Declarative Programming project: Taxi Company

Jeroen Heymans

January 13, 2012

Contents

1 Introduction

This is the report for the project for the course Declarative Programming, a master-course teached at the Vrije Universiteit Brussel. In this report we describe the design and the functionality that has been implemented.

The general idea of the project was to build a program that would calculate the routes for the taxis of the taxi company El Cabinero. This company works in a given city that is described in a graph.

2 The implemented program

2.1 Path calculation

Since we needed to implement the discovery of routes, we obviously needed an algorithm that could calculate a short path. Multiple algorithms exist that can calculate the shortest (or almost shortest) path in a graph. We have firstly implemented a version of Dijkstra's wellknown algorithm. With this, we could get a good indication of what the shortest path would be. Unfortunately, Dijkstra is quite heavy in the use of memory. This slowed the program down significantly. As a benchmark, we took the calculation time that was necessary to calculate the path between node # 0 and node # 2499, two of the utmost nodes in the graph.

With our implementation of Dijkstra, it took half an hour to calculate this path. In order to get the execution time down, we choose to add a heuristic to the algorithm. The problem with the naïve Dijkstra implementation was that we could have situations where the shortest path goes from left to right but that the algorithm would first see if there was any shortest path to the left. By adding a heuristic, we could guide the algorithm more to the correct direction.