



# CSC 384

## Introduction to Artificial Intelligence

### CSP 3

Alice Gao and Randy Hickey  
Winter 2023

# Learning Goals

---

By the end of this lecture, you should be able to

- Trace the execution of the AC-3 algorithm.
- Trace the execution of Backtracking Search and AC-3.
- Explain why we must add arcs back to the queue in the AC-3 algorithm.
- Explain the properties of the AC-3 algorithm (order of removing arcs, three outcomes of the algorithm, guaranteed to terminate).

# Outline

---

1. [The AC-3 Algorithm](#)
2. [Properties of the AC-3 Algorithm](#)
3. [Backtracking Search with AC-3](#)

---

# THE AC-3 ALGORITHM

# Types of Constraints

---

- **Unary** constraints (over 1 variable)

Examples:  $c(x): x = 2$ ;  $c(y): y > 5$

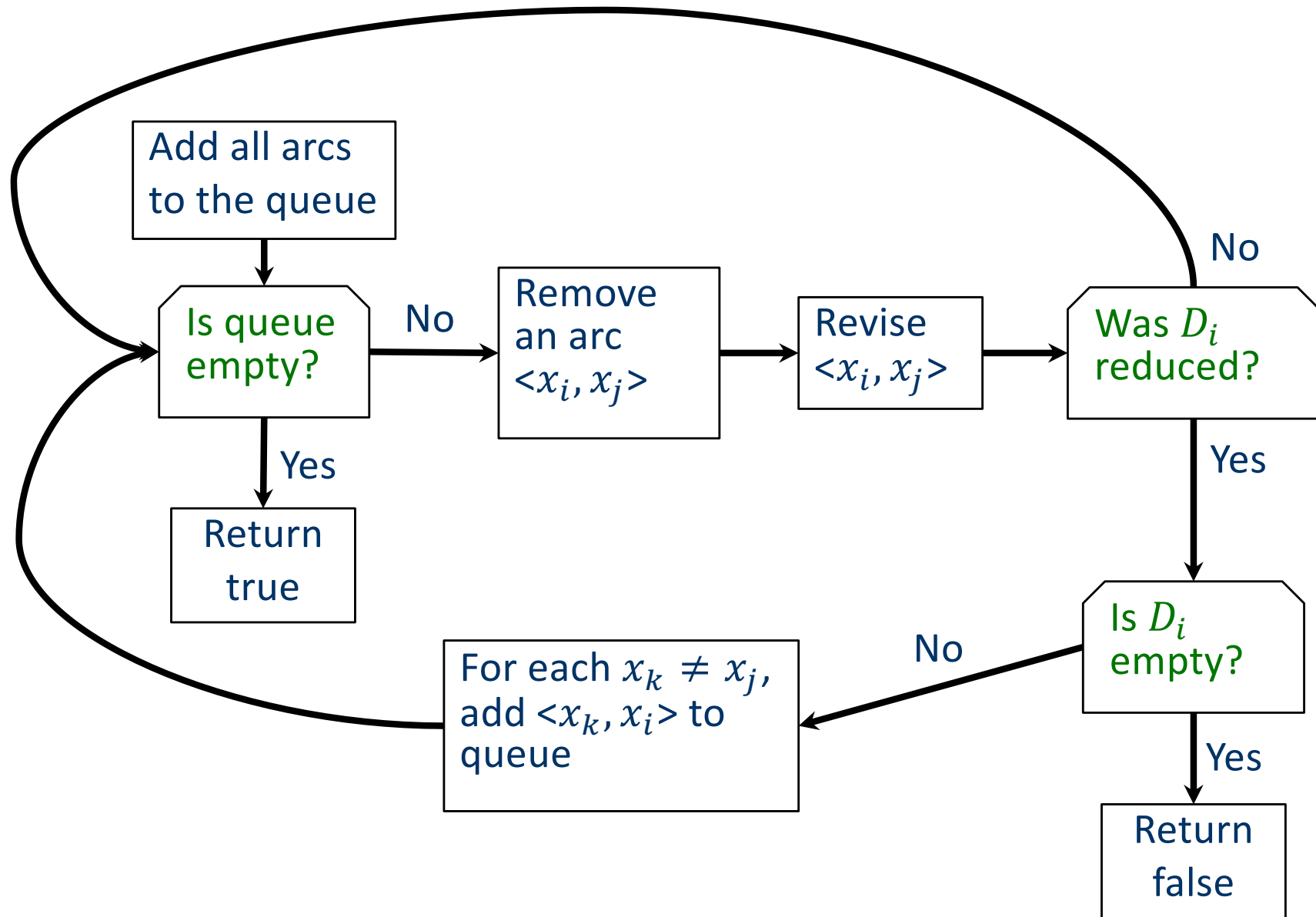
- **Binary** constraints (over 2 variables)

Examples:  $c(x, y): x + y < 6$

- **Higher-order** ( $n$ -ary) constraints (over  $\geq 3$  variables)

- Possible to convert any higher-order constraint to a binary constraint by defining “dummy” variables.

# AC-3 Arc-Consistency Algorithm (Flowchart)

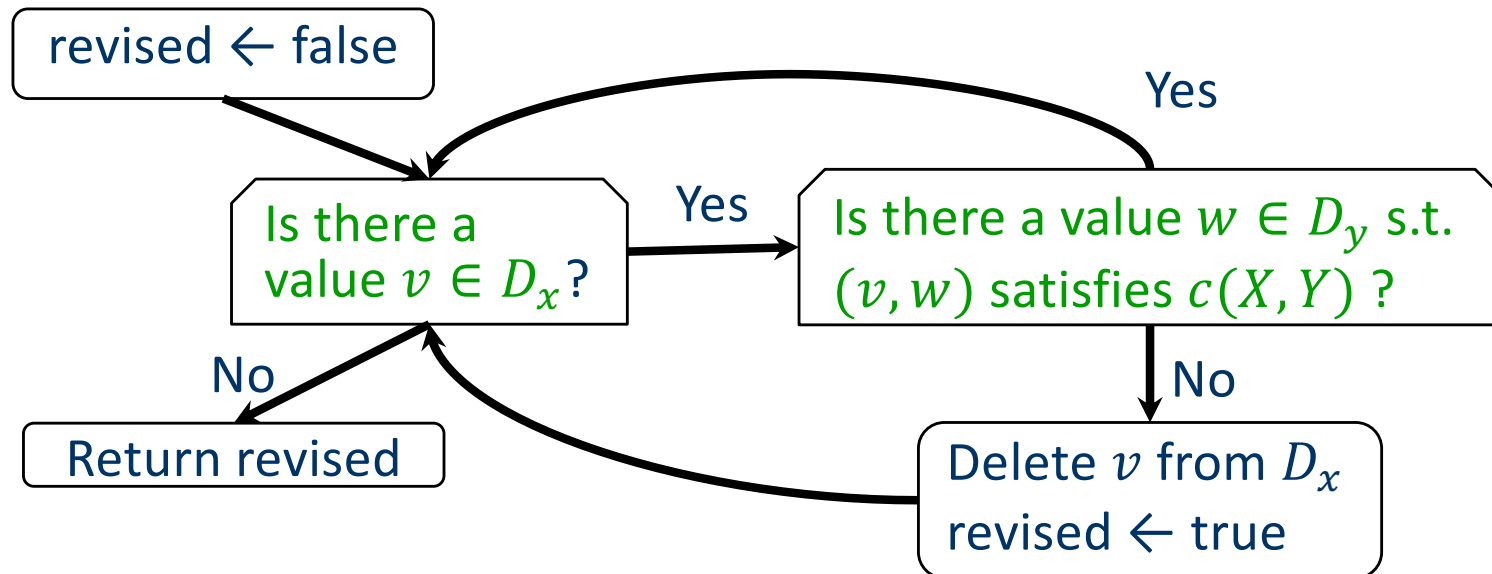


# AC-3 Arc-Consistency Algorithm (Pseudocode)

---

```
1. function AC-3(csp)
2.   # returns false if an inconsistency is found and true
   otherwise
3.   inputs: csp, a binary CSP with components (X, D, C)
4.   local variables: queue, a queue of arcs, initially all
   the arcs in csp
5.
6.   add all the arcs to the queue
7.   while queue is not empty do
8.      $\langle X_i, X_j \rangle \leftarrow \text{REMOVE-FIRST}(\text{queue})$ 
9.     if REVISE(csp,  $X_i$ ,  $X_j$ ) then
10.      if  $D_i$  is empty then
11.        return false
12.      for each  $X_k$  in  $X_i.\text{NEIGHBORS} - \{X_j\}$  do
13.        add  $\langle X_k, X_i \rangle$  to queue
14.   return true
```

# Revise Domain to Restore Arc-Consistency










1. function REVISE(csp,  $X$ ,  $Y$ )
2.     revised  $\leftarrow$  false
3.     for each  $v$  in  $D_x$  do
4.         if no value  $w$  in  $D_y$  allows  $(v, w)$  to satisfy the constraint between  $X$  and  $Y$  then
5.             delete  $v$  from  $D_x$
6.             revised  $\leftarrow$  true
7.     return revised

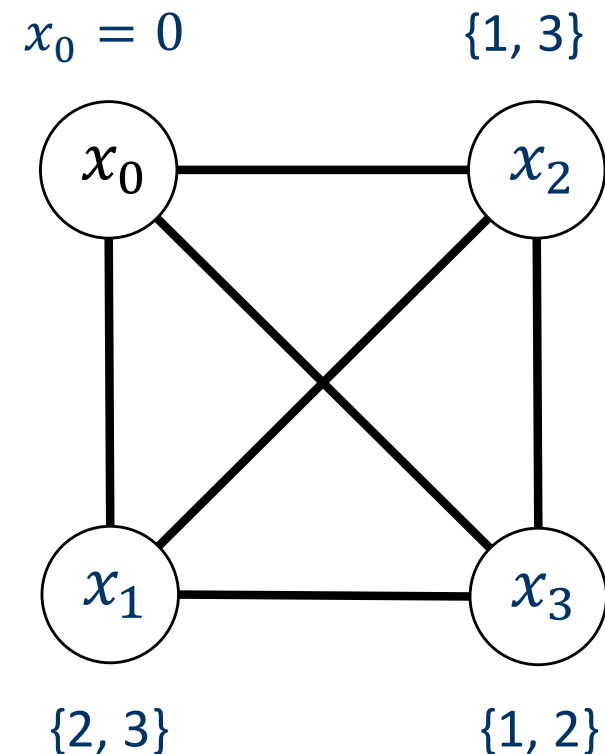


# Example 1: Execute AC-3 Algorithm

Queue:  $\langle x_1, x_2 \rangle, \langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_2, x_3 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_2 \rangle$

Step 1	
Arc removed	
Values deleted	
Arc(s) added	




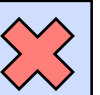


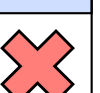
	0	1	2	3
0				
1				
2				
3				

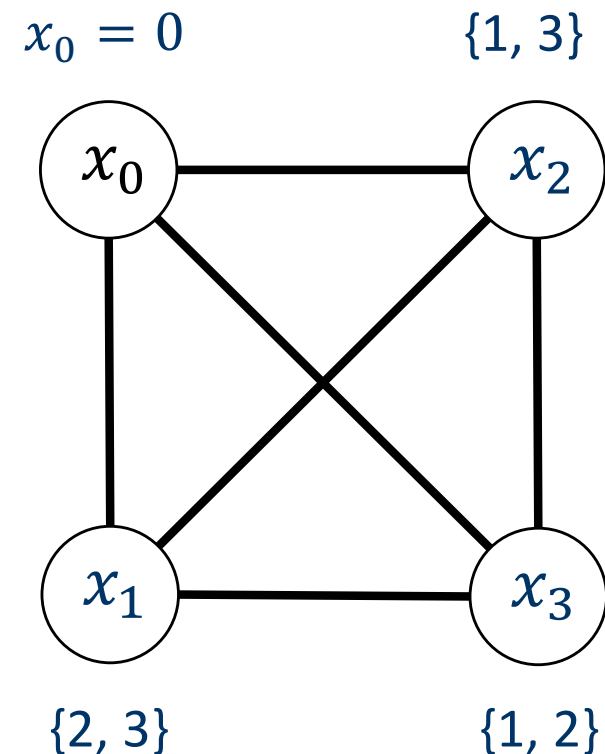


# Example 1: Execute AC-3 Algorithm

Queue:  $\langle x_1, x_2 \rangle$ ,  $\langle x_2, x_1 \rangle$ ,  $\langle x_3, x_1 \rangle$ ,  $\langle x_2, x_3 \rangle$ ,  $\langle x_1, x_3 \rangle$ ,  $\langle x_3, x_2 \rangle$

Step 1	
Arc removed	$\langle x_1, x_2 \rangle$
Values deleted	
Arc(s) added	

	0	1	2	3
0				
1				
2				
3				










# Step 1: Restore Arc Consistency

How can we make the arc below consistent?

$$\langle x_1, x_2 \rangle \text{ (or } \langle x_1, c(x_1, x_2) \rangle)$$

- A. It is already consistent.
- B. Remove 2 from  $D_1$ .
- C. Remove 3 from  $D_1$ .
- D. Remove 2 and 3 from  $D_1$ .
- E. Remove 1 from  $D_2$ .
- F. Remove 3 from  $D_2$ .
- G. Remove 1 and 3 from  $D_2$ .








	0	1	2	3
0				
1				
2				
3				

# Step 1: Restore Arc Consistency

How can we make the arc below consistent?

$$\langle x_1, x_2 \rangle \text{ (or } \langle x_1, c(x_1, x_2) \rangle)$$

- A. It is already consistent.
- B. Remove 2 from  $D_1$ .
- C. Remove 3 from  $D_1$ .
- D. Remove 2 and 3 from  $D_1$ .
- E. Remove 1 from  $D_2$ .
- F. Remove 3 from  $D_2$ .
- G. Remove 1 and 3 from  $D_2$ .




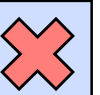



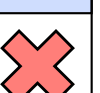
	0	1	2	3
0				
1				
2				
3				

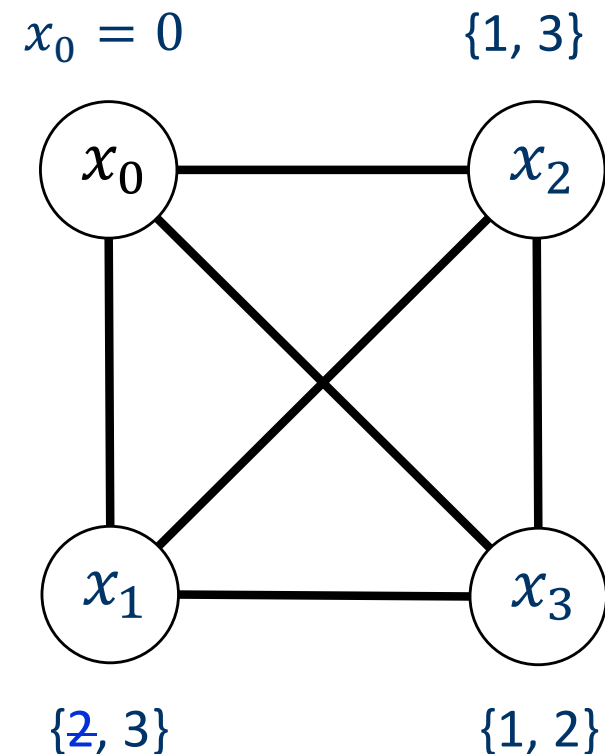
Answers: B

# Example 1: Execute AC-3 Algorithm

Queue:  $\langle \cancel{x_1}, \cancel{x_2} \rangle, \langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_2, x_3 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_2 \rangle$

Step 1	
Arc removed	$\langle x_1, x_2 \rangle$
Values deleted	Remove 2 from $D_1$ .
Arc(s) added	

	0	1	2	3
0				
1				
2				
3				

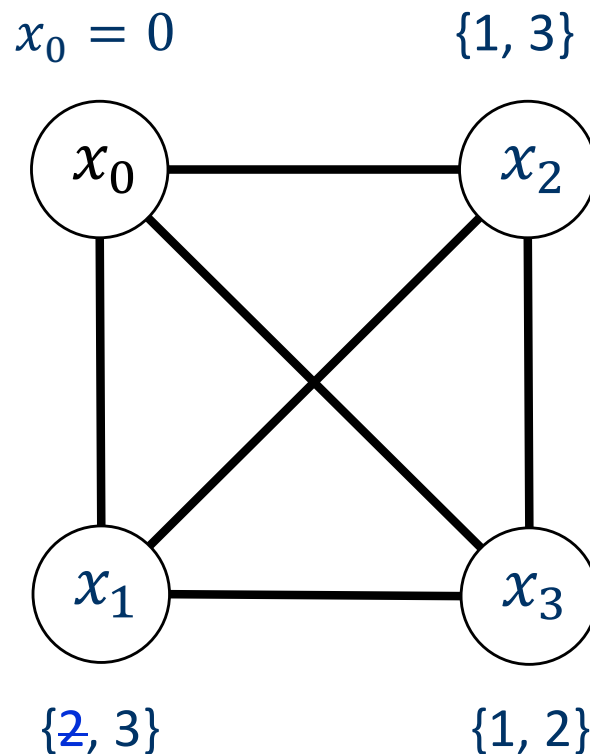


## Step 1: Add Arcs to Queue

---

After reducing  $D_1$  to restore consistency for  $\langle x_1, x_2 \rangle$ , which arc(s) should we add to the queue?

- A.  $\langle x_0, x_1 \rangle$
- B.  $\langle x_2, x_1 \rangle$
- C.  $\langle x_3, x_1 \rangle$
- D.  $\langle x_1, x_0 \rangle$
- E.  $\langle x_1, x_2 \rangle$
- F.  $\langle x_1, x_3 \rangle$



## Step 1: Add Arcs to Queue

After reducing  $D_1$  to restore consistency for  $\langle x_1, x_2 \rangle$ , which arc(s) should we add to the queue?

A.  $\langle x_0, x_1 \rangle$

B.  $\langle x_2, x_1 \rangle$

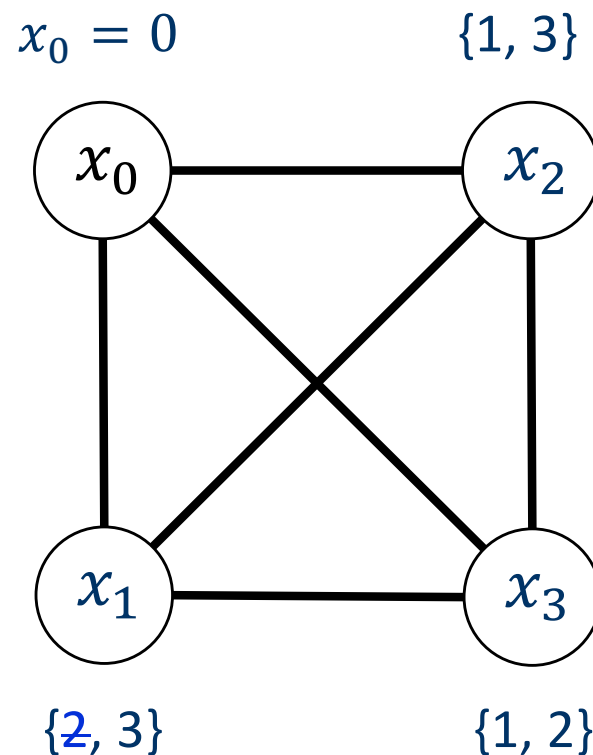
C.  $\langle x_3, x_1 \rangle$

D.  $\langle x_1, x_0 \rangle$

E.  $\langle x_1, x_2 \rangle$

F.  $\langle x_1, x_3 \rangle$




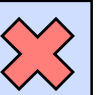



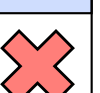
Answers: C

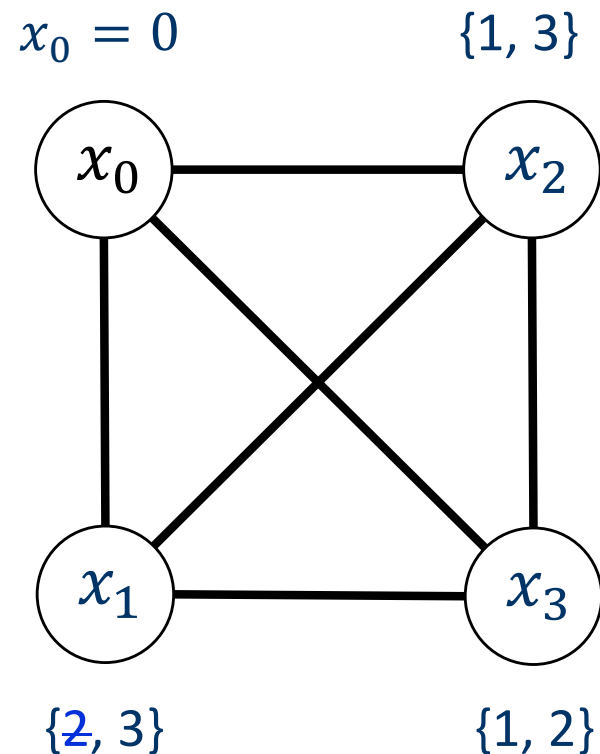


# Example 1: Execute AC-3 Algorithm

Queue:  $\langle x_1, x_2 \rangle$ ,  $\langle x_2, x_1 \rangle$ ,  $\langle x_3, x_1 \rangle$ ,  $\langle x_2, x_3 \rangle$ ,  $\langle x_1, x_3 \rangle$ ,  $\langle x_3, x_2 \rangle$

Step 1	
Arc removed	$\langle x_1, x_2 \rangle$
Values deleted	Remove 2 from $D_1$ .
Arc(s) added	None. ( $\langle x_3, x_1 \rangle$ already in the queue)

	0	1	2	3
0				
1				
2				
3				




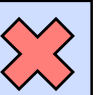



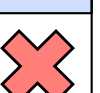


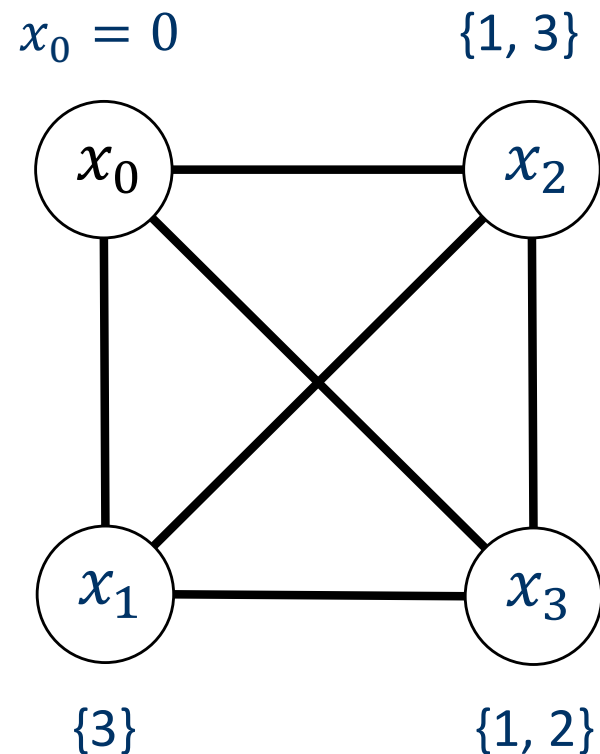


# Example 1: Execute AC-3 Algorithm

Queue:  $\langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_2, x_3 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_2 \rangle$

Step 2	
Arc removed	
Values deleted	
Arc(s) added	




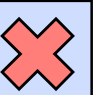



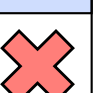
	0	1	2	3
0				
1				
2				
3				

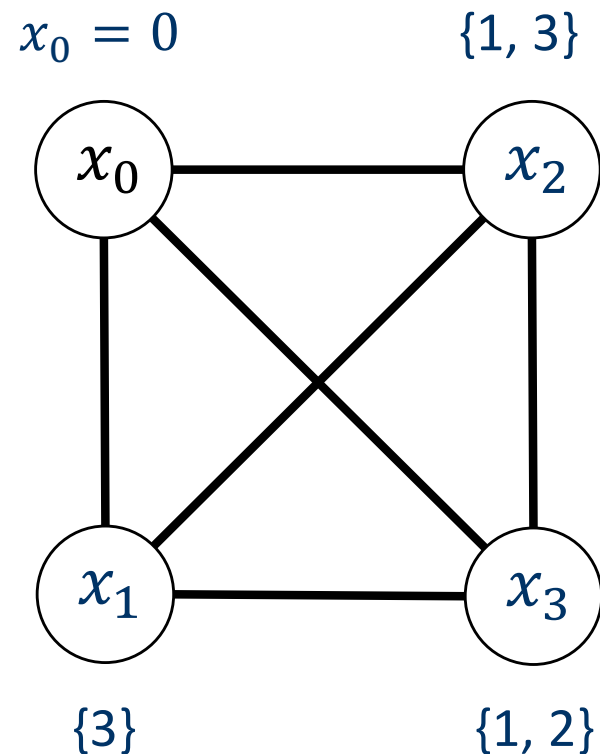


# Example 1: Execute AC-3 Algorithm

Queue:  $\langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_2, x_3 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_2 \rangle$

Step 2	
Arc removed	$\langle x_2, x_1 \rangle$
Values deleted	
Arc(s) added	




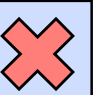




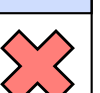
	0	1	2	3
0				
1				
2				
3				

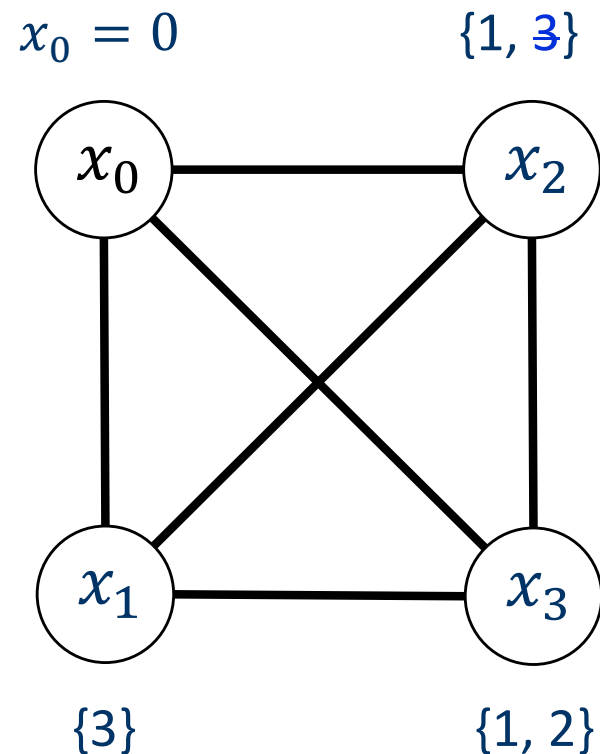


# Example 1: Execute AC-3 Algorithm

Queue:  $\langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_2, x_3 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_2 \rangle$

Step 2	
Arc removed	$\langle x_2, x_1 \rangle$
Values deleted	Delete 3 from $D_2$ .
Arc(s) added	




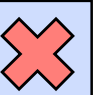




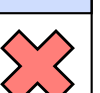
	0	1	2	3
0				
1				
2				
3				

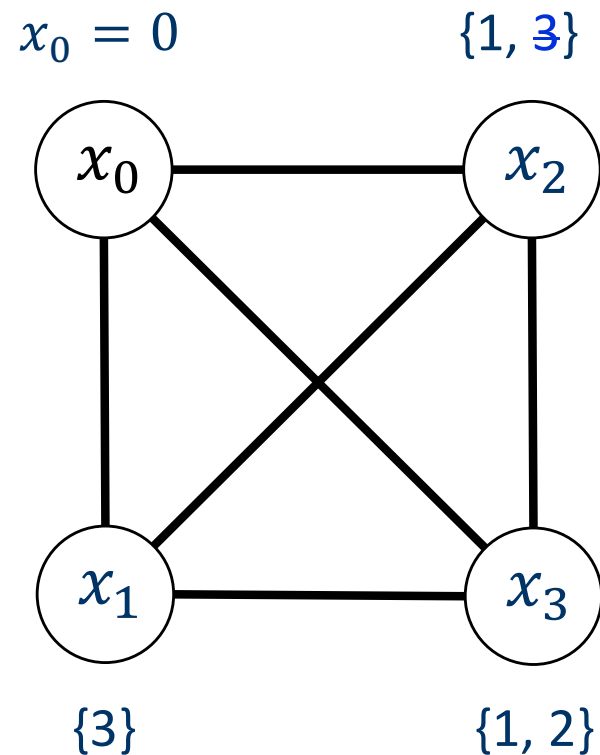


# Example 1: Execute AC-3 Algorithm

Queue:  $\langle x_2, x_1 \rangle$ ,  $\langle x_3, x_1 \rangle$ ,  $\langle x_2, x_3 \rangle$ ,  $\langle x_1, x_3 \rangle$ ,  $\langle x_3, x_2 \rangle$

Step 2	
Arc removed	$\langle x_2, x_1 \rangle$
Values deleted	Delete 3 from $D_2$ .
Arc(s) added	None. ( $\langle x_3, x_2 \rangle$ already in the queue)




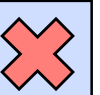




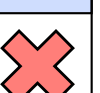
	0	1	2	3
0				
1				
2				
3				

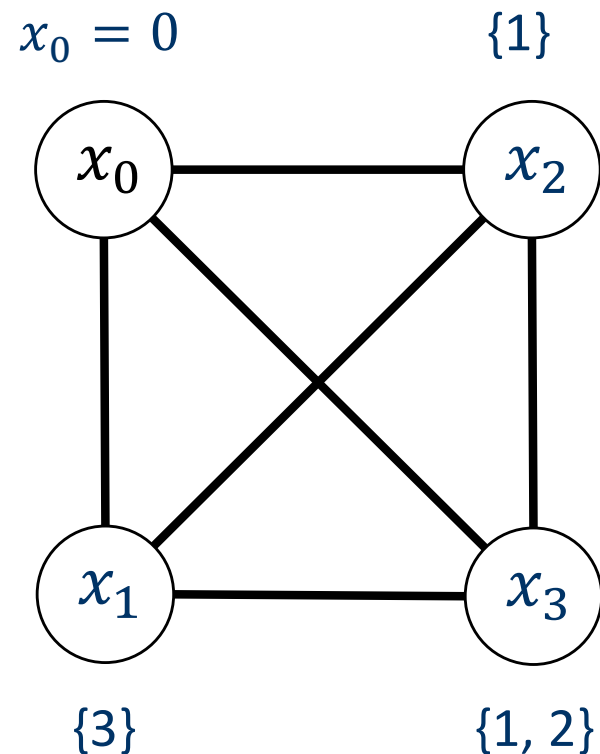


# Example 1: Execute AC-3 Algorithm

Queue:  $\langle x_3, x_1 \rangle, \langle x_2, x_3 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_2 \rangle$

Step 3	
Arc removed	
Values deleted	
Arc(s) added	




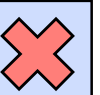





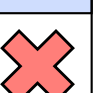
	0	1	2	3
0				
1				
2				
3				

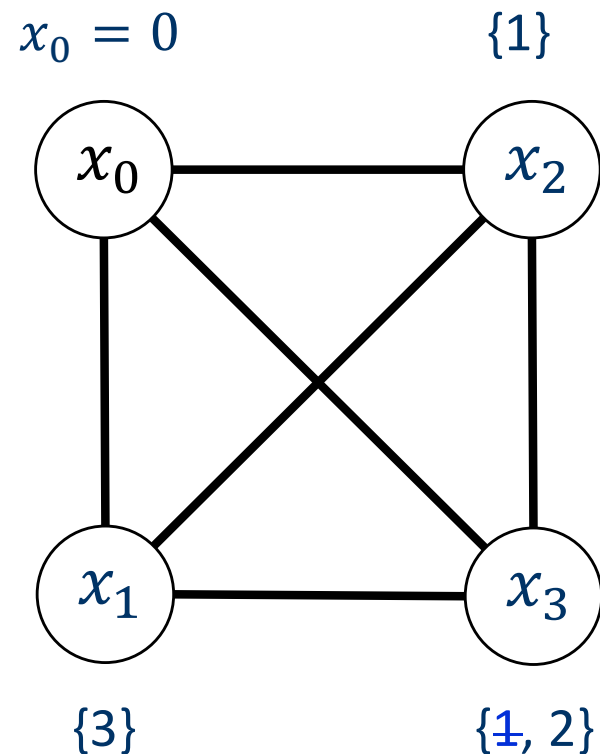


# Example 1: Execute AC-3 Algorithm

Queue:  $\langle \cancel{x_3}, \cancel{x_1} \rangle, \langle x_2, x_3 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_2 \rangle$

Step 3	
Arc removed	$\langle x_3, x_1 \rangle$
Values deleted	Remove 1 from $D_3$ .
Arc(s) added	None. ( $\langle x_2, x_3 \rangle$ already in the queue.)




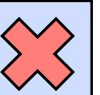





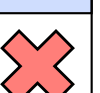
	0	1	2	3
0				
1				
2				
3				

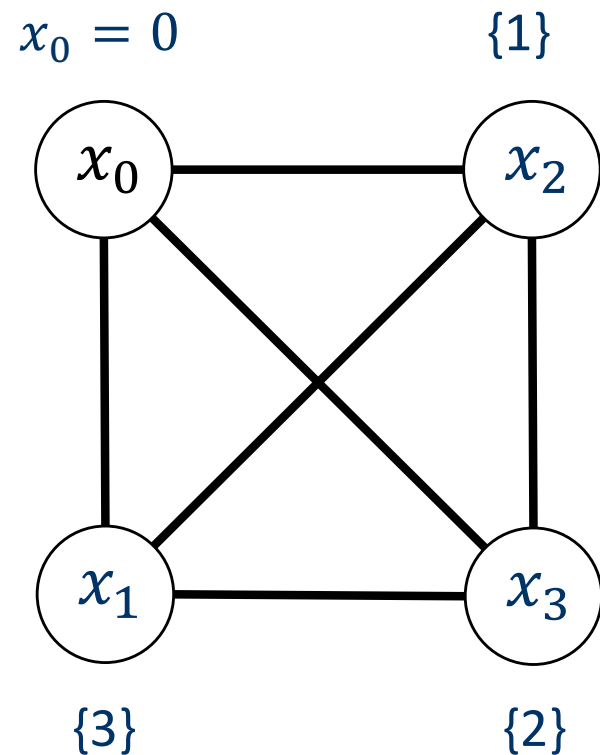


# Example 1: Execute AC-3 Algorithm

Queue:  $\langle x_2, x_3 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_2 \rangle$

Step 4	
Arc removed	
Values deleted	
Arc(s) added	




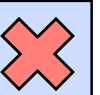





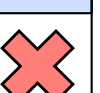
	0	1	2	3
0				
1				
2				
3				

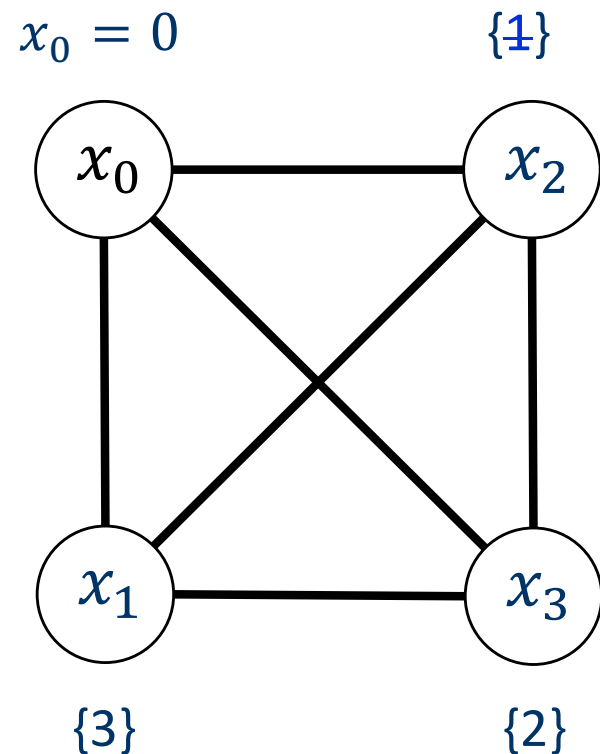


# Example 1: Execute AC-3 Algorithm

Queue:  $\langle \cancel{x_2}, x_3 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_2 \rangle$

Step 4	
Arc removed	$\langle x_2, x_3 \rangle$
Values deleted	Remove 1 from $D_2$ .
Arc(s) added	(Return false before reaching this step)

	0	1	2	3
0				
1				
2				
3				





# Example 1: Execute AC-3 Algorithm

---

Queue:  $\langle x_1, x_2 \rangle, \langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_2, x_3 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_2 \rangle$   
 $x_0 = 0, D_1 = \{2, 3\}, D_2 = \{1, 3\}, D_3 = \{1, 2\}$

Step	Arc removed	Values to delete	Updated Domains	Arc(s) added

# Answer 1: Execute AC-3 Algorithm

---

Queue:  $\langle x_1, x_2 \rangle, \langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_2, x_3 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_2 \rangle$   
 $x_0 = 0, D_1 = \{2, 3\}, D_2 = \{1, 3\}, D_3 = \{1, 2\}$

Step	Arc removed	Values to delete	Updated Domains	Arc(s) added
1	$\langle x_1, x_2 \rangle$	Remove 2 from $D_1$	$D_1 = \{3\}, D_2 = \{1, 3\}, D_3 = \{1, 2\}$	None
2	$\langle x_2, x_1 \rangle$	Remove 3 from $D_2$	$D_1 = \{3\}, D_2 = \{1\}, D_3 = \{1, 2\}$	None
3	$\langle x_3, x_1 \rangle$	Remove 1 from $D_3$	$D_1 = \{3\}, D_2 = \{1\}, D_3 = \{2\}$	None
4	$\langle x_2, x_3 \rangle$	Remove 1 from $D_2$	$D_1 = \{3\}, D_2 = \{\}, D_3 = \{2\}$	
5				

$D_2$  is empty. Return **false**.

# Why Do We Need to Add Arcs back to Queue?

---

Removing a value from a variable's domain may cause other arcs to become inconsistent.

After reducing  $X_i$ 's domain  
to restore consistency for  $\langle X_i, X_j \rangle$ ,








Add any arc  $\langle X_k, X_i \rangle$  to the queue where

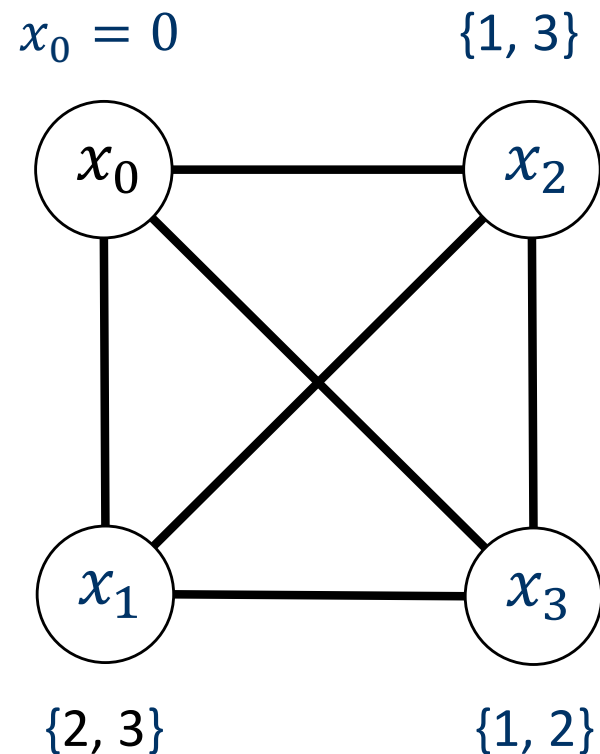
- $X_k$  is a neighbour of  $X_i$ , and
- $X_k \neq X_j$ .

## Example 2: Execute AC-3 Algorithm

Queue:  $\langle x_1, x_3 \rangle, \langle x_2, x_3 \rangle, \langle x_3, x_2 \rangle, \langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_1, x_2 \rangle$

Step 1	
Arc removed	
Values deleted	
Arc(s) added	




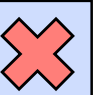


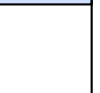
	0	1	2	3
0				
1				
2				
3				

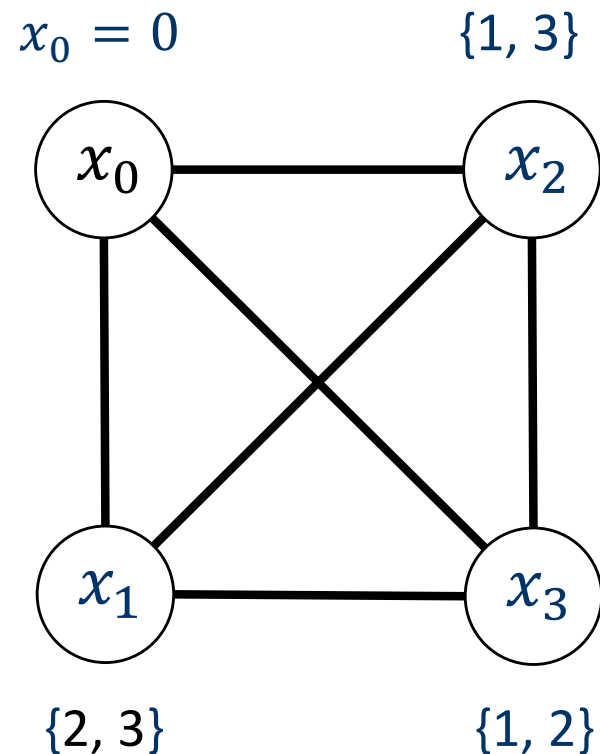


## Example 2: Execute AC-3 Algorithm

Queue:  $\langle x_1, x_3 \rangle, \langle x_2, x_3 \rangle, \langle x_3, x_2 \rangle, \langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_1, x_2 \rangle$

Step 1	
Arc removed	$\langle x_1, x_3 \rangle$
Values deleted	None
Arc(s) added	None. (domain was not reduced.)








	0	1	2	3
0				
1				
2				
3				

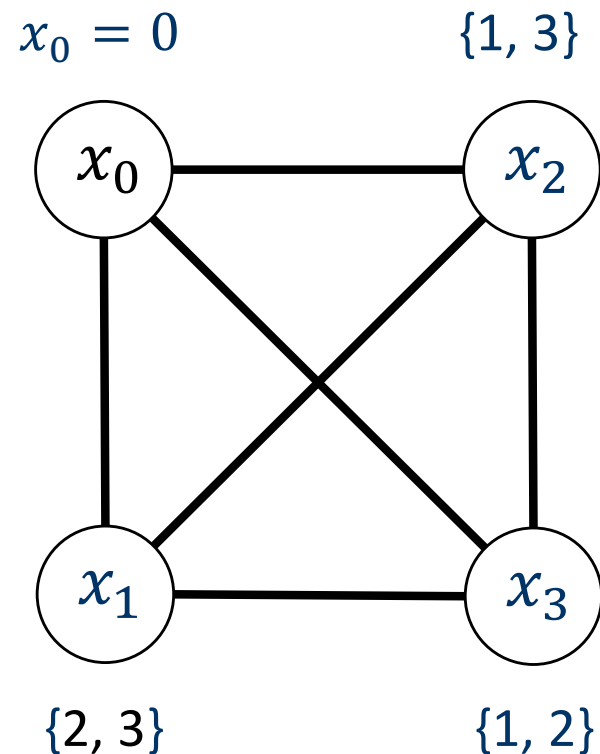


## Example 2: Execute AC-3 Algorithm

Queue:  $\langle x_2, x_3 \rangle, \langle x_3, x_2 \rangle, \langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_1, x_2 \rangle$

Step 2	
Arc removed	
Values deleted	
Arc(s) added	




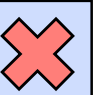



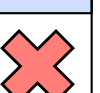
	0	1	2	3
0				
1				
2				
3				

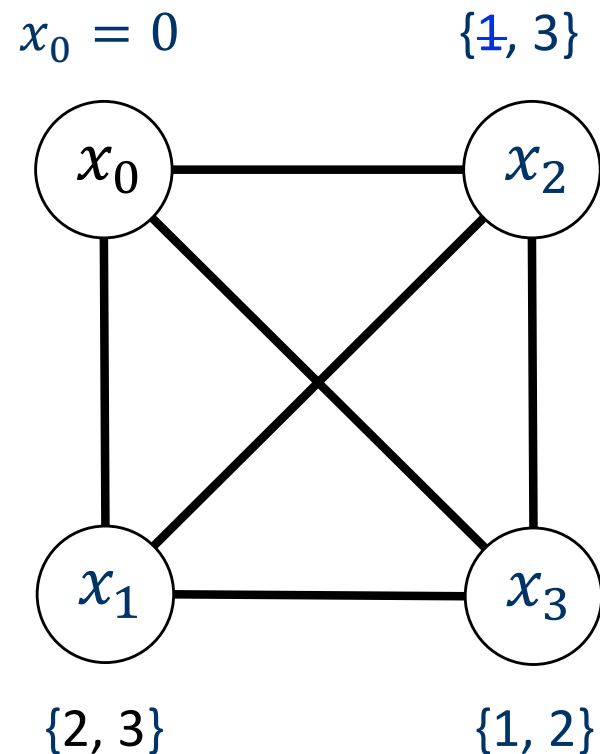


## Example 2: Execute AC-3 Algorithm

Queue:  $\langle \cancel{x_2}, x_3 \rangle, \langle x_3, x_2 \rangle, \langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_1, x_2 \rangle$

Step 2	
Arc removed	$\langle x_2, x_3 \rangle$
Values deleted	Remove 1 from $D_2$ .
Arc(s) added	None. ( $\langle x_1, x_2 \rangle$ already in the queue.)









	0	1	2	3
0				
1				
2				
3				

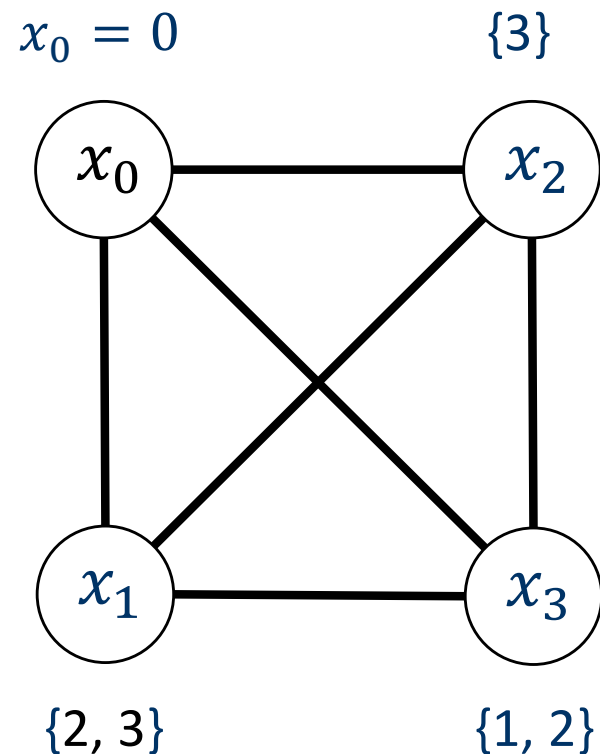


## Example 2: Execute AC-3 Algorithm

Queue:  $\langle x_3, x_2 \rangle, \langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_1, x_2 \rangle$

Step 3	
Arc removed	
Values deleted	
Arc(s) added	

	0	1	2	3
0				
1				
2				
3				




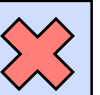



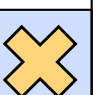
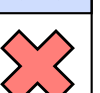


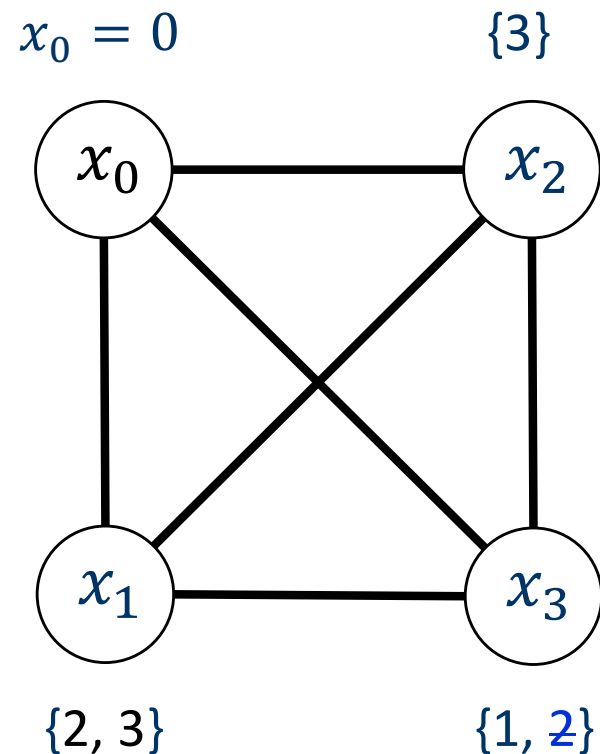


## Example 2: Execute AC-3 Algorithm

Queue:  $\langle \cancel{x_3}, \cancel{x_2} \rangle, \langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_1, x_2 \rangle$

Step 3	
Arc removed	$\langle x_3, x_2 \rangle$
Values deleted	Remove 2 from $D_3$ .
Arc(s) added	$\langle x_1, x_3 \rangle$










	0	1	2	3
0				
1				
2				
3				

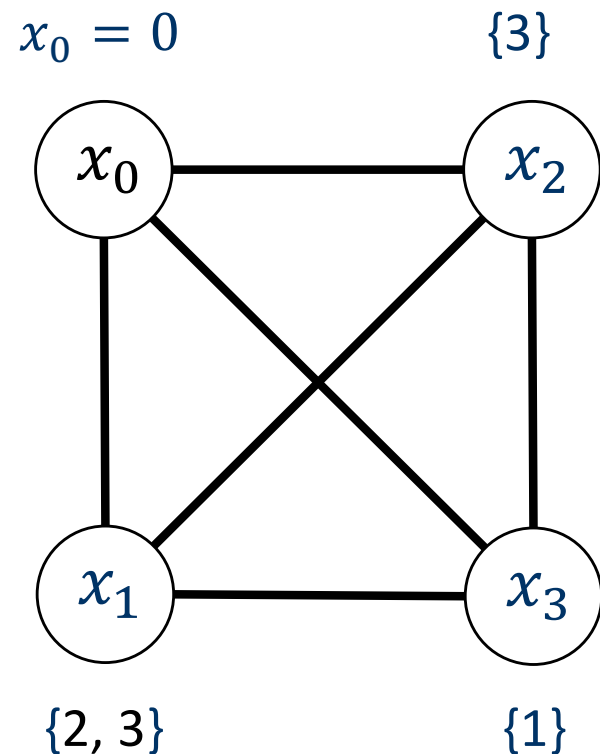


## Example 2: Execute AC-3 Algorithm

Queue:  $\langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_1, x_2 \rangle, \langle x_1, x_3 \rangle$

Step 4	
Arc removed	
Values deleted	
Arc(s) added	











	0	1	2	3
0				
1				
2				
3				

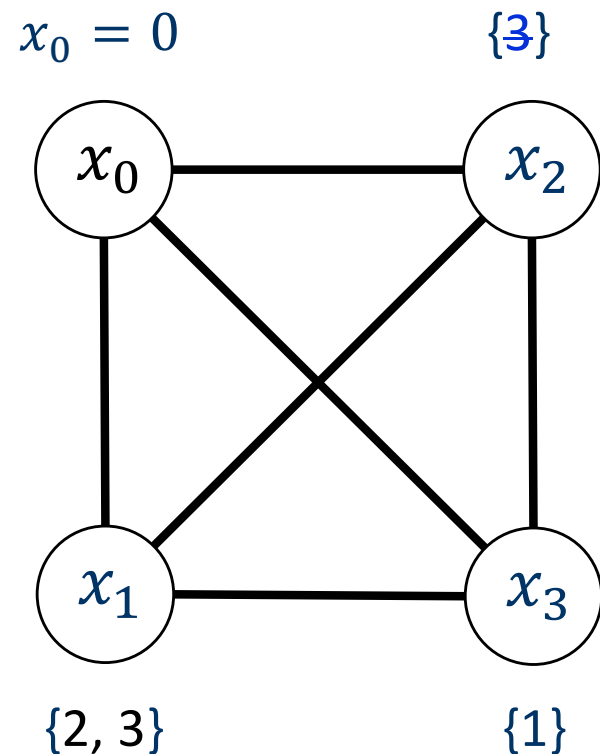


## Example 2: Execute AC-3 Algorithm

Queue:  $\langle \cancel{x_2}, \cancel{x_1} \rangle, \langle x_3, x_1 \rangle, \langle x_1, x_2 \rangle$

Step 4	
Arc removed	$\langle x_2, x_1 \rangle$
Values deleted	Remove 3 from $D_2$ .
Arc(s) added	(Return false before reaching this step)

	0	1	2	3
0				
1				
2				
3				



## Answer 2: Execute AC-3 Algorithm

---

Queue:  $\langle x_1, x_3 \rangle, \langle x_2, x_3 \rangle, \langle x_3, x_2 \rangle, \langle x_2, x_1 \rangle, \langle x_3, x_1 \rangle, \langle x_1, x_2 \rangle$   
 $x_0 = 0, D_1 = \{2, 3\}, D_2 = \{1, 3\}, D_3 = \{1, 2\}$

Step	Arc removed	Values to delete	Updated Domains	Arc(s) added
1	$\langle x_1, x_3 \rangle$	None	$D_1 = \{2, 3\}, D_2 = \{1, 3\}, D_3 = \{1, 2\}$	None
2	$\langle x_2, x_3 \rangle$	Remove 1 from $D_2$	$D_1 = \{2, 3\}, D_2 = \{3\}, D_3 = \{1, 2\}$	None
3	$\langle x_3, x_2 \rangle$	Remove 2 from $D_3$	$D_1 = \{2, 3\}, D_2 = \{3\}, D_3 = \{1\}$	$\langle x_1, x_3 \rangle$
4	$\langle x_2, x_1 \rangle$	Remove 3 from $D_2$	$D_1 = \{2, 3\}, D_2 = \{\}, D_3 = \{1\}$	
5				

$D_2$  is empty. Return **false**.

---

# PROPERTIES OF AC-3 ALGORITHM

# Properties of AC-3 Algorithm

---

- Order of removing arcs.
- Outcomes of AC-3 algorithm.
- Guaranteed to terminate? Complete?

# Order of Removing Arcs

---

Does the order of removing arcs  
affect the outcome of AC-3?

No. The “**queue**” should really be called a “**set**.”

# What are three outcomes of AC-3 algorithm?

---

1.

2.

3.



# What are three outcomes of AC-3 algorithm?

---

1. A domain is empty.
2. Every domain has one value.
3. Every domain has at least one value and at least one domain has at least two values.

# What are three outcomes of AC-3 algorithm?

---

1. A domain is empty.

No solution

2. Every domain has one value.

A unique solution

3. Every domain has at least one value and at least one domain has at least two values.

There may or may not be a solution.

# Is AC-3 Guaranteed to Terminate?

---

A. Yes.

B. No.

Why or why not?

# Is AC-3 Guaranteed to Terminate?

---

A. Yes.

B. No.

Why or why not?

Yes. Finite # of variables with finite sized domains. We cannot reduce the domains forever. Eventually, we check all the arcs or one domain becomes empty.

# Properties of AC-3 Algorithm

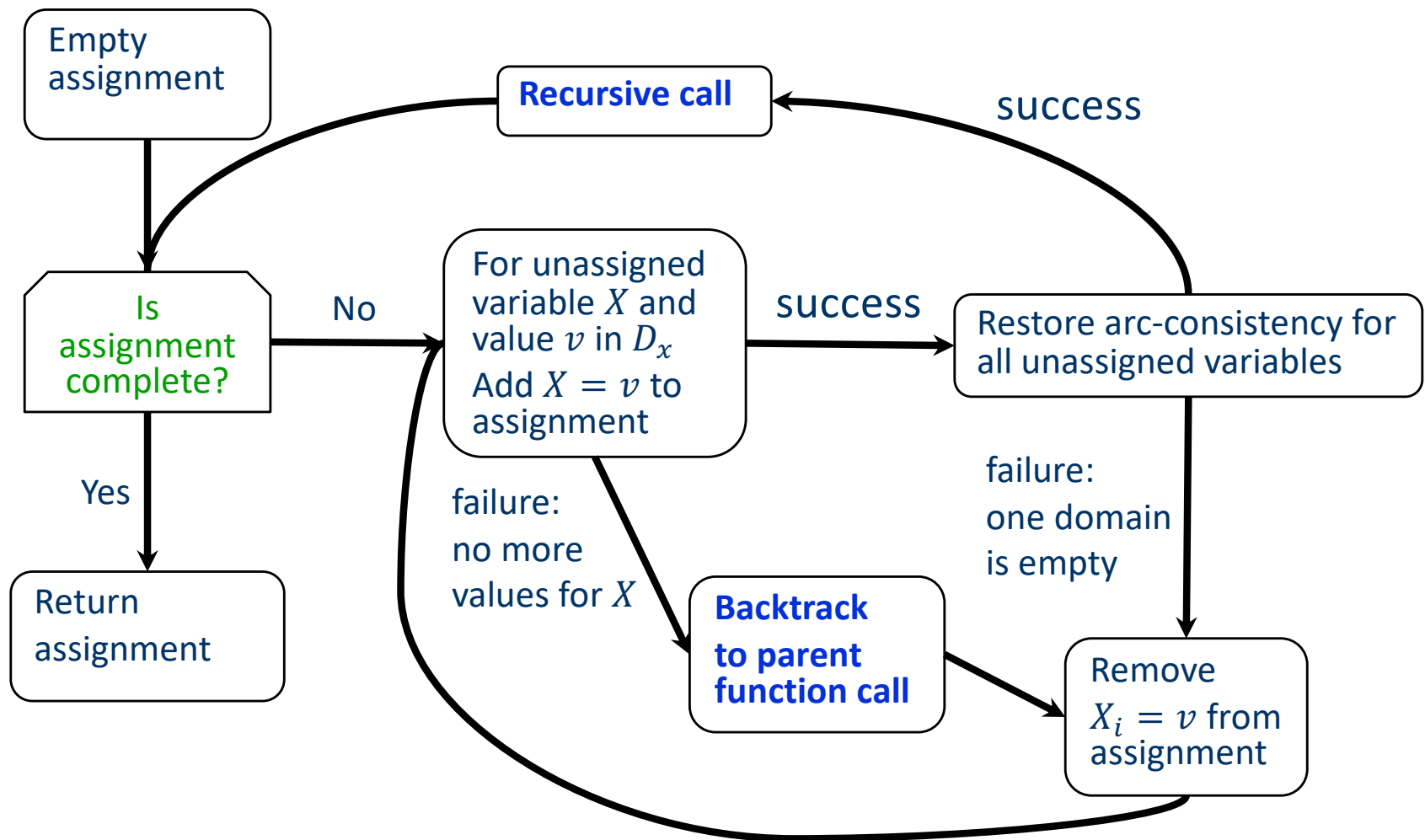
---

- Does the order of removing arcs affect the outcome?
  - No.
- What are three outcomes of AC-3 algorithm?
  1. A domain is empty. → no solution
  2. Every domain has one value. → a unique solution
  3. Every domain has at least one value and at least one domain has at least two values. → undetermined
- Is AC-3 guaranteed to terminate?
  - Yes. Finite # of variables with finite sized domains. We cannot reduce the domains forever. Eventually, we check all the arcs or one domain becomes empty.

---

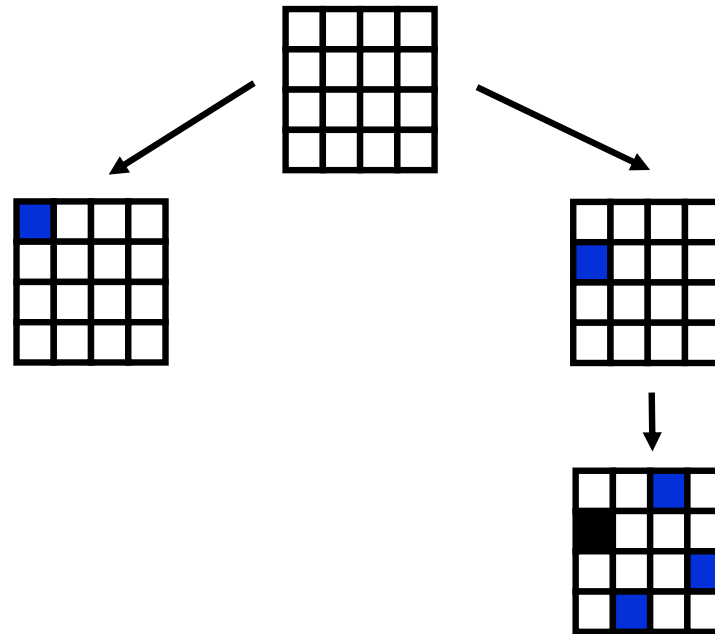
# BACKTRACKING SEARCH WITH AC-3

# Backtracking Search w/ AC-3



# Solving 4-Queens w/ Backtracking Search & AC-3

---





# Solving 4-Queens w/ Backtracking Search & AC-3

---

Choose the next queen to assign from left to right.

Choose the next value to assign from top to bottom.

Step	Assigning a Value and Revising Domains				What Next?
	$x_0$	$x_1$	$x_2$	$x_3$	
1	$x_0 = 0$				
2					
3					

# Solving 4-Queens w/ Backtracking Search & AC-3

---

After assigning  $x_0 = 0$ , execute AC-3.

Queue:  $\langle x_1, x_0 \rangle, \langle x_2, x_0 \rangle, \langle x_3, x_0 \rangle, \langle x_1, x_2 \rangle,$   
 $\langle x_2, x_1 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_1 \rangle, \langle x_2, x_3 \rangle, \langle x_3, x_2 \rangle$

- Every binary constraint has two arcs. We need to add both to the queue.
- The purpose of adding an arc  $\langle x_i, x_j \rangle$  to the queue is to potentially remove values from  $x_i$ 's domain. Because of this, there is no need to add any arc  $\langle x_0, x_j \rangle$  to the queue since we have fixed  $x_0$  to be 0 at this step.

# Solving 4-Queens w/ Backtracking Search & AC-3

---

After assigning  $x_0 = 0$ , execute AC-3.

Queue:  $\langle x_1, x_0 \rangle, \langle x_2, x_0 \rangle, \langle x_3, x_0 \rangle, \langle x_1, x_2 \rangle,$   
 $\langle x_2, x_1 \rangle, \langle x_1, x_3 \rangle, \langle x_3, x_1 \rangle, \langle x_2, x_3 \rangle, \langle x_3, x_2 \rangle$

The steps of AC-3 will depend on the order of removing the arcs from the queue.

However, the result of AC-3 will be that one of  $x_1, x_2$ , or  $x_3$ 's domain becomes empty, meaning that assigning  $x_0 = 0$  does not lead to a solution.

# Solving 4-Queens w/ Backtracking Search & AC-3

---

Choose the next queen to assign from left to right.

Choose the next value to assign from top to bottom.

Step	Assigning a Value and Revising Domains				What Next?
	$x_0$	$x_1$	$x_2$	$x_3$	
1	$x_0 = 0$	One domain becomes empty.			Backtrack
2	$x_0 = 1$	3	0	2	Continue
3		$x_1 = 3$	0	2	Continue
4			$x_2 = 0$		Continue
5				$x_3 = 2$	Solution Found!

# Extra Example 1: Execute AC-3 Algorithm

---

Queue:  $\langle x_2, x_1 \rangle$  ,  $\langle x_3, x_1 \rangle$  ,  $\langle x_3, x_2 \rangle$  ,  $\langle x_1, x_2 \rangle$  ,  $\langle x_1, x_3 \rangle$  ,  $\langle x_2, x_3 \rangle$   
 $x_0 = 1$ ,  $D_1 = \{3\}$ ,  $D_2 = \{0, 2\}$ ,  $D_3 = \{0, 2, 3\}$

Step	Arc removed	Values to delete	Updated Domains	Arc(s) added
1				
2				
3				
4				
5				
6				

# Extra 1 Answer: Execute AC-3 Algorithm

Queue:  $\langle x_2, x_1 \rangle$ ,  $\langle x_3, x_1 \rangle$ ,  $\langle x_3, x_2 \rangle$ ,  $\langle x_1, x_2 \rangle$ ,  $\langle x_1, x_3 \rangle$ ,  $\langle x_2, x_3 \rangle$   
 $x_0 = 1$ ,  $D_1 = \{3\}$ ,  $D_2 = \{0, 2\}$ ,  $D_3 = \{0, 2, 3\}$

Step	Arc removed	Values to delete	Updated Domains	Arc(s) added
1	$\langle x_2, x_1 \rangle$	Remove 2 from $D_2$ .	$D_1 = \{3\}$ , $D_2 = \{0\}$ , $D_3 = \{0, 2, 3\}$	None
2	$\langle x_3, x_1 \rangle$	Remove 3 from $D_3$ .	$D_1 = \{3\}$ , $D_2 = \{0\}$ , $D_3 = \{0, 2\}$	None
3	$\langle x_3, x_2 \rangle$	Remove 0 from $D_3$ .	$D_1 = \{3\}$ , $D_2 = \{0\}$ , $D_3 = \{2\}$	None
4	$\langle x_1, x_2 \rangle$	None	$D_1 = \{3\}$ , $D_2 = \{0\}$ , $D_3 = \{2\}$	None
5	$\langle x_1, x_3 \rangle$	None	$D_1 = \{3\}$ , $D_2 = \{0\}$ , $D_3 = \{2\}$	None
6	$\langle x_2, x_3 \rangle$	None	$D_1 = \{3\}$ , $D_2 = \{0\}$ , $D_3 = \{2\}$	None

Return **true**.