```cpp
#include <iostream>
#include <vector>
#include <unordered_set>
#include <unordered_map>
#include <sstream>

using namespace std;

// Function to split a string by delimiter
vector<string> split(const string& s, char delimiter) {
    vector<string> tokens;
    string token;
    istringstream tokenStream(s);
    while (getline(tokenStream, token, delimiter)) {
        tokens.push_back(token);
    }
    return tokens;
}

int main() {
    unordered_map<char, vector<string>> grammar;
    char startSymbol;

    // Read the grammar from standard input
    string line;
    while (getline(cin, line)) {
        vector<string> parts = split(line, ' ');
        char nonTerminal = parts[0][0];
        if (grammar.empty()) {
            startSymbol = nonTerminal;
        }
        vector<string> rules;
        for (size_t i = 2; i < parts.size(); ++i) {
            rules.push_back(parts[i]);
        }
        grammar[nonTerminal] = rules;
    }

    // Check if the grammar is LL(1)
    cout << "Grammar Rules:" << endl;
```

```
    for (auto& it : grammar) {
        cout << it.first << " -> ";
        for (const string& rule : it.second) {
            cout << rule << " | ";
        }
        cout << endl;
    }

    cout << "Start Symbol: " << startSymbol << endl;

    cout << "Checking if the grammar is LL(1)..." << endl;

    // Perform LL(1) check
    cout << "The grammar is LL(1)." << endl; // Placeholder
, replace with actual LL(1) check

    return 0;
}
```