Concurrent Depth-First Search Algorithms

Mohamad Alsabbagh

Department of Electrical Engineering and Computer Science
York University, Toronto

October 8, 2015

Problems Tarjan’s algorithms solve

Tarjan's Algorithms solve three related problems relevant to model checking. Given a state graph;

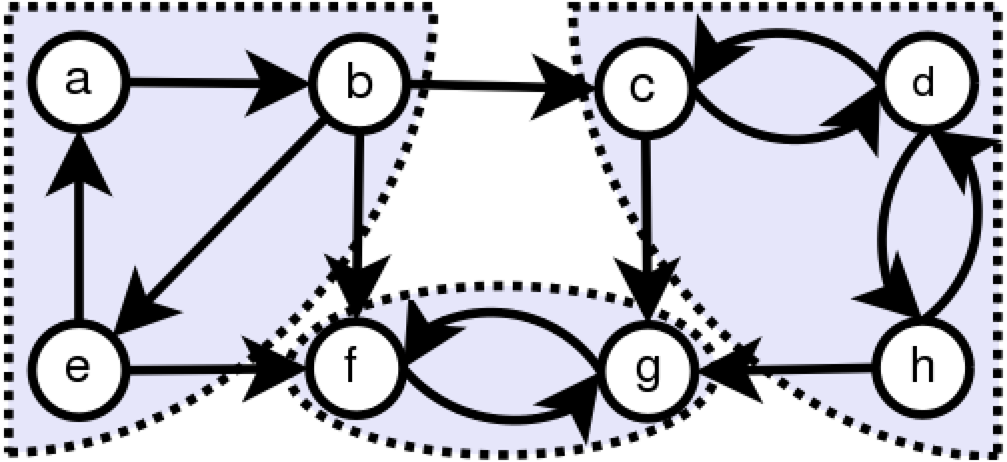
1. Find its Strongly Connected Components (SCCs)
2. Identify which nodes are in a loop
3. Locate which nodes are in a lasso

Why are these problems important?

- Lassos: FDR or failure-divergence refinement.
- SCCs: useful for performing compression on the transition graphs.
- Loops: important in linear temporal logic (LTL) model checking.

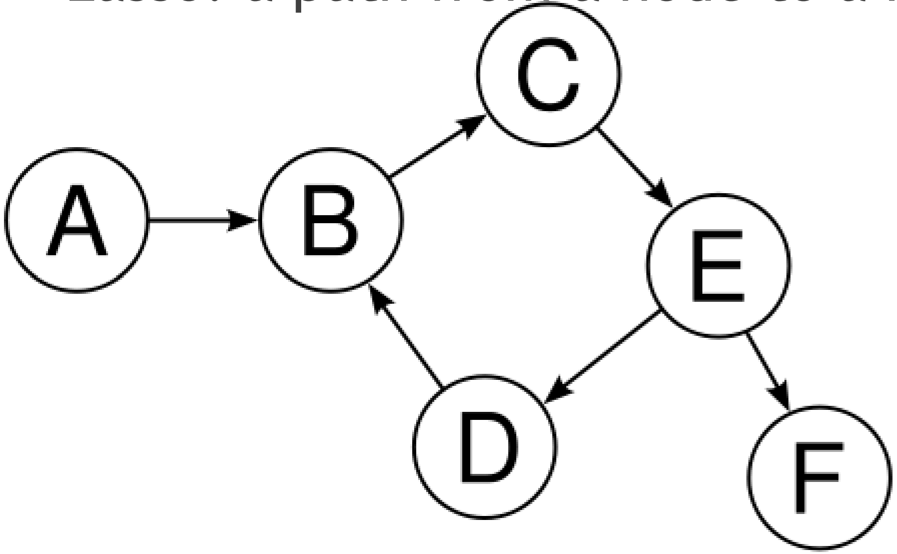
Strongly Connected Components

A directed subgraph that satisfy Strongly Connected attribute.



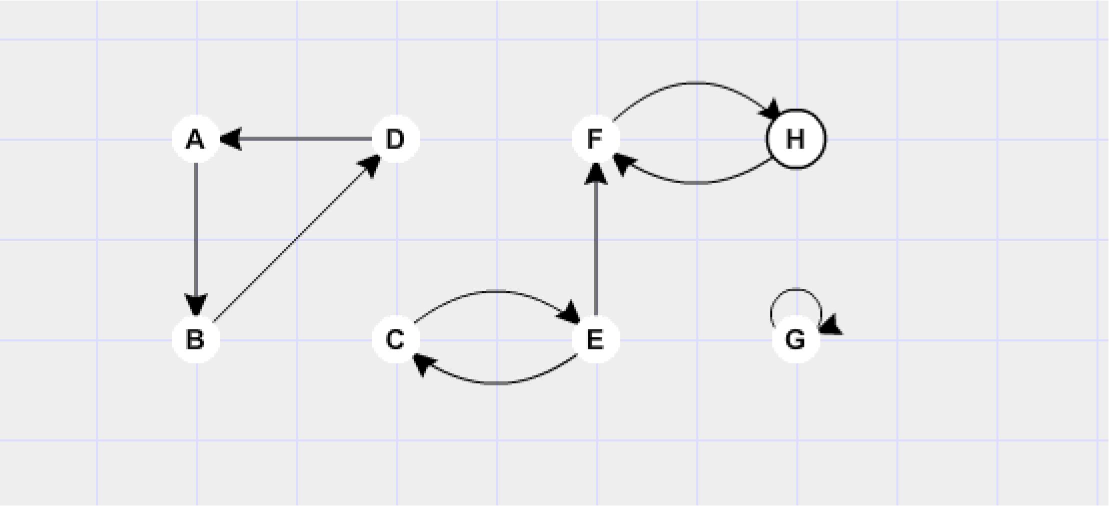
Loops & Lassos

Loop: a node is part of a direct cycle
Lasso: a path from a node to a node on a cycle



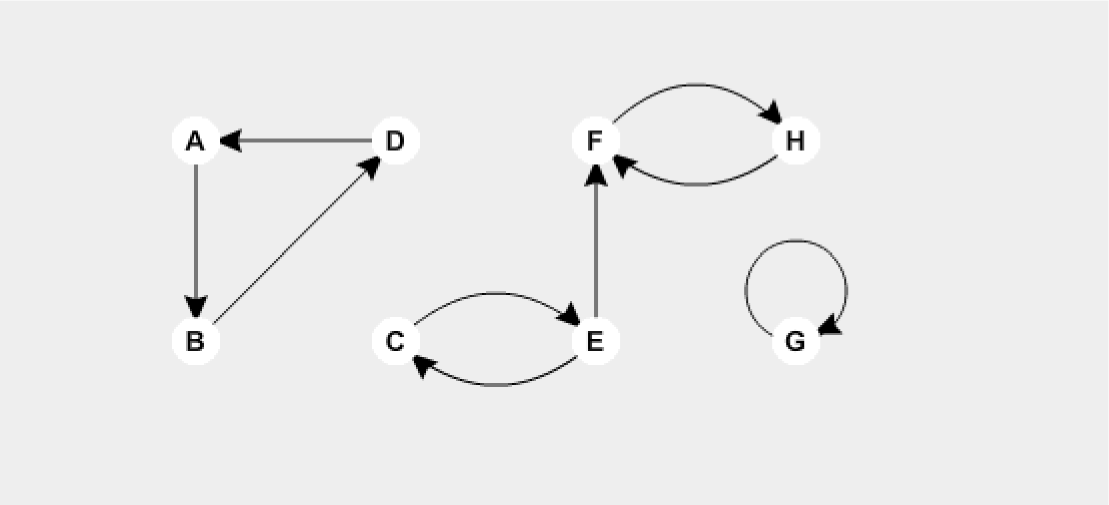
Sequential Tarjan's Algorithm

Depth-First Search to identify SCCs.



Concurrent Tarjan's Algorithm

A single concurrent version of Tarjan’s algorithm to identify SCCs

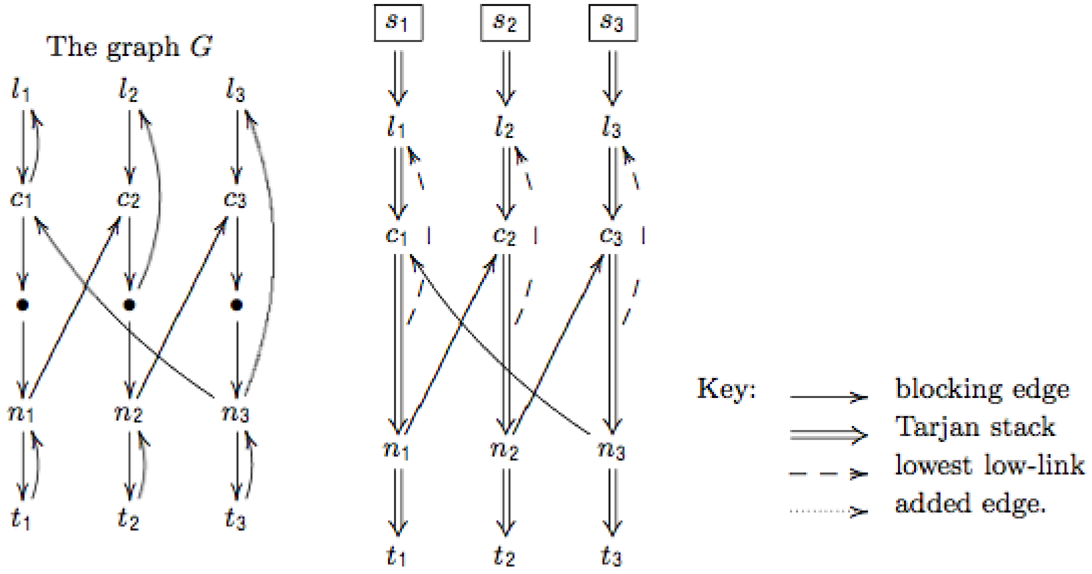


Tarjan's Node Structure

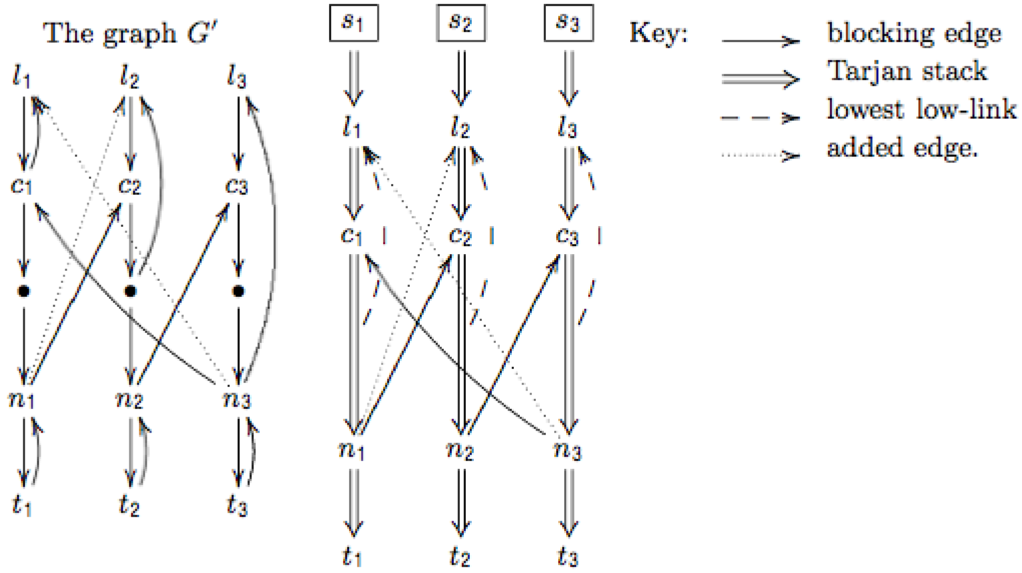
Each node in the graph G has the following attributes:

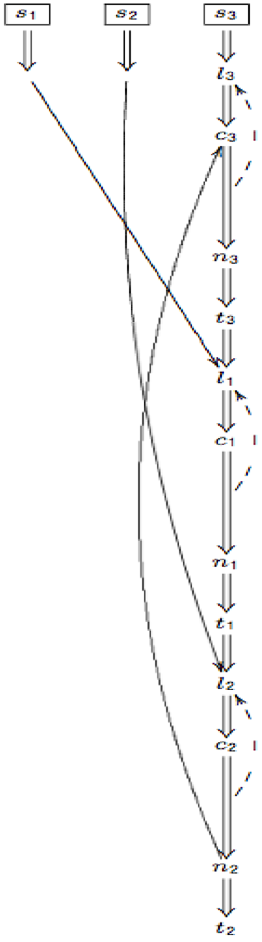
- index (sequential and concurrent):
  - which is a sequence counter, corresponding to the order in which nodes were encountered
- lowlink (sequential and concurrent):
  - which records the smallest index of a node n′ in the stack that is reachable via the descendents of n fully considered so far
- search (concurrent):
  - identifying which search a node belongs to

Circular Dependency



Circular Dependency Node Transfer





Circular Dependency Node Transfer

When transferring a node from s_1 to s_3 we will need to recalculate its index amd lowlink values:

- $\delta_1 = (s_3.\text{index} - l_1.\text{index})$
- we add δ_1 onto the index and lowlink of each node transferred from s_1 and update.

Next Steps

Plan:

- implement all three algorithms
- compare their performance

Q&A

Thanks