Demonstration of the Algorithm

This section presents an execution example of the algorithm that was detailed above (Section *Implementation of the Algorithm*), starting from the presentation of the road map on which the execution will be demonstrated to the changes that are made in the data structures of the algorithm, over four units of time.

Figure 1 presents a map of four different intersections.

Figure 2 presents the map's transformation from Fig.1 to a weighted and directed graph.

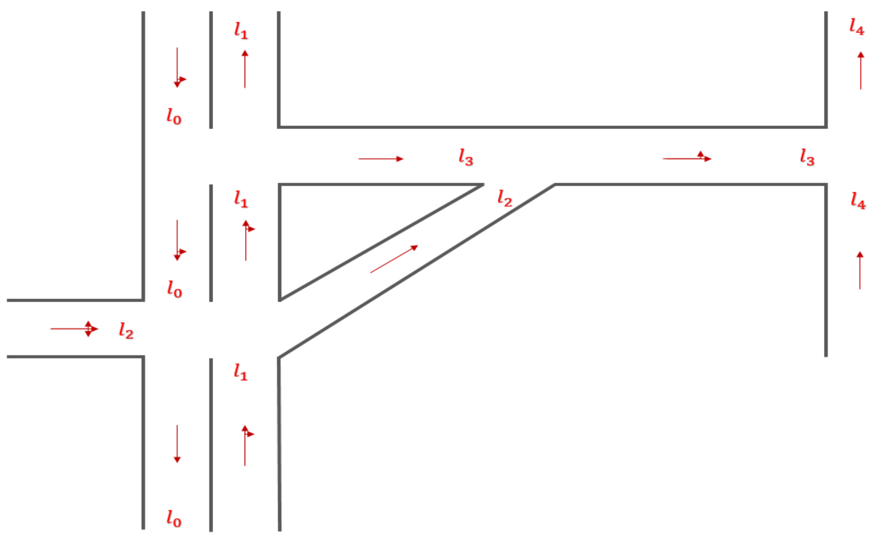

Figure 1: Map example

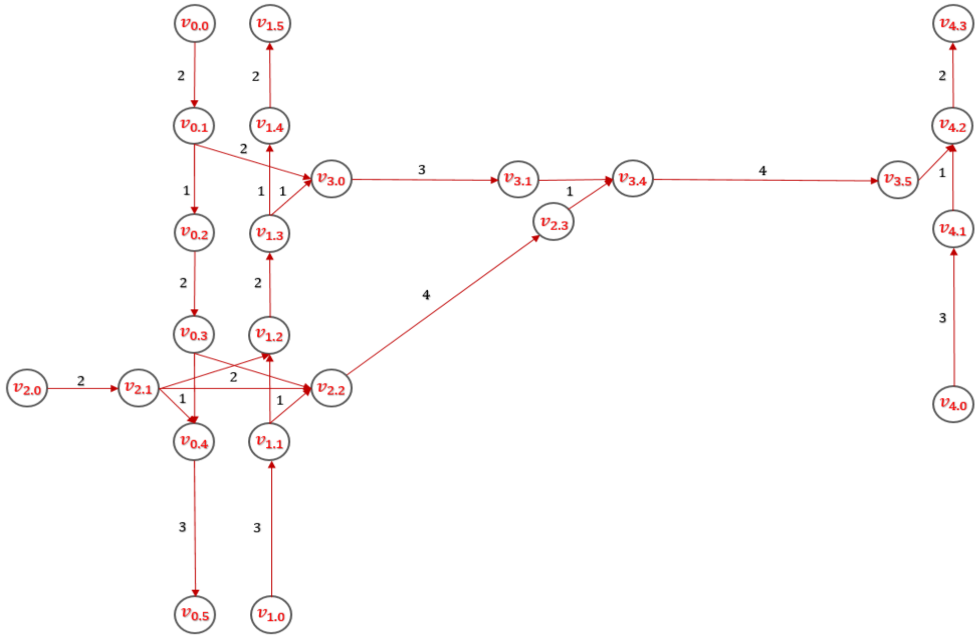

Figure 2: Map transformation

Figure 3 presents the initial data values in the data structure of the graph presented in Fig.2. The red arrows show the possible route for each node.

Figure 4 presents the changes in time $t = 0$, of the values of the data in the data structure that is presented in Fig.3, when there are in the queue two vehicles $[x_1(v_{4.0}, v_{4.3}), x_2(v_{0.0}, v_{0.5})]$ where the tuple means (source, destination). The algorithm finds the path $P_1 = [v_{4.0}, v_{4.1}, v_{4.2}, v_{4.3}]$ for vehicle x_1 first, since x_1 is the first in the queue, and then finds the path $P_2 = [v_{0.0}, v_{0.1}, v_{0.2}, v_{0.3}, v_{0.4}, v_{0.5}]$ for vehicle x_2. In addition, after finding every path, the algorithm updates the dynamic movements lists $ML's$ of all the nodes in every path and all the nodes which have conflict with one of the nodes that in some path. For example, the list ML of node $v_{1.1}$ updates since it is in the conflicts list of node $v_{0.3}$ that is in path P_1.

Figure 5 presents the changes in time $t = 1$, of the values of the data in the data structure that is presented in Fig.4, when the is empty.

Figure 6 presents the changes in time $t = 2$, of the values of the data in the data structure that is presented in Fig.5, when there is in the queue one vehicle $[x_3(v_{2.0}, v_{4.3})]$. For this vehicle the algorithm finds the path $P_1 = [v_{2.0}, v_{2.1}, v_{2.2}, v_{2.3}, v_{3.4}, v_{3.5}, v_{4.2}, v_{4.3}]$ and, updates the dynamic movements lists $ML's$ of all the nodes in the path and, all the nodes which has conflict with one of the nodes that in the path. Note that in node $v_{2.1}$, the vehicle x_3 is in the list ML with value $at = 6$, since the vehicle x_1 that in the list ML, with value $at = 5$. So, vehicle x_3 should wait for 2 time-units to cross safely.

Figure 7 presents the changes in time $t = 3$, of the values of the data in the data structure that is presented in Fig.6, when there is in the queue one more vehicle $[x_4(v_{1.0}, v_{1.5})]$. For this vehicle, the algorithm finds the path $P_1 = [v_{1.0}, v_{1.1}, v_{1.2}, v_{1.3}, v_{1.4}, v_{1.5}]$ and, updates the dynamic movements lists $ML's$ of all the nodes in the path and, all the nodes which have conflict with one of the nodes that in the path. Note that in node $v_{1.1}$, the vehicles x_1, x_3 are in the list ML with value $at = 5, 6$ respectively, and since those vehicles are in the list ML, vehicle x_4 is in the list ML with value $at = 8$.

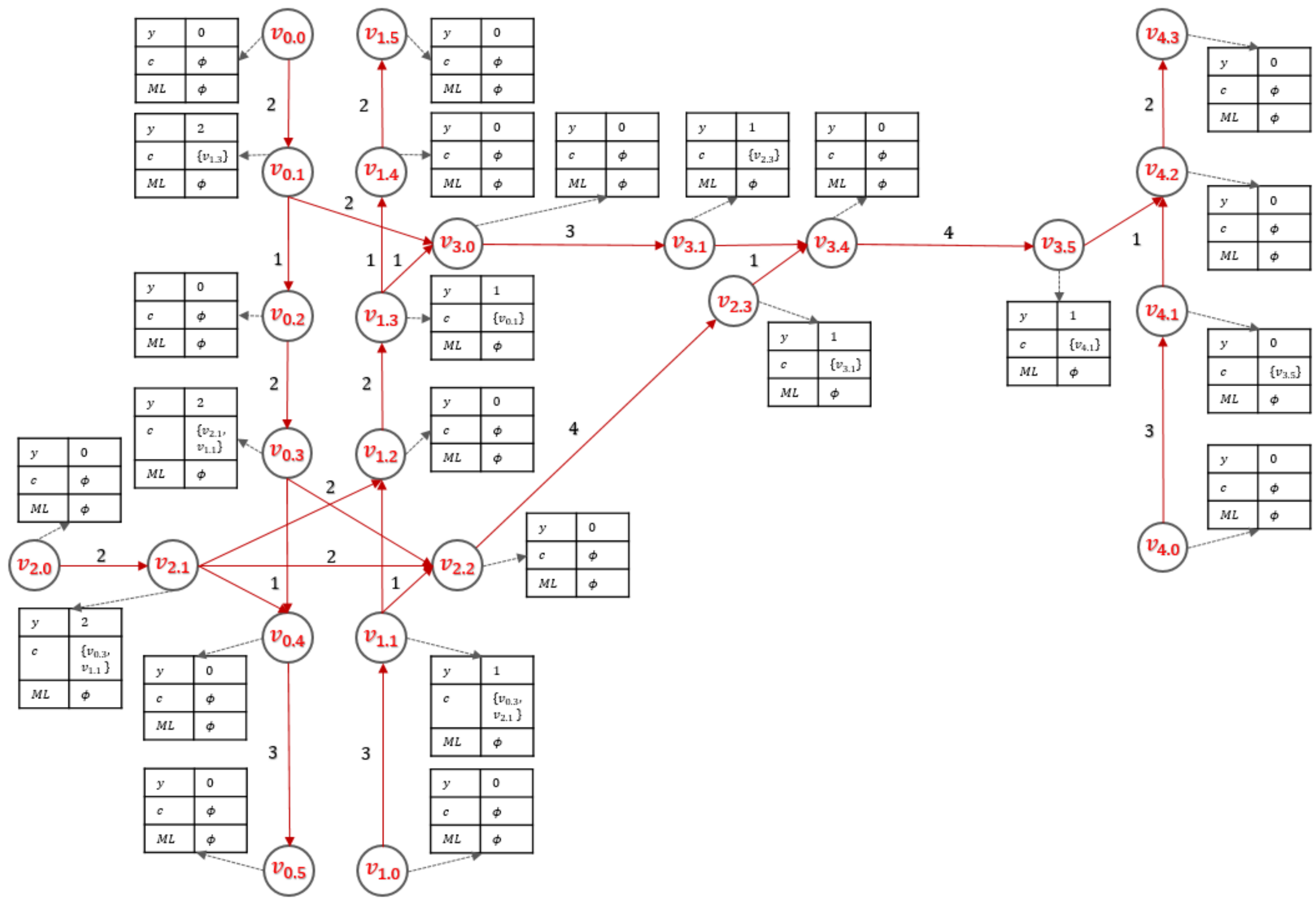

Figure 3: Initial data structure

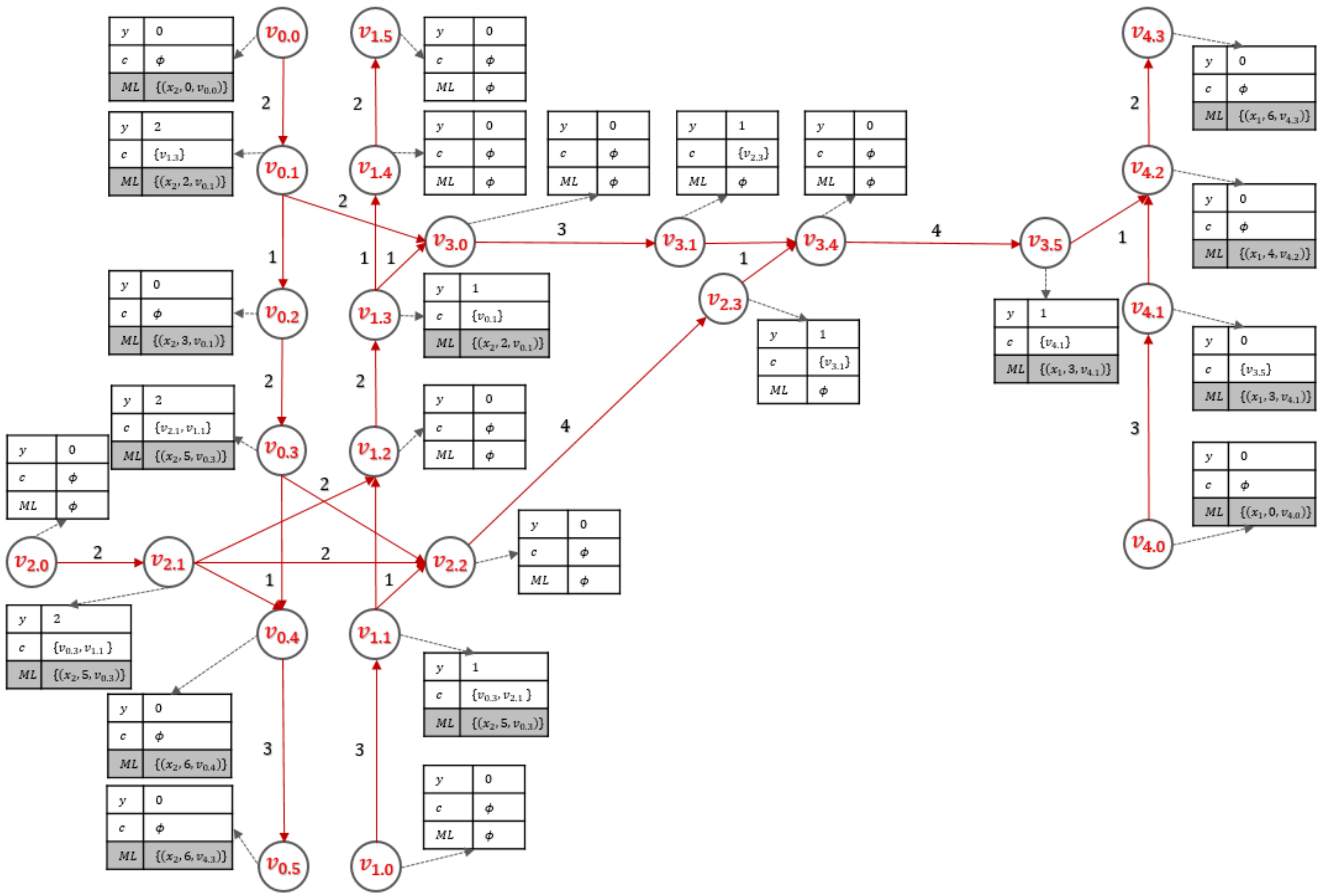

Figure 4: The graph's data structure at time $t = 0$

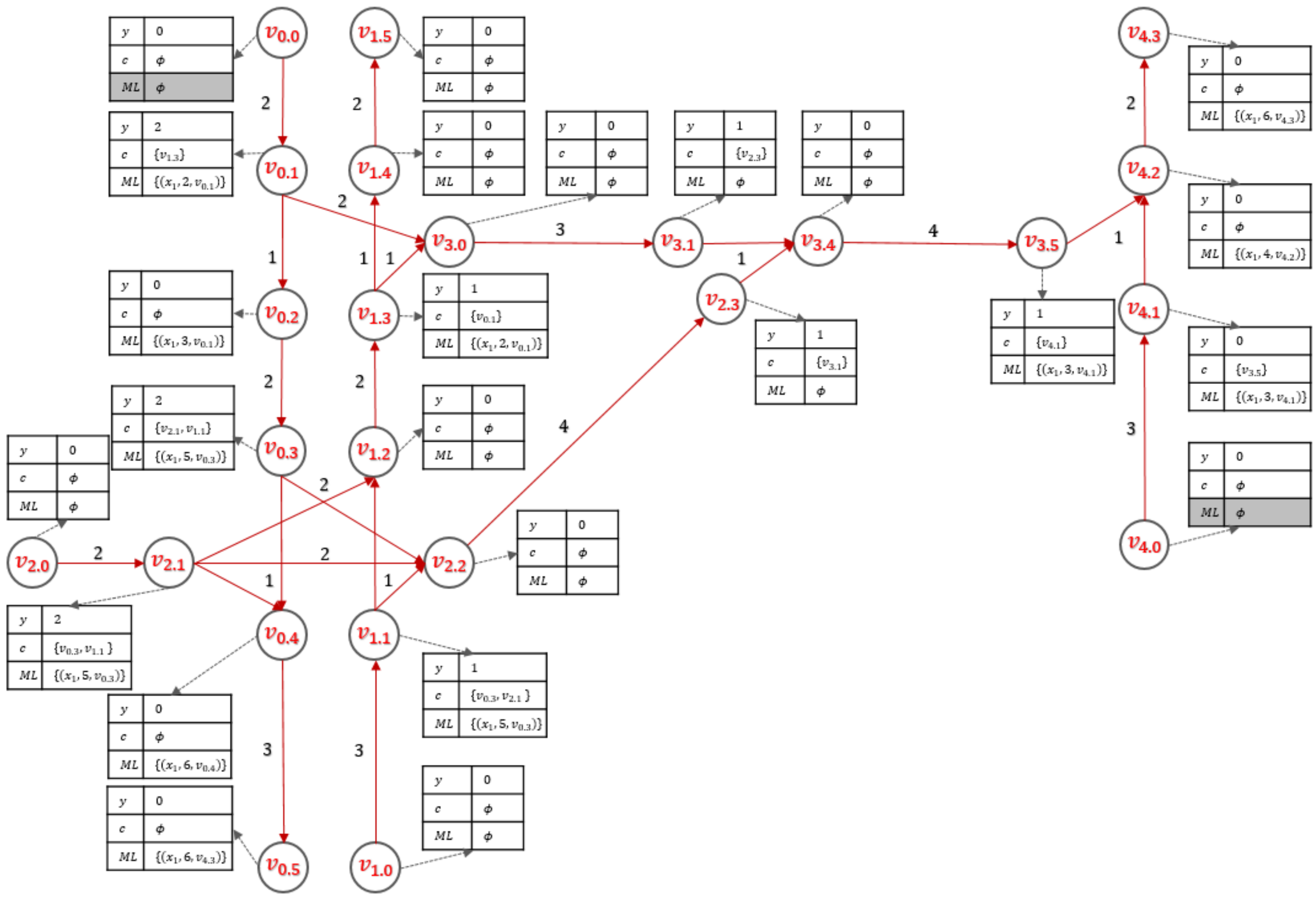

Figure 5: The graph's data structure at time $t = 1$

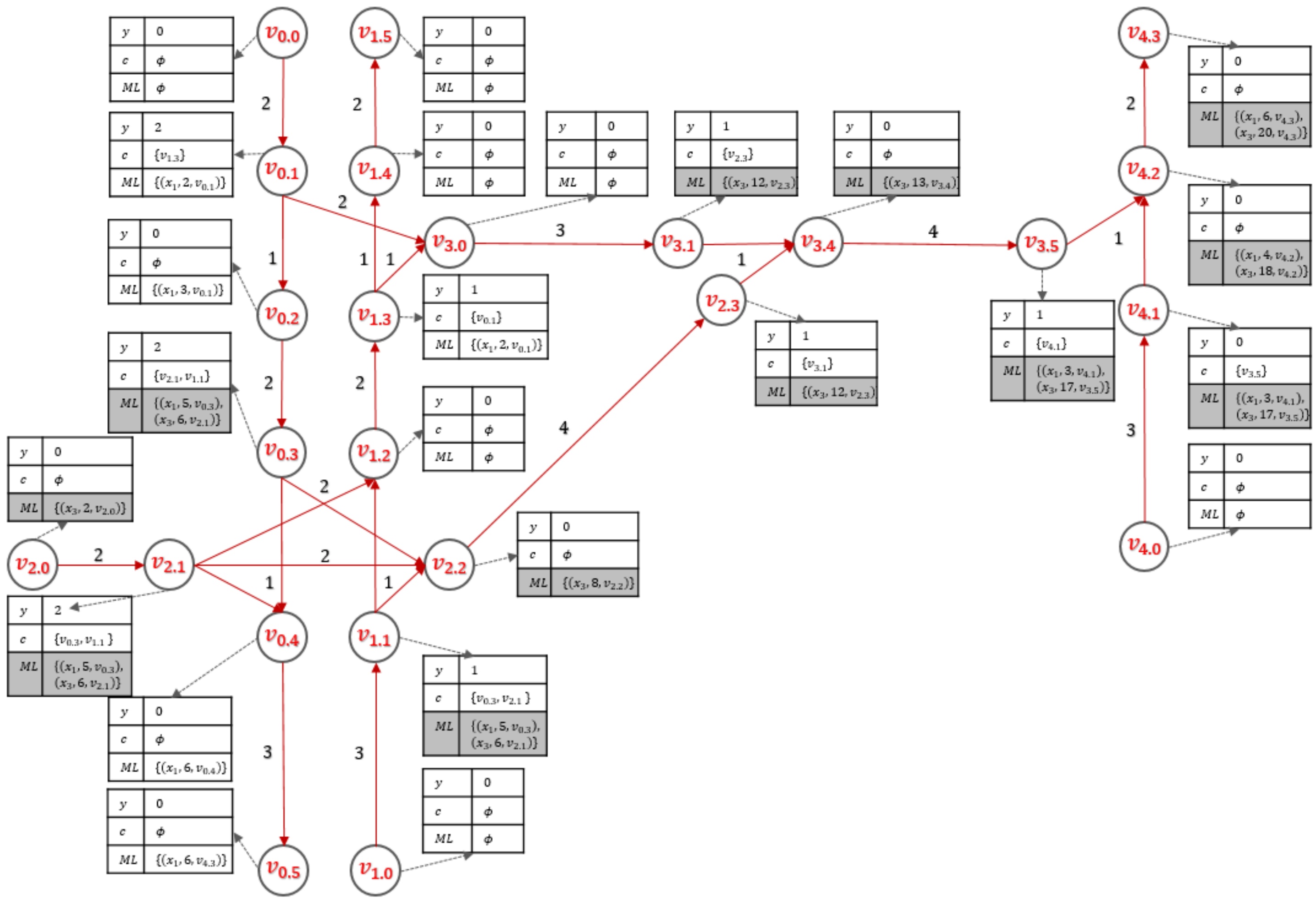

Figure 6: The graph's data structure at time $t = 2$

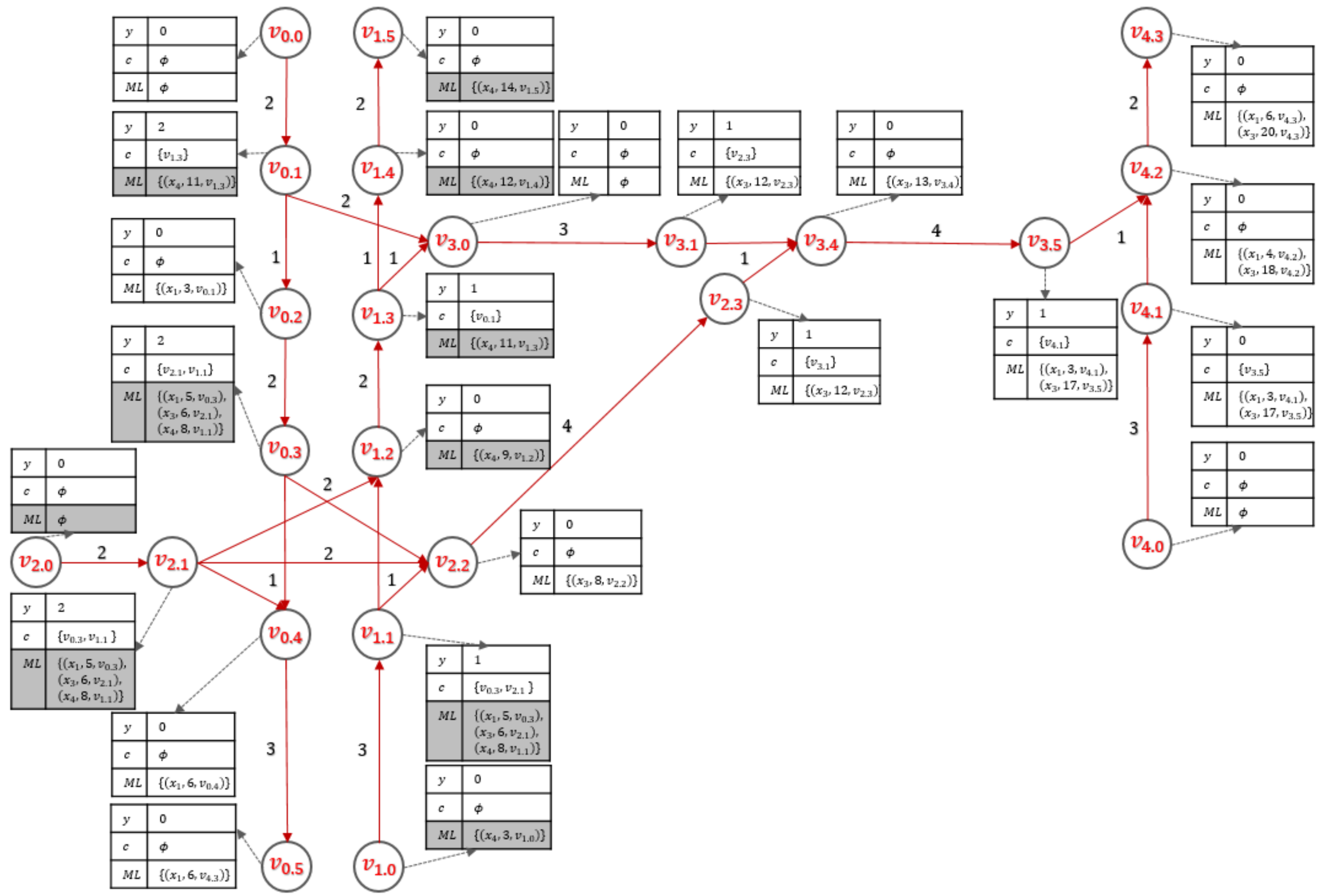

Figure 7: The graph's data structure at time $t = 3$