

```

// This program converts one representation of abstract syntax trees
// to a different representation. Currently, the conversion goes
// from a Prolog term to a Schem list

#include <fstream>
#include <iostream>
#include <stdlib.h>
// #include "tokenizer.h"

using namespace std;

int state;    // record the state of the tokenizer
char ch;      // the next character in the input stream

bool alphanum(char ch) {
    // Alphanumeric characters
    return (isalpha(ch) || isdigit(ch));
}

bool single(char ch) {
    // These are the single character tokens
    return (/* (ch == '!') || */ (ch == '{') || (ch == '}') || (ch == ',') ||
            (ch == '(') || (ch == ')') || (ch == '[') || (ch == ']') || (ch == '%') ||
            (ch == '-') || /* (ch == '/') || */ (ch == ';') || (ch == '?') ||
            (ch == ':') || (ch == '#') || (ch == '.') || (ch == '"') ||
            (ch == '!' || (ch == '||')));
}

void transition() {
    // the standard transitions: if
    if (single(ch)) {
        state = 0;
    } else if (ch == '=' || ch == '>' || ch == '!') {
        state = 1;
    } else if (ch == '<') {
        state = 2;
    } else if (ch == '/') {
        state = 3;
    } else if (ch == '&') {
        state = 4;
    } else if (ch == '|') {
        state = 5;
    } else if (isalpha(ch)) {
        state = 6;
    } else if (isdigit(ch)) {
        state = 7;
    } else if (ch == ' ' || ch == '\n' || ch == '\t') {
        state = 8;
    }
    else {
        state = 9;
    }
}

string getToken(istream & ins) {

```

```

// find and return the next token
// leading blanks are skipped

string messagePart = "";

ins.peek(); // to set the eof flag
if (! ins.eof()) {
    transition();
    switch (state) {
    case 0:
        messagePart = ch;
        break;
    case 1:
        messagePart = ch;
        ins.get(ch);
        if (ins.eof()) break;
        if (ch == '=')
            messagePart += ch;
        else
            return messagePart;
        break;
    case 2:
        messagePart = ch;
        ins.get(ch);
        if (ins.eof()) break;
        if (ch == '=' || ch == '<')
            messagePart += ch;
        else
            return messagePart;
        break;
    case 3:
        messagePart = ch;
        ins.get(ch);
        if (ins.eof()) break;
        if (ch == '/')
            while (! ins.eof() && ch != '\n') {
                messagePart += ch;
                ins.get(ch);
            }
        else
            return messagePart;
        break;
    case 4:
        messagePart = ch;
        ins.get(ch);
        if (ins.eof()) break;
        if (ch == '&')
            messagePart += ch;
        else
            return messagePart; // not sure if I like this
        break;
    case 5:
        messagePart = ch;
        ins.get(ch);
        if (ins.eof()) break;
        if (ch == '|')
            messagePart += ch;

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```

else
    return messagePart;
break;
case 6:
messagePart = ch;
ins.get(ch);
while (! ins.eof() && (alphanum(ch) || ch == '_')) {
    messagePart += ch;
    ins.get(ch);
}
return messagePart;
break;
case 7:
messagePart = ch;
ins.get(ch);
while (! ins.eof() && isdigit(ch)) {
    messagePart += ch;
    ins.get(ch);
}
return messagePart;
break;
case 8:
//      transition();
messagePart = ch;
ins.get(ch);
while (! ins.eof() && (ch == ' ' || ch == '\t' || ch == '\n')) {
    messagePart += ch;
    ins.get(ch);
}
return messagePart;
break;
case 9:
break;
}
ins.get(ch);
}
return messagePart;
}

int main(int argc, char** argv) {
if (argc < 2) {
    cerr << "usage: ast.exe <file>\n";
    exit(0);
}

ifstream ins;
ins.open(argv[1]);
if (ins == 0) {
    cerr << "invalid file: " << argv[1] << endl;
    exit(0);
}

string prevToken = "";
string nextToken = getToken(ins);

cout << "(define ptree (quote";
while (!ins.eof()) {

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```

if (nextToken == "(") {
    cout << nextToken << prevToken << " ";
    prevToken = getToken(ins);
}
else {
    if (prevToken == "[")
        cout << "(" << " ";
    else if (prevToken == "]")
        cout << ")" << " ";
    else if (prevToken != ",")
        cout << prevToken << " ";

    prevToken = nextToken;
}
nextToken = getToken(ins);
}
if (prevToken == "]")
    cout << ")" << " ";
else
    cout << prevToken;
cout << ")\n";
return 0;
}

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder