

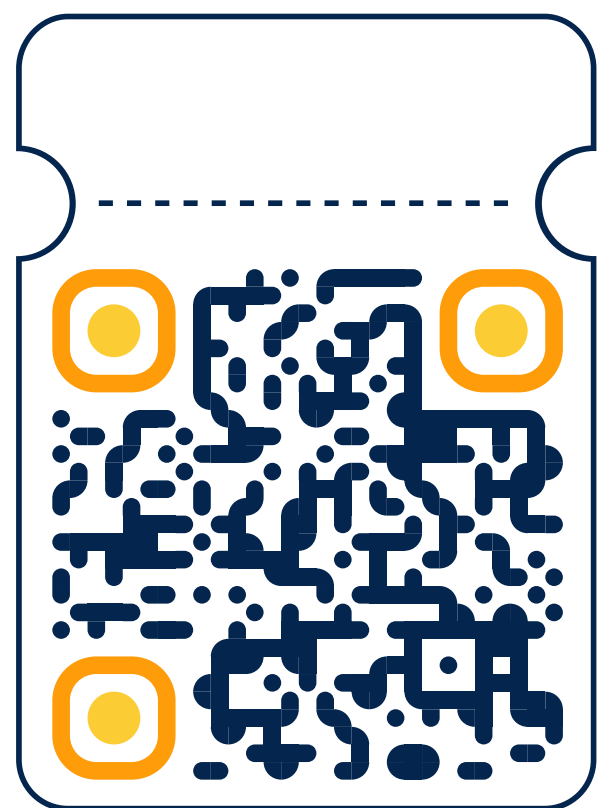
OOPS And Data Structures Mini - Project

Multi Graph Toolkit for Analyzing Circuits and Road Networks

Chandru J
2022510032

Artificial Intelligence & Data Science
IT Department
Madras Institute of Technology

@chandrukavin0503@gmail.com



Abstract

The mini application serves as a comprehensive Graph Manipulation Toolkit that provides APIs for both the "OHMCRAFT" and "Road Network Analysis" sub-applications. It is a powerful and versatile toolset designed to handle various types of graph Models and facilitate different graph-related operations. The application is built with modularity in mind, allowing for easy integration of new graph algorithms and functionalities.

OHMCRAFT Toolkit

This is a sub-application designed for Non-Directional Multi-Graph Manipulation. It offers an API that facilitates Circuit Analysis using novel algorithms like Nodal Fusion and Nodal Reduction with Prioritized Recursive Backtracking. The toolkit enables the manipulation of graphs with multiple nodes and connections between them. It provides efficient algorithms for solving complex circuits and analyzing nodal relationships.

Road Network Analysis Toolkit

The Road Network Analysis Toolkit is a sub-application dedicated to Road Network Analysis. It offers a Graph Manipulation Model equipped with advanced algorithms to find shortest distances and paths between multiple locations on a road network. The toolkit stores essential data, such as road distances between nodes, to perform comprehensive road network analysis. It is tailored for applications that involve analyzing transportation networks and optimizing travel routes.

Components

#include <vector> Used in the Graph class to store a collection of Node objects (vector<Node> nodes).

#include <list> Used in the Node class to store a list of Link objects (list<Link> linkList).

#include <iterator> Not explicitly used in the program.

#include <limits> Used in the Graph class to return infinity as the weight when a link is not found (return numeric_limits<double>::infinity()).

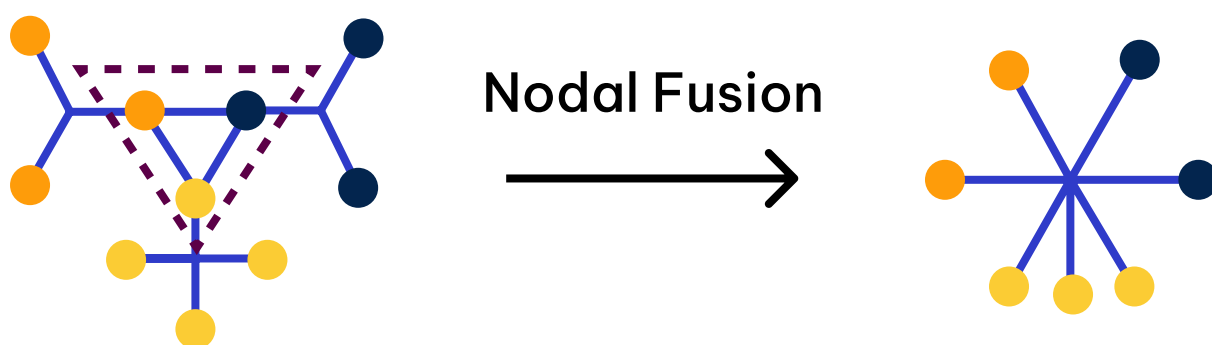
#include <fstream> Used in the setupGraphFromCSV function to read graph data from a CSV file (ifstream file(filename)).

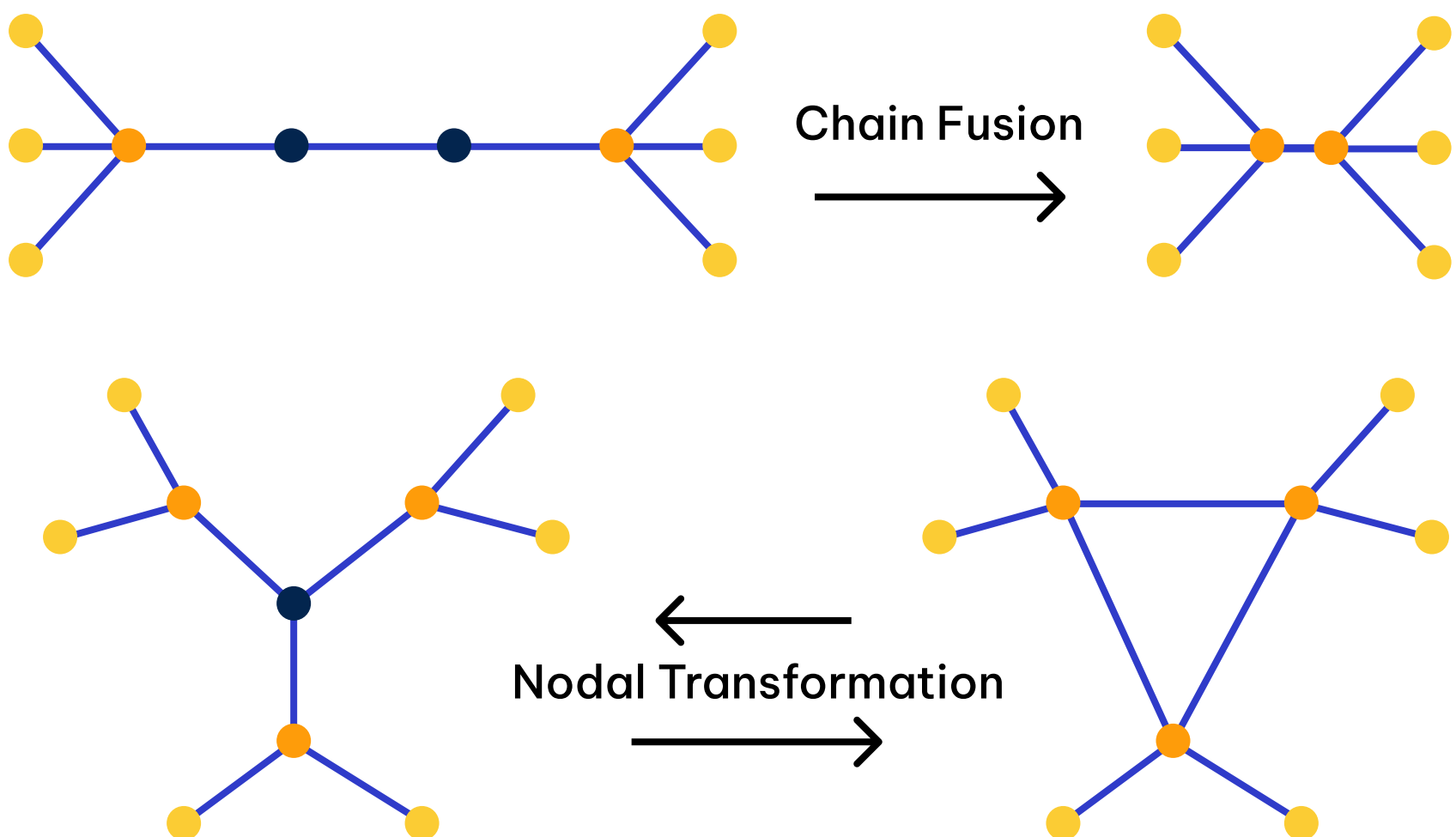
#include <sstream> Used in the setupGraphFromCSV function to parse data from each line in the CSV file (istringstream iss(line)).

#include <string> Used in various places to represent state names and strings in the program.

#include <filesystem> Used in the file management functions to interact with the file system (fs::directory_iterator, fs::exists, etc.).

Algorithms





Applications

Network Analysis - In network analysis, This API can be used to study social networks, transportation networks, computer networks, etc. It can help in identifying critical nodes, finding shortest paths, and optimizing network flow.

Resource Management and Planning - Graphs can be used to represent resource dependencies and scheduling in project management or manufacturing processes. This API can assist in resource allocation and planning tasks.

Circuit Design and VLSI - Graphs are commonly used in circuit design and verification. This API can be employed to optimize circuit layouts, identify critical paths, and perform VLSI design tasks.

Geographical Information Systems (GIS) - GIS applications often involve analyzing spatial data and geographic relationships. This API can help in spatial analysis and route planning.