

Data Structures and Algorithms

Lecture 37

Aniket Basu Roy

2026-04-27 Mon

Contents

1 Agenda

1.1 Directed Graphs and the Reachability Relation

- The reachability relation: reflexive and transitive
- When is reachability symmetric? Strongly Connected Components
- When is reachability antisymmetric? Directed Acyclic Graphs
- Partial orders and topological orderings
- Claim: G is a DAG if and only if G has a topological ordering
- Kahn's algorithm

2 The Reachability Relation

Given a directed graph $G = (V, E)$, define the binary relation $\text{reach} \subseteq V \times V$ by:

$$\text{reach}(u, v) \iff \text{there exists a directed path from } u \text{ to } v \text{ in } G.$$

(The trivial path of length zero from u to u counts.)

2.1 Properties

- **Reflexive:** $\text{reach}(u, u)$ for every $u \in V$. ✓
- **Transitive:** $\text{reach}(u, v)$ and $\text{reach}(v, w)$ imply $\text{reach}(u, w)$ (concatenate the two paths). ✓

So reach is a **preorder** on V .

Two further properties a relation can have: symmetry and antisymmetry. Asking which graphs make reach symmetric or antisymmetric leads to two fundamental graph classes.

3 Symmetry: Strongly Connected Components

reach is symmetric if $\text{reach}(u, v) \Rightarrow \text{reach}(v, u)$ for all u, v . This holds globally only when G is strongly connected (every vertex can reach every other).

In general, define the relation $u \sim v$ on V by:

$$u \sim v \iff \text{reach}(u, v) \text{ and } \text{reach}(v, u).$$

3.1 \sim is an Equivalence Relation

- **Reflexive:** $u \sim u$ (trivial path). ✓
- **Symmetric:** $u \sim v \Rightarrow v \sim u$ (by definition). ✓
- **Transitive:** $u \sim v$ and $v \sim w$ imply $u \sim w$ (by transitivity of reach). ✓

3.2 Strongly Connected Components (SCCs)

The equivalence classes of \sim are the **strongly connected components** of G .

Within each SCC every vertex can reach every other; no vertex outside the SCC is in the same class. Collapsing each SCC to a single supernode yields the **condensation** of G , which is a DAG. (We will study efficient SCC algorithms in Lecture 38.)

4 Antisymmetry: Directed Acyclic Graphs

reach is antisymmetric if $\text{reach}(u, v)$ and $\text{reach}(v, u)$ together imply $u = v$.

This means G contains no directed cycle of length ≥ 1 ; such a graph is a **Directed Acyclic Graph (DAG)**.

On a DAG, reach is reflexive, transitive, and antisymmetric: a **partial order** on V .

4.1 Partial Orders and Linear Extensions

A partial order need not compare every pair of elements. A **linear extension** of a partial order is a total ordering v_1, v_2, \dots, v_n of V such that whenever $\text{reach}(v_i, v_j)$ and $i \neq j$, we have $i < j$.

In graph terms this is exactly a **topological ordering**.

4.2 Definition: Topological Ordering

A **topological ordering** of a directed graph G is a linear ordering of all vertices such that for every edge $(u, v) \in E$, u appears before v in the ordering.

5 Claim: G is a DAG iff G has a Topological Ordering

5.1 Direction 1: Topological ordering $\Rightarrow G$ is a DAG

Suppose G has a topological ordering v_1, v_2, \dots, v_n .

Suppose for contradiction that G has a directed cycle $v_{i_1} \rightarrow v_{i_2} \rightarrow \dots \rightarrow v_{i_k} \rightarrow v_{i_1}$.

Each edge $v_{i_j} \rightarrow v_{i_{j+1}}$ requires v_{i_j} to appear before $v_{i_{j+1}}$, giving:

$$i_1 < i_2 < \dots < i_k < i_1.$$

Contradiction. So G is a DAG. ■

5.2 Direction 2: G is a DAG $\Rightarrow G$ has a topological ordering

We need the following lemma.

5.2.1 Lemma: Every DAG has a source vertex

Every DAG $G = (V, E)$ contains at least one vertex with in-degree zero (a **source**).

Proof. Start at any vertex and repeatedly follow a predecessor. If the walk ever revisits a vertex it creates a cycle, contradicting the DAG assumption. Since $|V|$ is finite the walk must terminate at a vertex with no predecessor: a source. ■

5.2.2 Proof by induction on $|V|$

Base case: $|V| = 1$: the single vertex is trivially a topological ordering. ✓

Inductive step: Assume every DAG with fewer than n vertices has a topological ordering.

Let G be a DAG on n vertices. By the Lemma, G has a source u .

Let $G' = G \setminus \{u\}$ (delete u and all edges leaving u). G' is a DAG (subgraph of a DAG) on $n - 1$ vertices.

By the inductive hypothesis G' has a topological ordering v_1, v_2, \dots, v_{n-1} .

Then $u, v_1, v_2, \dots, v_{n-1}$ is a topological ordering of G : every edge from u leads to some v_i , and u appears first. ■

6 Kahn's Algorithm

The inductive proof gives an algorithm: repeatedly identify and output a source.

KAHN(G)

```
Input: directed graph  $G = (V, E)$ 
Output: topological ordering of  $V$ , or "cycle detected"
1. compute in-degree[ $v$ ] for all  $v$  in  $V$ 
2.  $S = \{v \text{ in } V : \text{in-degree}[v] == 0\}$  // queue/set of sources
3. order = empty list
4. while  $S$  is not empty
5.      $u =$  any vertex removed from  $S$ 
6.     append  $u$  to order
7.     for each edge  $(u, v)$  in  $E$ 
8.         in-degree[ $v$ ] = in-degree[ $v$ ] - 1
9.         if in-degree[ $v$ ] == 0
10.            add  $v$  to  $S$ 
11. if |order| ==  $|V|$ 
12.     return order
13. else
14.     return "cycle detected"
```

6.1 Correctness

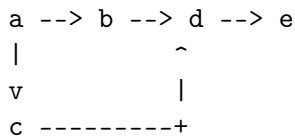
- **G is a DAG:** a source always exists by the Lemma; the algorithm outputs all $|V|$ vertices in topological order. ✓
- **G has a cycle:** the vertices on the cycle never reach in-degree zero, so $|\text{order}| < |V|$ and the cycle is detected. ✓

6.2 Time Complexity

Step	Cost
Compute in-degrees	$O(V + E)$
Each vertex added/removed from S	$O(V)$ total
Each edge decremented once	$O(E)$ total

$$T_{\text{Kahn}} = O(V + E).$$

6.3 Example



Initial in-degrees: $a = 0, b = 1, c = 1, d = 2, e = 1$.

Step	Extracted	in-degree updates	Sources after step
1	a	$b : 0, c : 0$	$\{b, c\}$
2	b	$d : 1$	$\{c\}$
3	c	$d : 0$	$\{d\}$
4	d	$e : 0$	$\{e\}$
5	e	—	$\{\}$

Topological order: a, b, c, d, e . (Also valid: a, c, b, d, e .)

7 Questions

- Does every partial order on a finite set have a linear extension? Prove it or give a counterexample.
- What happens in Kahn's algorithm when multiple sources exist at a step? Are all resulting orderings valid?

- Show that the condensation (SCC DAG) of any directed graph is acyclic.