Proposal of CS263: Formal Verification of Hungarian Algorithm

Shushu Wu, 517030910338
Zhihui Xie,
Yiwen Song, 517030901380
Weizhe Wang, 517030910381

1 Introduction to Hungarian Algorithm

The Hungarian Algorithm [3] is an $O(mn)$ matching algorithm for bipartite graph proposed by H.W.Kuhn in 1955. Till now, there are many different variants of Hungarian Algorithms. Therefore, we need to specify that in our proposal Hungarian Algorithm just means Kuhn-Munkres Algorithm, the original version.
The Hungarian Algorithm is designed to find the maximum matching/perfect matching for an unweighted bipartite graph. There are some formal definitions used in the description of this algorithm.

Definition 1. *A **bipartite graph** is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{V}_1 \bigcup \mathcal{V}_2, \mathcal{V}_1 \bigcap \mathcal{V}_2 = \emptyset$, while $\forall e = (v_1, v_2) \in \mathcal{E}, v_1 \in \mathcal{V}_1 \& v_2 \in \mathcal{V}_2$ or $v_1 \in \mathcal{V}_2 \& v_2 \in \mathcal{V}_1$.*

Definition 2. *A **matching** is a collection of edges $\mathcal{M} \subset \mathcal{E}$, where $\forall m_1, m_2 \in \mathcal{M}, m_1$ and m_2 don't have a common node. From this definition we can also intuitively define **matched nodes, matched edges, unmatched nodes, unmatched edges**. We omit the definition here.*

Definition 3. *A **maximum matching** is a matching that makes $|\mathcal{M}|$ the largest one.*

Definition 4. *A **perfect matching** is a matching that lets all nodes be in the matching. Obviously a perfect matching is a maximum matching.*

Definition 5. *An **augmenting path** P is a path that connects two unmatched nodes and sequentially passes matched edges and unmatched edges.*

And the algorithm is shown in Algorithm 1. In order to verify the Hungarian Algorithm, we need to prove that:

Theorem 1. *For all bipartite graphs, given a matching $\mathcal{M}$, if there exists no augmenting path, then $\mathcal{M}$ is a maximum matching.*

2 Proof Sketch

We have found a Coq library containing basic declarations for graph theory [1]. On the basis of the library, we can set up our own definitions related to the algorithm.

Algorithm 1 Hungarian Algorithm

$\mathcal{M} \leftarrow \emptyset$;
$\mathcal{A} \leftarrow \mathcal{V}$;
repeat
  $x \leftarrow \mathcal{A}[0]$;
  $S \leftarrow \{x\}$, $T \leftarrow \emptyset$;
  while $!N(S) \subset T$ **do**
    $y \leftarrow N(S) - T$;
    if y IS MATCHED **then**
      $S \leftarrow S \bigcup \{z\}, T \leftarrow T \bigcup \{y\}$ for all $yz \in \mathcal{M}$;
    else
      $\mathcal{M} \leftarrow \mathcal{M} \oplus E(P), \mathcal{A} \leftarrow \mathcal{A} - \{x, y\}$;
      BREAK;
    end if
  end while
  $\mathcal{A} \leftarrow \mathcal{A} - x$;
until $\mathcal{A} = \emptyset$

Then we may use the property of the augmenting path, where there are one more unmatched points than the matched points. By this property we may prove that if there does not exist an augmenting path, the graph is either perfect matched, or maximum matched. Because if there exists a larger matching, we can always find an augmenting path.

3 Possible Goals

A popular variant of Hungarian Algorithm is Hopcroft-Karp Algorithm [2] developed by J.E. Hopcroft, which is more time-efficient than Hungarian Algorithm. The time complexity of Hopcroft-Karp Algorithm is $O(n^{1/2}m)$, which is $o(n^{1/2})$ times faster than the Hungarian algorithm. It marks the distance of each node to the original node during the BFS of augmenting path so that time may be saved. If we have more time after proving the Hungarian Algorithm, we would like to verify the Hopcroft-Karp Algorithm as a bonus.

References

[1] FreeSoftwareFoundation. graph-basics. `https://github.com/coq-contribs/graph-basics/`.

[2] Hopcroft, J. E., and Karp, R. M. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. In *SIAM Journal on computing* (1973), vol. 2, pp. 225–231.

[3] Kuhn, H. W. The hungarian method for the assignment problem. *Naval research logistics quarterly 2*, 1-2 (1955), 83–97.