Type is:WORD1
Word is:sake

LEXICAL ERROR: tokyo is not a valid token
Type is:ERROR
Word is:tokyo

Type is:WORD1

Word is:kyuushuu

LEXICAL ERROR: Osaka is not a valid token
Type is:ERROR
Word is:Osaka

Type is:WORD1
Word is:choucho

LEXICAL ERROR: butterfly is not a valid token
Type is:ERROR
Word is:butterfly

Type is:WORD1
Word is:an

Type is:WORD1
Word is:idea

Type is:WORD1
Word is:yasashii

LEXICAL ERROR: easy is not a valid token
Type is:ERROR
Word is:easy

Type is:WORD1
Word is:muzukashii

LEXICAL ERROR: difficult is not a valid token
Type is:ERROR
Word is:difficult

Type is:WORD1
Word is:ureshii

LEXICAL ERROR: pleased is not a valid token
Type is:ERROR
Word is:pleased

Type is:WORD1
Word is:shiawase

LEXICAL ERROR: happy is not a valid token
Type is:ERROR
Word is:happy

Type is:WORD1
Word is:kanashii

LEXICAL ERROR: sad is not a valid token
Type is:ERROR
Word is:sad

Type is:WORD1
Word is:omoi

LEXICAL ERROR: heavy is not a valid token
Type is:ERROR
Word is:heavy

Type is:WORD1
Word is:oishii

LEXICAL ERROR: delicious is not a valid token
Type is:ERROR
Word is:delicious

Type is:WORD1
Word is:tennen

LEXICAL ERROR: natural is not a valid token
Type is:ERROR
Word is:natural

Type is:WORD2
Word is:nakI

LEXICAL ERROR: cry is not a valid token
Type is:ERROR
Word is:cry

Type is:WORD2
Word is:ikI

LEXICAL ERROR: go* is not a valid token
Type is:ERROR
Word is:go*

Type is:WORD2
Word is:tabE

LEXICAL ERROR: eat is not a valid token
Type is:ERROR
Word is:eat

Type is:WORD2
Word is:ukE

LEXICAL ERROR: take* is not a valid token
Type is:ERROR
Word is:take*

Type is:WORD2
Word is:kakI

LEXICAL ERROR: write is not a valid token
Type is:ERROR

Word is:write

Type is:WORD2
Word is:yomI

LEXICAL ERROR: read is not a valid token
Type is:ERROR
Word is:read

Type is:WORD2
Word is:nomI

LEXICAL ERROR: drink is not a valid token
Type is:ERROR
Word is:drink

Type is:WORD2
Word is:agE

LEXICAL ERROR: give is not a valid token
Type is:ERROR
Word is:give

Type is:WORD2
Word is:moraI

LEXICAL ERROR: receive is not a valid token
Type is:ERROR
Word is:receive

Type is:WORD2
Word is:butsI

LEXICAL ERROR: hit is not a valid token
Type is:ERROR
Word is:hit

Type is:WORD2
Word is:kerI

LEXICAL ERROR: kick is not a valid token
Type is:ERROR
Word is:kick

Type is:WORD2
Word is:shaberI

LEXICAL ERROR: talk is not a valid token
Type is:ERROR
Word is:talk

End of file is encountered.
]0;sumne007@empress:~/CS421_Scanner_G5-master [sumne007@empress CS421_Scanner_G5-master]$ exit
exit

Script done on Mon 10 Dec 2018 07:15:52 PM PST

CS421 Yoshii - Hint on Left Factoring
Email the left factored rules to me before Week12A.

Recursive Descent Parsing requires each rule to have a unique start.

Keep it one rule until it starts to be different

```
2 <s> ::=  [CONNECTOR] <noun> SUBJECT  <verb> <tense> PERIOD
3 <s> ::=  [CONNECTOR] <noun> SUBJECT  <noun> <be>    PERIOD
4 <s> ::=  [CONNECTOR] <noun> SUBJECT  <noun> DESTINATION  <verb> <tense>
PERIOD
5 <s> ::=  [CONNECTOR] <noun> SUBJECT  <noun> OBJECT <verb> <tense> PERIOD
6 <s> ::=  [CONNECTOR] <noun> SUBJECT  <noun> OBJECT <noun> DESTINATION
                                                    <verb> <tense> PERIOD
```

UPDATED GRAMMAR RULES FOR TRANSLATOR:

Using #getEword#, and #gen#

1 <story> :: = <s> { <s> }

2 <s> ::= [CONNECTOR #getEword# #gen("CONNECTOR")#] <noun> #getEword# SUBJECT #gen("ACTOR")# <after subject>

3 <after subject> ::= <verb> #getEword# #gen("ACTION")# <tense> #gen("TENSE")# PERIOD | <noun> #getEword# <after noun>

4 <after noun> ::= <be> #gen("DESCRIPTION")# #gen("TENSE")# PERIOD | DESTINATION #gen("TO")# <verb> #getEword# #gen("ACTION")# <tense> #gen("TENSE")# PERIOD| OBJECT #gen("OBJECT")# <after object>

5 <after object> ::= <verb> #getEword# #gen("ACTION")# <tense> gen("TENSE")# PERIOD | <noun> #getEword# DESTINATION #gen("TO")# <verb> #getEword# #gen("ACTION")# <tense> #gen("TENSE")# PERIOD

6 <noun> ::= WORD1 | PRONOUN

7 <verb> ::= WORD2

8 <be> ::= IS | WAS

9 <tense> ::= VERBPAST | VERBPASTNEG | VERB | VERBNEG

```
#include<iostream>
#include<fstream>
#include<string>
#include "scanner.cpp"
#include <stdlib.h>
#include <vector>
using namespace std;

// INSTRUCTION:  copy and edit your parser.cpp to create this file.
// cp ../ParserFiles/parser.cpp .
// Complete all ** parts.
// ----------------------------------------------------------

//===================================================
// File translator.cpp written by Group Number: 5
//===================================================

// ----- Changes to the parser.cpp ---------------------

// ** Declare dictionary that will hold the content of lexicon.txt
// Make sure it is easy and fast to look up the translation
// Do not change the format or content of lexicon.txt

// ** Additions to parser.cpp here:
//    getEword - using the current lexeme, look up the English word
//               in the Lexicon if it is there -- save the result
//               in saved_E_word
//    gen(line_type) - using the line type,
//                     sends a line of an IR to translated.txt
//                     (saved_E_word or saved_token is used)



// ** Be sure to put the name of the programmer above each function

// ** Be sure to put the corresponding grammar
//    rule with semantic routines
//    above each non-terminal function

//====================================================================



void s();
void afterSubject();
void afterNoun();
void afterObject();
void noun();
void verb();
void be();
void tense();

ofstream outFile;

string saved_lexeme;
bool  token_available = false;
token_type saved_token;

string savedEword;

//Table for translation
//Done By: Jack Sumners
vector<string> wordJ ;
vector <string> wordE ;
```

```
//Done By: Eric Fink
//Function getEword: takes the current saved lexeme and checks it aganst the translation table
void getEword()
{
  //outFile<< "GETTING EWORD USING:   "<< saved_lexeme <<endl;
  bool found = false;
  for(int a =0;a< wordJ.size(); a++)
    {
     if(wordJ[a]==saved_lexeme)
       {
        savedEword = wordE[a];
        found = true;
       }
    }
  if(found == false)
    {savedEword = saved_lexeme;}
  //outFile<<"new saved E word  "<< savedEword <<endl;
}


//Done By: Nick Garza
//Function gen: generates a file output based on our saved line and our token or savedEword
void gen(string line_type)
{
  if(line_type == "TENSE")
    {outFile<< line_type<< "  "<< tokenName[saved_token] <<endl;}
  else{
  outFile<< line_type<< "  "<< savedEword <<endl;
  }
}



// INSTRUCTION:  Complete all ** parts.
// You may use any method to connect this file to scanner.cpp
// that you had written.
// You can copy scanner.cpp here by cp ../ScannerFiles/scanner.cpp .
// ----------------------------------------------------

//==================================================
// File parser.cpp written by Group Number: 5
//==================================================

// ----- Utility and Globals -----------------------------------

// ** Need syntaxerror1 and syntaxerror2 functions (each takes 2 args)
// ** Be sure to put the name of the programmer above each function
// i.e. Done by:

//Done By: Nick Garza
void syntax_error1(token_type expected, string saved_lexeme)
{
  cout<<"SYNTAX ERROR: expected "<< tokenName[expected]  << " but found "<<  saved_lexeme <<endl;
  exit (1); //halting
}

//Done By: Nick Garza
void syntax_error2(string saved_lexeme, string parserFunct)
{
  cout<<"SYNTAX ERROR: unexpected "<< saved_lexeme << " found in "<< parserFunct <<endl;
  exit (1); //halting
}

// ** Need the updated match and next_token (with 2 global vars)
// ** Be sure to put the name of the programmer above each function
// i.e. Done by:
```

```
//Done By: Eric Fink
token_type next_token()
{

  if (!token_available)   // if there is no saved token yet
    {
      scanner(saved_token, saved_lexeme);  // call scanner to grab a new token
      cout<<"Scanner called using word: "<< saved_lexeme <<endl;
      token_available = true;                          // mark that fact that you have saved it
    }
  return saved_token;    // return the saved token
}



//Done By: Eric Fink
bool match(token_type expected)
{
  if (next_token() != expected)  // mismatch has occurred with the next token
    {
      syntax_error1(expected, saved_lexeme);// calls a syntax error function here to  generate a syntax error message here and do recovery

    }
  else  // match has occurred
    {
      cout<<"Matched "<< tokenName[expected] <<endl; //print the matched token type, when successful
      token_available = false;  // eat up the token
      return true;              // say there was a match
    }
}



// ----- RDP functions - one per non-term ----------

// ** Make each non-terminal into a function here
// ** Be sure to put the corresponding grammar rule above each function
// i.e. Grammar:
// ** Be sure to put the name of the programmer above each function
// i.e. Done by:


//Grammar:<story> ::= <s>{<s>}
//Done By: Jack Sumners
void story()
{
  cout<<"Processing <story>"<<endl;
  s();
  //while()  call more s()
  while (true && (saved_lexeme != "eofm"))
    {
      outFile<<endl;
      s();
    }
  cout<<"\nSuccessfully parsed story"<<endl;
}

//Grammar:<s> ::= [CONNECTOR #getEword# #gen("CONNECTOR")]<noun> #getEword# SUBJECT #gen("ACTOR")<after subject>
//Done By: Jack Sumners
void s()
{
  next_token();
  if(saved_lexeme != "eofm")
    {

      cout<<"Processing <s>"<<endl;
```

```
    if(next_token() == CONNECTOR)
      {
       match(CONNECTOR);
       getEword();
       gen("CONNECTOR");
      }

    noun();
    match(SUBJECT);
    gen("ACTOR");
    afterSubject();


   }

}

//Grammar:<noun> ::= WORD1|PRONOUN
//Done By: Jack Sumners
void noun()
{

  cout<<"Processing <noun>"<<endl;
  switch(next_token())  // look ahead at next token
    {
    case  WORD1:
      match(WORD1);
      getEword();
      break;
    case  PRONOUN:
      match(PRONOUN);
      getEword();
      break;
    default:
      syntax_error2(saved_lexeme, "noun");  // none of the alternatives found
    }
}

//Grammar: <after subject> ::= <verb> #getEword# #gen("ACTION")# <tense> #gen("TENSE")# PERIOD | <noun> #getEword# <after noun>
//Done By: Eric Fink
void afterSubject()
{
  cout<<"Processing <afterSubject>"<<endl;

  switch(next_token())  // look ahead at next token
    {
    case  WORD2:
      verb();
      tense();
      match(PERIOD);
      break;
    case  WORD1:
      noun();
      afterNoun();
      break;
    case PRONOUN:
      noun();
      afterNoun();
      break;
    default:
      syntax_error2(saved_lexeme, "afterSubject");  // none of the alternatives found
    }
}


//Grammar: <verb> ::= WORD2
//Done By: Eric Fink
```

```
void verb()
{
 cout<<"Processing <verb>"<<endl;

 switch(next_token())  // look ahead at next token
   {
   case  WORD2:
     match(WORD2);
     getEword();
     gen("ACTION");
     break;
   default:
     syntax_error2(saved_lexeme, "verb");  // none of the alternatives found
   }
}

//Grammar: <be> ::= IS | WAS
//Done By: Eric Fink
void be()
{

 cout<<"Processing <be>"<<endl;
 //outFile<< "Before Desc:   "<< savedEword <<endl;
 gen("DESCRIPTION");
 switch(next_token())  // look ahead at next token
   {
   case  IS:
     match(IS);
     //   gen("DESCRIPTION");
     gen("TENSE");
     break;
   case  WAS:
     match(WAS);
     // gen("DESCRIPTION");
     gen("TENSE");
     break;
   default:
     syntax_error2(saved_lexeme, "be");  // none of the alternatives found
   }
}



//Grammar:<after noun> ::= <be> #gen("DESCRIPTION")# #gen("TENSE")# PERIOD  | DESTINATION #gen("TO")#<verb> #getEword#
#gen("ACTION")# <tense> #gen("Tense")# PERIOD | OBJECT #gen("OBJECT")# <after object>
//Done By: Nick Garza
void afterNoun()
{
 cout<<"Processing <afterNoun>"<<endl;


 switch(next_token())  // look ahead at next token
   {
   case  IS:
     be();
     match(PERIOD);
     break;
   case  WAS:
     be();
     match(PERIOD);
     break;
   case DESTINATION:
     match(DESTINATION);
     gen("TO");
     verb();
     tense();
     match(PERIOD);
```

```
    break;
   case OBJECT:
    match(OBJECT);
    gen("OBJECT");
    afterObject();
    break;
   default:
    syntax_error2(saved_lexeme, "afterNoun");  // none of the alternatives found
   }

}

//Grammar:<after object> ::= <verb> #getEword# #gen("ACTION")#<tense> #gen("TENSE")# PERIOD | <noun> #getEword# DESTINATION
#gen("TO")# <verb> #getEword# #gen("ACTION")# <tense> #gen("TENSE")# PERIOD
//Done By: Nick Garza
void afterObject()
{
 cout<<"Processing <afterObject>"<<endl;

 switch(next_token())
   {
   case WORD2:
    verb();
    tense();
    match(PERIOD);
    break;
   case WORD1:
    noun();
    match(DESTINATION);
    gen("TO");
    verb();
    tense();
    match(PERIOD);
    break;
   case PRONOUN:
    noun();
    match(DESTINATION);
    gen("TO");
    verb();
    tense();
    match(PERIOD);
    break;
   default:
    syntax_error2(saved_lexeme, "afterObject");  // none of the alternatives found
   }
}


//Grammar: <tense> ::= VERBPAST | VERBPASTNEG | VERB | VERBNEG
//Done By: Nick Garza
void tense()
{
 cout<<"Processing <tense>"<<endl;

 switch(next_token())  // look ahead at next token
   {
   case  VERBPAST:
    match(VERBPAST);
    gen("TENSE");
    break;
   case  VERBPASTNEG:
    match(VERBPASTNEG);
    gen("TENSE");
    break;
   case  VERB:
    match(VERB);
    gen("TENSE");
    break;
```

```
    case  VERBNEG:
      match(VERBNEG);
      gen("TENSE");
      break;
    default:
      syntax_error2(saved_lexeme, "tense");  // none of the alternatives found
    }
}
```

```
// -------------------------------------------

// The final test driver to start the translator
// Done by  **
int main()
{
  //** opens the lexicon.txt file and reads it in
  ifstream input;
  string tJ;
  string tE;
  input.open("lexicon.txt");
  cout<<"opening file"<<endl;
  while(input)
    {

      input>> tJ;
      input>> tE;
      wordJ.push_back(tJ);
      wordE.push_back(tE);
    }



  input.close();
  //** closes lexicon.txt

  //** opens the output file translated.txt

  outFile.open("translated.txt");

  string filename;
  cout << "Enter the input file name: ";
  cin >> filename;
  fin.open(filename.c_str());

  //** calls the <story> to start parsing
  story();
  //** closes the input file
  fin.close();
  //** closes translated.txt
  outFile.close();
}// end
```

```
Script started on Mon 10 Dec 2018 03:10:29 PM PST
]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [?1034h[sumne007@empress TranslatorFiles]$ g++ scanner [K [K [K [K [K [K [Ktranslatr [Kor.cpp
]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ ./a.out
opening file
Enter the input file name: partCtest1
Processing <story>
Scanner called using word: watashi
Processing <s>
Processing <noun>
Matched PRONOUN
Scanner called using word: wa
Matched SUBJECT
Processing <afterSubject>
Scanner called using word: rika
Processing <noun>
Matched WORD1
Processing <afterNoun>
Scanner called using word: desu
Processing <be>
Matched IS
Scanner called using word: .
Matched PERIOD
Scanner called using word: watashi
Processing <s>
Processing <noun>
Matched PRONOUN
Scanner called using word: wa
Matched SUBJECT
Processing <afterSubject>
Scanner called using word: sensei
Processing <noun>
Matched WORD1
Processing <afterNoun>
Scanner called using word: desu
Processing <be>
Matched IS
Scanner called using word: .
Matched PERIOD
Scanner called using word: rika
Processing <s>
Processing <noun>
Matched WORD1
Scanner called using word: wa
Matched SUBJECT
Processing <afterSubject>
Scanner called using word: gohan
Processing <noun>
Matched WORD1
Processing <afterNoun>
Scanner called using word: o
Matched OBJECT
Processing <afterObject>
```

```
Scanner called using word: tabE
Processing <verb>
Matched WORD2
Processing <tense>
Scanner called using word: masu
Matched VERB
Scanner called using word: .
Matched PERIOD
Scanner called using word: watashi
Processing <s>
Processing <noun>
Matched PRONOUN
Scanner called using word: wa
Matched SUBJECT
Processing <afterSubject>
Scanner called using word: tesuto
Processing <noun>
Matched WORD1
Processing <afterNoun>
Scanner called using word: o
Matched OBJECT
Processing <afterObject>
Scanner called using word: seito
Processing <noun>
Matched WORD1
Scanner called using word: ni
Matched DESTINATION
Processing <verb>
Scanner called using word: agE
Matched WORD2
Processing <tense>
Scanner called using word: mashita
Matched VERBPAST
Scanner called using word: .
Matched PERIOD
Scanner called using word: shikashi
Processing <s>
Matched CONNECTOR
Processing <noun>
Scanner called using word: seito
Matched WORD1
Scanner called using word: wa
Matched SUBJECT
Processing <afterSubject>
Scanner called using word: yorokobI
Processing <verb>
Matched WORD2
Processing <tense>
Scanner called using word: masendeshita
Matched VERBPASTNEG
Scanner called using word: .
Matched PERIOD
Scanner called using word: dakara
Processing <s>
```

```
Matched CONNECTOR
Processing <noun>
Scanner called using word: watashi
Matched PRONOUN
Scanner called using word: wa
Matched SUBJECT
Processing <afterSubject>
Scanner called using word: kanashii
Processing <noun>
Matched WORD1
Processing <afterNoun>
Scanner called using word: deshita
Processing <be>
Matched WAS
Scanner called using word: .
Matched PERIOD
Scanner called using word: soshite
Processing <s>
Matched CONNECTOR
Processing <noun>
Scanner called using word: rika
Matched WORD1
Scanner called using word: wa
Matched SUBJECT
Processing <afterSubject>
Scanner called using word: toire
Processing <noun>
Matched WORD1
Processing <afterNoun>
Scanner called using word: ni
Matched DESTINATION
Processing <verb>
Scanner called using word: ikI
Matched WORD2
Processing <tense>
Scanner called using word: mashita
Matched VERBPAST
Scanner called using word: .
Matched PERIOD
Scanner called using word: rika
Processing <s>
Processing <noun>
Matched WORD1
Scanner called using word: wa
Matched SUBJECT
Processing <afterSubject>
Scanner called using word: nakI
Processing <verb>
Matched WORD2
Processing <tense>
Scanner called using word: mashita
Matched VERBPAST
Scanner called using word: .
Matched PERIOD
```

```
Scanner called using word: eofm

Successfully parsed story
 ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ emacs translated.txt
 [?1049h [?12;25h [?1h = [H [2J [58d [K [?1l > [?12l [?25h [?1049l [39;49m
 [?1049h [?12;25h [?1h = [H [2J [57d [?25l-UUU:----F1 [1m*scratch* [0m All L1 (Fundamental) -- ----------------------------------------------------
 [A [2d [?12l [?25h [?12;25h [58d [?25lLoading /usr/share/emacs/site-lisp/site-start.d/desktop-entry-mode-init.el (source)... [K [H
 [?12l [?25h [?12;25h [58;87H [?25ldone [H
 [?12l [?25h [?12;25h [58;49H [?25lsystemtap-init.el (source)... [K [H
 [?12l [?25h [?12;25h [58;78H [?25ldone [H
 [?12l [?25h [?12;25h [>0c [58d [?25lLoading /usr/share/emacs/site-lisp/site-start.d/systemtap-init.el (source)...done [K [H [7mFile Edit Options Buffers Tools Lisp-Interaction Help
 [0m [39;49m [27m
 [A
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
```

[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[30m [48;5;250m-UUU:----F1 [39;49m [1m [30m [48;5;250m*scratch*
[0m [39;49m [30m [48;5;250m All L1 (Lisp Interaction) ---------- [39;49m
[A [2d [?12l [?25h [?12;25h [58d [?25lFor information about GNU Emacs and the GNU system, type C-h C-a. [K [H
[?12l [?25h [?12;25h [58d [?25lFor information about GNU Emacs and the GNU system, type C-h C-a. [K [H [7mFile Edit Options Buffers Tools Help
[0m [39;49m [27m
[A
ACTOR I/me [K
DESCRIPTION rika [K
TENSE IS [K
[K
ACTOR I/me [K
DESCRIPTION teacher [K
TENSE IS [K
[K
ACTOR rika [K
OBJECT meal [K
ACTION eat [K
TENSE VERB [K
[K
ACTOR I/me [K
OBJECT test [K
TO student [K
ACTION give [K
TENSE VERBPAST [K
[K
CONNECTOR However [K
ACTOR student [K
ACTION enjoy [K
TENSE VERBPASTNEG [K
[K
CONNECTOR Therefore [K
ACTOR I/me [K
DESCRIPTION sad [K
TENSE WAS [K

```
  [K
CONNECTOR Then [K
ACTOR rika [K
TO restroom [K
ACTION go [K
TENSE VERBPAST [K
  [K
ACTOR rika [K
ACTION cry [K
TENSE VERBPAST [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [30m [48;5;250m-UU-:----F1
  [39;49m [1m [30m [48;5;250mtranslated.txt [0m [39;49m [30m [48;5;250m All L1 (Text) ----------
------------------------------------------ [39;49m
  [A [2d [?12l [?25h [?12;25h [58d [K [2d [58d [?25l(No files need saving) [H
  [?12l [?25h [?12;25h [58d [K [?1l > [?12l [?25h [?1049l [39;49m
  ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ exit
exit

Script done on Mon 10 Dec 2018 03:12:17 PM PST
```

```
Script started on Mon 10 Dec 2018 03:14:38 PM PST
]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [?1034h[sumne007@empress TranslatorFiles]$
g_ [K++translated [K [K [K [K [K [K [K [K [K [K
translated [K [K [Ktor.cpp
]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ ./a.out
opening file
Enter the input file name: partCtest2
Processing <story>
Scanner called using word: soshite
Processing <s>
Matched CONNECTOR
Processing <noun>
Scanner called using word: watashi
Matched PRONOUN
Scanner called using word: wa
Matched SUBJECT
Processing <afterSubject>
Scanner called using word: rika
Processing <noun>
Matched WORD1
Processing <afterNoun>
Scanner called using word: desu
Processing <be>
Matched IS
Scanner called using word: ne
SYNTAX ERROR: expected PERIOD but found ne
]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ emacs
translated.txt
[?1049h [?12;25h [?1h = [H [2J [58d [K [?1l > [?12l [?25h [?1049l [39;49m
[?1049h [?12;25h [?1h = [H [2J [57d [?25l-UUU:----F1 [1m*scratch* [0m All L1 (Fundamental) --
----------------------------------------------------
[A [2d [?12l [?25h [?12;25h [58d [?25lLoading /usr/share/emacs/site-lisp/site-start.d/desktop-entry-mode-
init.el (source)... [K [H
[?12l [?25h [?12;25h [58;87H [?25ldone [H
[?12l [?25h [?12;25h [58;49H [?25lsystemtap-init.el (source)... [K [H
[?12l [?25h [?12;25h [58;78H [?25ldone [H
[?12l [?25h [?12;25h [>0c [58d [?25lLoading /usr/share/emacs/site-lisp/site-start.d/systemtap-init.el
(source)...done [K [H [7mFile Edit Options Buffers Tools Lisp-Interaction Help
[0m [39;49m [27m
[A
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
```

```
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [30m [48;5;250m-UUU:----F1  [39;49m [1m [30m [48;5;250m*scratch*
 [0m [39;49m [30m [48;5;250m All L1  (Lisp Interaction) ------------------------------------------------ [39;49m
 [A [2d [?12l [?25h [?12;25h [58d [?25lFor information about GNU Emacs and the GNU system, type C-h C-
a. [K [H
 [?12l [?25h [?12;25h [58d [?25lFor information about GNU Emacs and the GNU system, type C-h C-
a. [K [H [7mFile Edit Options Buffers Tools Help
 [0m [39;49m [27m
 [A
CONNECTOR Then [K
ACTOR I/me [K
DESCRIPTION rika [K
TENSE IS [K
```

```
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[30m [48;5;250m-UU-:----F1
[39;49m [1m [30m [48;5;250mtranslated.txt [0m [39;49m [30m [48;5;250m All L1 (Text) ------------------
------------------------------------------ [39;49m
```

[A [2d [?12l [?25h [?12;25h [58d [K [2d [58d [?25l(No files need saving) [H [?12l [?25h [?12;25h [58d [K [?1l > [?12l [?25h [?1049l [39;49m]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ exit
exit

Script done on Mon 10 Dec 2018 03:15:12 PM PST

```
Script started on Mon 10 Dec 2018 03:15:35 PM PST
 ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [?1034h[sumne007@empress TranslatorFiles]$ g++
trn [Kanslator.cpp
 ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ ./a.out
opening file
Enter the input file name: partCtest3
Processing <story>
Scanner called using word: dakara
Processing <s>
Matched CONNECTOR
Processing <noun>
Scanner called using word: watashi
Matched PRONOUN
Scanner called using word: de
SYNTAX ERROR: expected SUBJECT but found de
 ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ emacs
translated.txt
 [?1049h [?12;25h [?1h = [H [2J [58d [K [?1l > [?12l [?25h [?1049l [39;49m
 [?1049h [?12;25h [?1h = [H [2J [57d [?25l-UUU:----F1 [1m*scratch* [0m All L1 (Fundamental) --
----------------------------------------------------
 [A [2d [?12l [?25h [?12;25h [58d [?25lLoading /usr/share/emacs/site-lisp/site-start.d/desktop-entry-mode-
init.el (source)... [K [H
 [?12l [?25h [?12;25h [58;87H [?25ldone [H
 [?12l [?25h [?12;25h [58;49H [?25lsystemtap-init.el (source)... [K [H
 [?12l [?25h [?12;25h [58;78H [?25ldone [H
 [?12l [?25h [?12;25h [>0c [58d [?25lLoading /usr/share/emacs/site-lisp/site-start.d/systemtap-init.el
(source)...done [K [H [7mFile Edit Options Buffers Tools Lisp-Interaction Help
 [0m [39;49m [27m
 [A
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
```

[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[30m [48;5;250m-UUU:----F1 [39;49m [1m [30m [48;5;250m*scratch*
[0m [39;49m [30m [48;5;250m All L1 (Lisp Interaction) -- [39;49m
[A [2d [?12l [?25h [?12;25h [58d [?25lFor information about GNU Emacs and the GNU system, type C-h C-a. [K [H
[?12l [?25h [?12;25h [58d [?25lFor information about GNU Emacs and the GNU system, type C-h C-a. [K [H [7mFile Edit Options Buffers Tools Help
[0m [39;49m [27m
[A
CONNECTOR Therefore [K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K

[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[30m [48;5;250m-UU-:----F1
[39;49m [1m [30m [48;5;250mtranslated.txt [0m [39;49m [30m [48;5;250m All L1 (Text) -- [39;49m
[A [2d [?12l [?25h [?12;25h [58d [K [2d [58d [?25l(No files need saving) [H
[?12l [?25h [?12;25h [58d [K [?1l > [?12l [?25h [?1049l [39;49m
]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$. [Kexit
exit

Script done on Mon 10 Dec 2018 03:17:04 PM PST

```
Script started on Mon 10 Dec 2018 03:17:33 PM PST
 ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [?1034h[sumne007@empress TranslatorFiles]$ g==
trans [K [K [K [K [K [K [K [K++ translator.cpp
 ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ ./a.out
opening file
Enter the input file name: partCtest4
Processing <story>
Scanner called using word: watashi
Processing <s>
Processing <noun>
Matched PRONOUN
Scanner called using word: wa
Matched SUBJECT
Processing <afterSubject>
Scanner called using word: rika
Processing <noun>
Matched WORD1
Processing <afterNoun>
Scanner called using word: mashita
SYNTAX ERROR: unexpected mashita found in afterNoun
 ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ emacs
translated.txt
 [?1049h [?12;25h [?1h = [H [2J [58d [K [?1l > [?12l [?25h [?1049l [39;49m
 [?1049h [?12;25h [?1h = [H [2J [57d [?25l-UUU:----F1 [1m*scratch* [0m All L1 (Fundamental) --
----------------------------------------------------
 [A [2d [?12l [?25h [?12;25h [58d [?25lLoading /usr/share/emacs/site-lisp/site-start.d/desktop-entry-mode-
init.el (source)... [K [H
 [?12l [?25h [?12;25h [58;87H [?25ldone [H
 [?12l [?25h [?12;25h [58;49H [?25lsystemtap-init.el (source)... [K [H
 [?12l [?25h [?12;25h [58;78H [?25ldone [H
 [?12l [?25h [?12;25h [>0c [58d [?25lLoading /usr/share/emacs/site-lisp/site-start.d/systemtap-init.el
(source)...done [K [H [7mFile Edit Options Buffers Tools Lisp-Interaction Help
 [0m [39;49m [27m
 [A
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
```

[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[30m [48;5;250m-UUU:----F1 [39;49m [1m [30m [48;5;250m*scratch*
[0m [39;49m [30m [48;5;250m All L1 (Lisp Interaction) -- [39;49m
[A [2d [?12l [?25h [?12;25h [58d [?25lFor information about GNU Emacs and the GNU system, type C-h C-a. [K [H
[?12l [?25h [?12;25h [58d [?25lFor information about GNU Emacs and the GNU system, type C-h C-a. [K [H [7mFile Edit Options Buffers Tools Help
[0m [39;49m [27m
[A
ACTOR I/me [K
[K
[K
[K
[K
[K
[K
[K
[K
[K

[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[30m [48;5;250m-UU-:----F1
[39;49m [1m [30m [48;5;250mtranslated.txt [0m [39;49m [30m [48;5;250m All L1 (Text) -----------------
-- [39;49m
[A [2d [?12l [?25h [?12;25h [58d [K [2d [58d [?25l(No files need saving) [H
[?12l [?25h [?12;25h [58d [K [?1l > [?12l [?25h [?1049l [39;49m
]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ exit
exit

Script done on Mon 10 Dec 2018 03:19:02 PM PST

```
Script started on Mon 10 Dec 2018 03:19:50 PM PST
 ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [?1034h[sumne007@empress TranslatorFiles]$
ls [K [Kg++ translator.cpp
 ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ ./a.out
opening file
Enter the input file name: partCtest5
Processing <story>
Scanner called using word: wa
Processing <s>
Processing <noun>
SYNTAX ERROR: unexpected wa found in noun
 ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ emacs
translate.txt d.txt [C [C [C [C
 [?1049h [?12;25h [?1h = [H [2J [58d [K [?1l > [?12l [?25h [?1049l [39;49m
 [?1049h [?12;25h [?1h = [H [2J [57d [?25l-UUU:----F1 [1m*scratch* [0m All L1 (Fundamental) --
----------------------------------------------------
 [A [2d [?12l [?25h [?12;25h [58d [?25lLoading /usr/share/emacs/site-lisp/site-start.d/desktop-entry-mode-
init.el (source)... [K [H
 [?12l [?25h [?12;25h [58;87H [?25ldone [H
 [?12l [?25h [?12;25h [58;49H [?25lsystemtap-init.el (source)... [K [H
 [?12l [?25h [?12;25h [58;78H [?25ldone [H
 [?12l [?25h [?12;25h [>0c [58d [?25lLoading /usr/share/emacs/site-lisp/site-start.d/systemtap-init.el
(source)...done [K [H [7mFile Edit Options Buffers Tools Lisp-Interaction Help
 [0m [39;49m [27m
 [A
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
 [K
```

[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[30m [48;5;250m-UUU:----F1 [39;49m [1m [30m [48;5;250m*scratch*
[0m [39;49m [30m [48;5;250m All L1 (Lisp Interaction) -- [39;49m
[A [2d [?12l [?25h [?12;25h [58d [?25lFor information about GNU Emacs and the GNU system, type C-h C-a. [K [H
[?12l [?25h [?12;25h [58d [?25lFor information about GNU Emacs and the GNU system, type C-h C-a. [K [H [7mFile Edit Options Buffers Tools Help
[0m [39;49m [27m
[A
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K

```
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[30m [48;5;250m-UUU:----F1
[39;49m [1m [30m [48;5;250mtranslated.txt [0m [39;49m [30m [48;5;250m All L1 (Text) -----------------
------------------------------------------ [39;49m
[A [2d [?12l [?25h [?12;25h [58d [K [2d [58d [?25l(No files need saving) [H
[?12l [?25h [?12;25h [58d [K [?1l > [?12l [?25h [?1049l [39;49m
]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ exit
exit

Script done on Mon 10 Dec 2018 03:20:35 PM PST
```

```
Script started on Mon 10 Dec 2018 03:21:07 PM PST
  ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles  [?1034h[sumne007@empress TranslatorFiles]$ g++
translatorc.pp    [K    [K    [K    [K.cpp
  ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ ./a.out
opening file
Enter the input file name: partCtest6
Processing <story>
LEXICAL ERROR: apple is not a valid token
Scanner called using word: apple
Processing <s>
Processing <noun>
SYNTAX ERROR: unexpected apple found in noun
  ]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ emacs
translated.txt
  [?1049h  [?12;25h  [?1h  =  [H  [2J  [58d  [K  [?1l  >  [?12l  [?25h  [?1049l  [39;49m
  [?1049h  [?12;25h  [?1h  =  [H  [2J  [57d  [?25l-UUU:----F1   [1m*scratch*   [0m  All L1   (Fundamental) --
----------------------------------------------------
  [A  [2d  [?12l  [?25h  [?12;25h  [58d  [?25lLoading /usr/share/emacs/site-lisp/site-start.d/desktop-entry-mode-
init.el (source)...  [K  [H
  [?12l  [?25h  [?12;25h  [58;87H  [?25ldone  [H
  [?12l  [?25h  [?12;25h  [58;49H  [?25lsystemtap-init.el (source)...  [K  [H
  [?12l  [?25h  [?12;25h  [58;78H  [?25ldone  [H
  [?12l  [?25h  [?12;25h  [>0c  [58d  [?25lLoading /usr/share/emacs/site-lisp/site-start.d/systemtap-init.el
(source)...done  [K  [H  [7mFile Edit Options Buffers Tools Lisp-Interaction Help
  [0m  [39;49m  [27m
  [A
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
  [K
```

[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[30m [48;5;250m-UUU:----F1 [39;49m [1m [30m [48;5;250m*scratch*
[0m [39;49m [30m [48;5;250m All L1 (Lisp Interaction) -- [39;49m
[A [2d [?12l [?25h [?12;25h [58d [?25lFor information about GNU Emacs and the GNU system, type C-h C-a. [K [H
[?12l [?25h [?12;25h [58d [?25lFor information about GNU Emacs and the GNU system, type C-h C-a. [K [H [7mFile Edit Options Buffers Tools Help
[0m [39;49m [27m
[A
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K

```
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[K
[30m [48;5;250m-UUU:----F1
[39;49m [1m [30m [48;5;250mtranslated.txt [0m [39;49m [30m [48;5;250m All L1 (Text) -----------------
------------------------------------------ [39;49m
[A [2d [?12l [?25h [?12;25h [58d [K [2d [58d [?25l(No files need saving) [H
[?12l [?25h [?12;25h [58d [K [?1l > [?12l [?25h [?1049l [39;49m
]0;sumne007@empress:~/CS421/CS421Progs/TranslatorFiles [sumne007@empress TranslatorFiles]$ exit
exit

Script done on Mon 10 Dec 2018 03:22:13 PM PST
```