Dijkstra's Algorithm Verification

Yazhe Feng

January 27, 2019

1 Algorithm Implementation

1.1 Pseudocode(Idris)

```
−− data structures
Node : (n : Nat) → Fin n
NodeSet : List Node          −− ‘NodeSet‘ can represents edges to each node
Graph : (List Node, List NodeSet)

dijkstras : (weight : Type) →
            (gtW : weight → weight → Bool) →
            (add : weight → weight → weight) →
            (size : Nat) →
            (source : Node size) →
            (graph : Graph size weight) →
            (Vect size weight)
dijkstras weight gtW add size source g@(nodes, edges)
      = runDijkstras weight gtW add size size g dist reflProof nodes
            where
                  dist = list of nodes and their distance to source. $dist[source] = 0$
                  reflProof = proof of (lte size size = True)

−− if unexplored is Nil, then we have calculated the min distance for all nodes
runDijkstras : (weight : Type) →
            (gtW: weight → weight → Bool) →
            (add : weight → weight → weight) →
            (size : Nat) →
            (size' : Nat) → $--numberofunexplorednodes$
            (graph : Graph size weight) →
            (dist : Vect size weight) →
            (lte size' size = True) →
```

```
          (unexplored : Vect size' (Node size)) →
          (Vect size weight)
runDijkstras _ _ _ _ Z g dist _ Nil = dist
runDijkstras _ _ _ _ (S s') g dist refl ((MKNode x) :: xs)
     = let dist' = dist after updating distance for each adjacent node of x in
          call (runDijkstras _ _ _ _ s' g dist' refl xs)
```

1.2 Time Analysis

Time Analysis goes here. $O(n^2)$

1.3 Proof of Correctness

Lemma. *Lemma goes here*

Proof. Base Case:
Inductive Hypothesis:
Inductive Step □

Collaboration

Name1, Name2, ...