Updates-Aware Graph Pattern based Node Matching

Guohao Sun[1], Guanfeng Liu[1], Yan Wang[1] and Xiaofang Zhou[2]

[1] *Department of Computing, Macquarie University, Sydney, NSW 2109, Australia*

[2] *School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane 4072, Australia*

[1] guohao.sun@students.mq.edu.au; [1] {guangfeng.liu, yan.wang}@mq.edu.au; [2] zxf@itee.uq.edu.au

The Proof of Theorem 1: When U_{Pa} is applied to G_P prior to U_{Pb}, suppose $U_{Pa} \sqsupseteq U_{Pb}$. Then, according to the definition of an elimination relationship of *Type I*, $Can\_N(U_{Pa}) \sqsupseteq Can\_N(U_{Pb})$, namely, for any node $n_i \in Can\_N(U_{Pb})$, n_i is also in $Can\_N(U_{Pa})$. When U_{Pb} is applied to G_D prior to U_{Pa}, suppose U_{Pa} and U_{Pb} do not have the elimination relationship. Then, there is at least one node n_i such that $n_i \in Can\_N(U_{Pb})$ and $n_i \notin Can\_N(U_{Pa})$. However, this contradicts $n_i \in Can\_N(U_{Pa})$ when U_{Pa} is applied to G_D. Therefore, *Theorem 1* is proven. □

The Proof of Theorem 2: When U_{Da} is applied to G_D prior to U_{Db}, suppose $U_{Da} \succeq U_{Db}$. Then, according to the definition of the elimination relationships of *Type II*, $Aff\_N(U_{Da}) \supseteq Aff\_N(U_{Db})$, namely, for any node $n_i \in Aff\_N(U_{Db})$, n_i is also in $Aff\_N(U_{Da})$. When U_{Db} is applied to G_D prior to U_{Da}, suppose U_{Da} and U_{Db} do not have the elimination relationship. Then, there is at least one node n_i such that $n_i \in Aff\_N(U_{Db})$ and $n_i \notin Aff\_N(U_{Da})$. However, this contradicts $n_i \in Aff\_N(U_{Da})$ when U_{Da} is applied to G_D. Therefore, *Theorem 2* is proven. □

The Proof of Theorem 3:

- If V_a and V_b are in the same partition (V_a, $V_b \in P_i$), and there exists another path from V_a to V_b in the data graph, the length of which is less than $SP_D(V_a, V_b)$.
  a) Suppose $OB(P_i) = \emptyset$. Then based on the Dijkstra's algorithm, there exists at least one edge $e(V_c, V_d)$ in the shortest path with $V_c \in P_i$ and $V_d \in P_j$, which contradicts to $OB(P_i)$=$\emptyset$;
  b) Suppose $OB(P_i) \neq \emptyset$. Since we recursively combine the partition of the node in $OB(P_i)$, for the combined partition, there is no outer bridge node. Therefore, there exists at least one edge $e(V_c, V_d)$ in the shortest path where V_c is in the combined partition and V_d is not in the combined partition, which contradicts that there is no outer bridge node in the combined partition.
- If V_a and V_b are in the different partitions ($V_a \in P_i$, and $V_b \in P_j$).
  a) Suppose $OB(P_i) = \emptyset$, which means any of the nodes in partition P_i cannot connect with any of nodes in P_j. Then the shortest path length between these nodes in P_i and P_j is infinity;
  b) Suppose $OB(P_i) \neq \emptyset$, and there exists another path from V_a to V_b in the data graph, the length of which is less than $SP_D(V_a, V_b)$. Because we first compute $SP_D(V_a, V_c)$ ($V_c \in IB(P_i)$ and $V_c \in OB(P_j)$), $SP_D(V_c, V_d)$, and then get the least value among the summation of $SP_D(V_a, V_c)$ and $SP_D(V_c, V_d)$. So, there exists at least one edge $e(V_c, V_d)$ in the shortest path with $V_d \notin P_j$, which contradicts that V_c is one of the outer bridge nodes in P_j.

Therefore, *Theorem 3* is proven. □