Directed Planar Reachability is in Unambiguous Log-Space

(Bourke, Tewari, Vinodchandran)

UL Complexity Class

- UL denotes the unambiguous subclass of NL.
- A decision problem L is in UL, if and only if there exists a nondeterministic log-space machine M deciding L such that, for every instance x, M has at most one accepting computation on input x.
- To prove: Planar reachability can be decided by a nondeterministic machine in log-space with at most one accepting computation.

CLAIM 1

- [Allender 2005]: The PLANARREACH problem log space many-one reduces to GGR.

Grid Graphs

- A $n \times n$ grid graph is a directed graph whose vertices are $[n] \times [n] = \{1, \ldots, n\} \times \{1, \ldots, n\}$ so that if $((i1, j1), (i2, j2))$ is an edge then $|i1 - i2| + |j1 - j2| = 1$.
- Grid graphs are a very natural subclass of planar graphs.
- Grid Graph Reachability problem reduces to its complement.

Min-Unique Graph

- Min-unique graph is a directed graph with positive weights associated with each edge where for every pair of vertices u, v, if there is a path from u to v, then there is a unique minimum weight path from u to v.
- Here, the weight of a path is the sum of the weights on its edges.

Claim 2

- [REINHARDT AND ALLENDER 2000]. Let G be a class of
- graphs and let H = (V, E) ∈ G. If there is a polynomially bounded log-space computable function f that on input H outputs a weighted graph f(H) so that
- (1) f (H) is min-unique and
- (2) H has an st-path if and only if f (H) has an st-path

Algorithm 1:Determining whether d(v) ≤ k or not.

Input: (G, v, k, ck , Σk)

Output: true if d(v) ≤ k else false

```
Initialize count ← 0; sum ← 0; path.to.v ← false
foreach x ∈ V do
    Nondeterministically guess if d(x) ≤ k
    If guess is Yes then
            Guess a path of length l ≤ k from s to x
            if guess is correct then
                Set count ← count + 1
                Set sum ← sum + l
                if x = v then set path.to.v ← true
            else
                return "?"
            end
    end
end
if count = ck and sum = Σk then
    return path.to.v
else
    return "?"
end
```

Algorithm 2:Computing ck and

```
Input: (G, k, c(k−1) , Σ(k−1) )
Output: c(k) , Σ(k)
Initialize c(k) ← c(k−1) and Σ(k) ← Σ(k−1)
for each v ∈ V do
    if ¬(d(v) ≤ k − 1) then
        for each x such that (x, v) ∈ E do
            if d(x) ≤ k − 1 then
                Set c(k) ← c(k) + 1
                Set   Σ(k) ← Σ(k) + k
            end
        end
    end
end
return c(k) and  Σ(k)
```

Algorithm 3: Determining if there exists a path from s to t in G.

Input: A directed graph G
Output: true if there is a path from s to t, false otherwise
1 Initialize c(0) ← 1, Σ(0) ← 0, k ← 0
2 for k = 1, . . . , n do
3 Compute c(k) and Σ(k) by invoking Algorithm 2 on (G, k, c(k−1) , Σ(k−1))
4 end
5 Invoke Algorithm 1 on (G, t, n, c(n) , Σ(n)) and return its value

Taking log space computable func. to define weights(|G|=n)

- Weights to the edges can be assigned on following criteria:
  - Case 1: North edge

    $w(e)= i+ n^4$, for column i
  - Case 2: South edge

    $w(e)= -i+ n^4$, for column i
  - Case 3: East/West edge

    $w(e)= n^4$
- The weight of any path P can be denoted as: $a + b.(n^4)$

Claim 3

- Let C be a simple directed cycle in G. Then $a(C) = +A(C)$ if C is a counter-clockwise cycle and $a(C) = -A(C)$ if C is a clockwise cycle.
- $A(C)$: denotes the number of unit squares the cycle encloses in grid graph.
- $a(C)$: denotes the ‘a’ component of the path.

Claim 4

- Let G be a grid graph. With respect to the weight function w, for any two vertices u and v, the minimum weight path from u to v, if one exists, is unique.

Thank you.