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

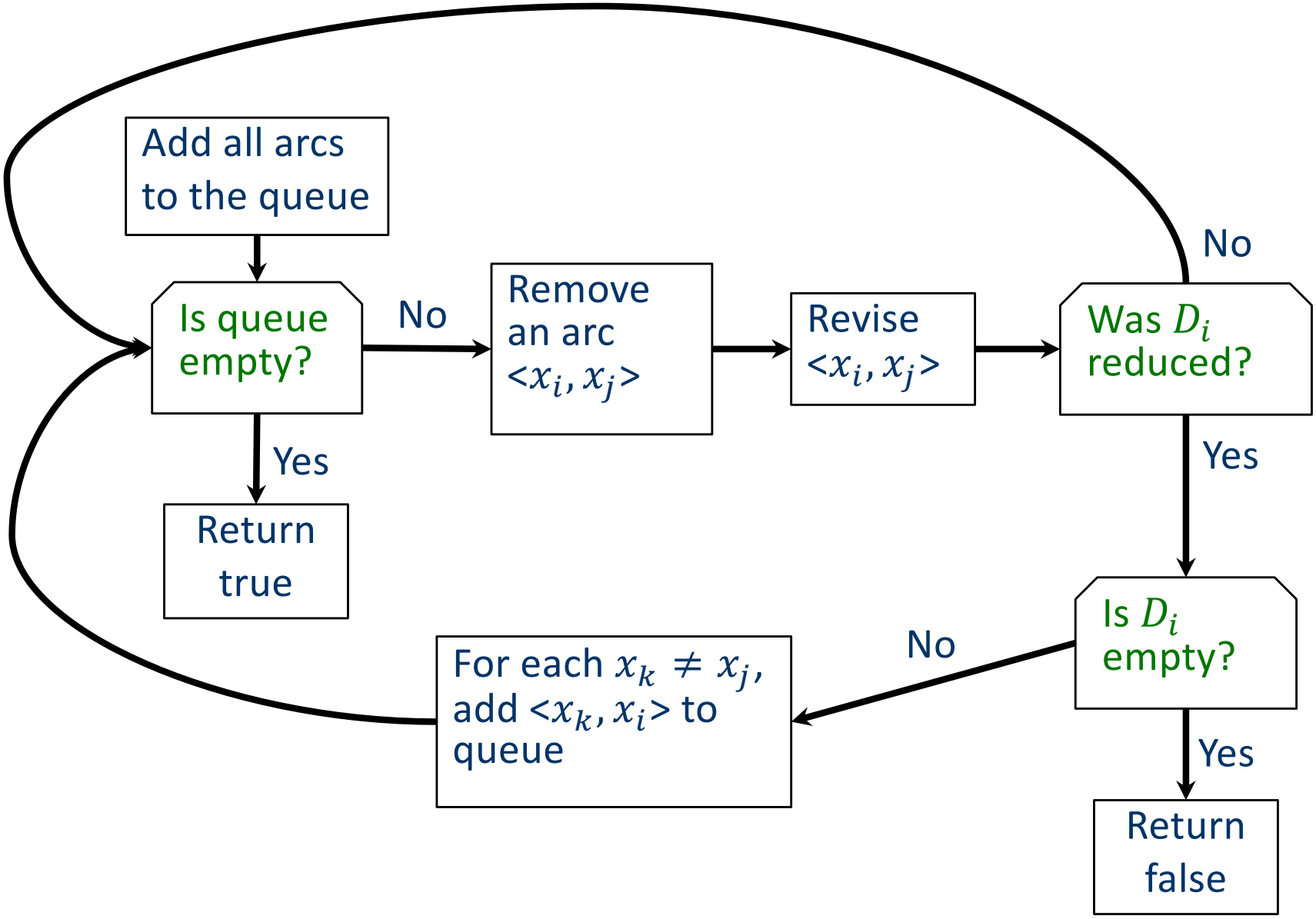
- Unary constraints (over 1 variable)

  Examples: $\mathrm{c}(x): x = 2;\ \mathrm{c}(y): y > 5$
- Binary constraints (over 2 variables)

  Examples: $\mathrm{c}(x, y): x + y < 6$
- Higher-order (n-ary) constraints (over ≥ 3 variables)

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

AC-3 Arc-Consistency Algorithm (Flowchart)

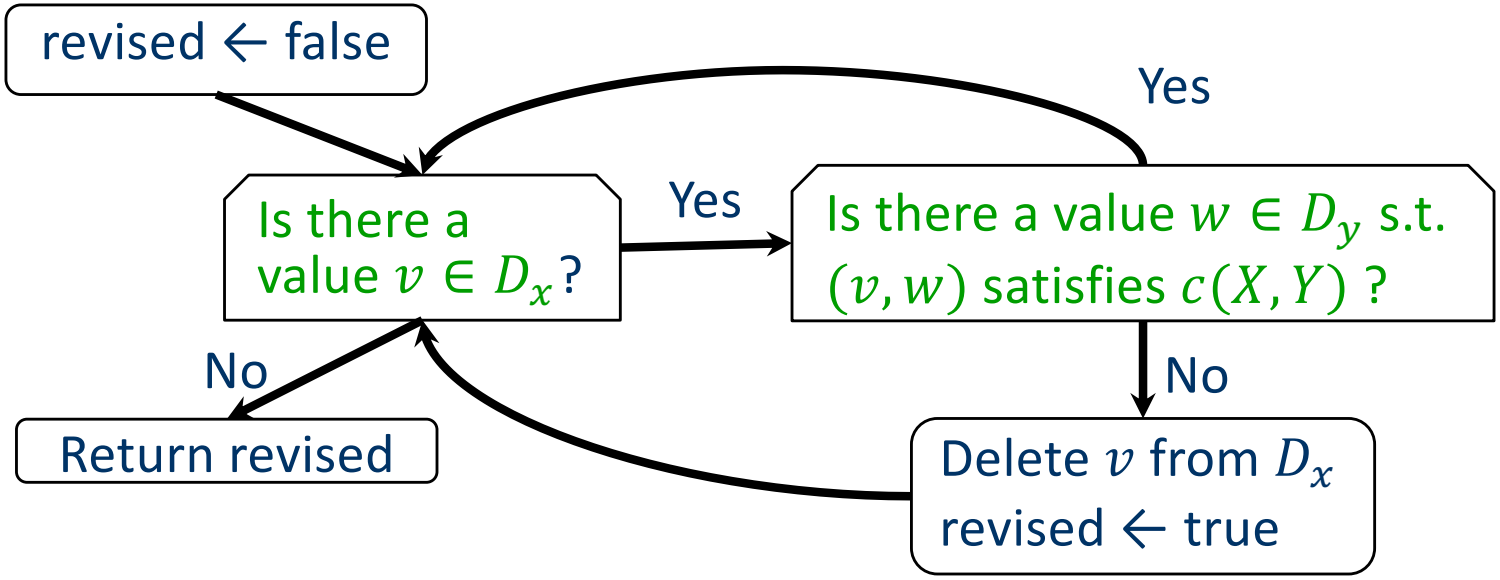


AC-3 Arc-Consistency Algorithm (Pseudocode)

```
function AC-3(csp)
  # returns false if an inconsistency is found and true otherwise
  inputs: csp, a binary CSP with components (X, D, C)
  local variables: queue, a queue of arcs, initially all the arcs in csp

  add all the arcs to the queue
  while queue is not empty do
      <X_i,X_j> ← REMOVE-FIRST(queue)
      if REVISE(csp, X_i, X_j) then
          if D_i is empty then
              return false
          for each X_k in X_i.NEIGHBORS - {X_j} do
              add <X_k,X_i> to queue
  return true
```

Revise Domain to Restore Arc-Consistency

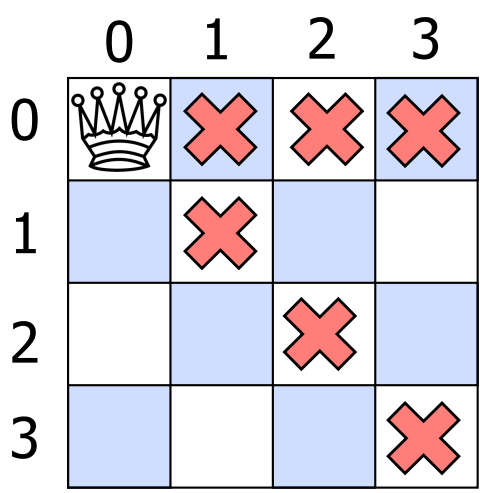


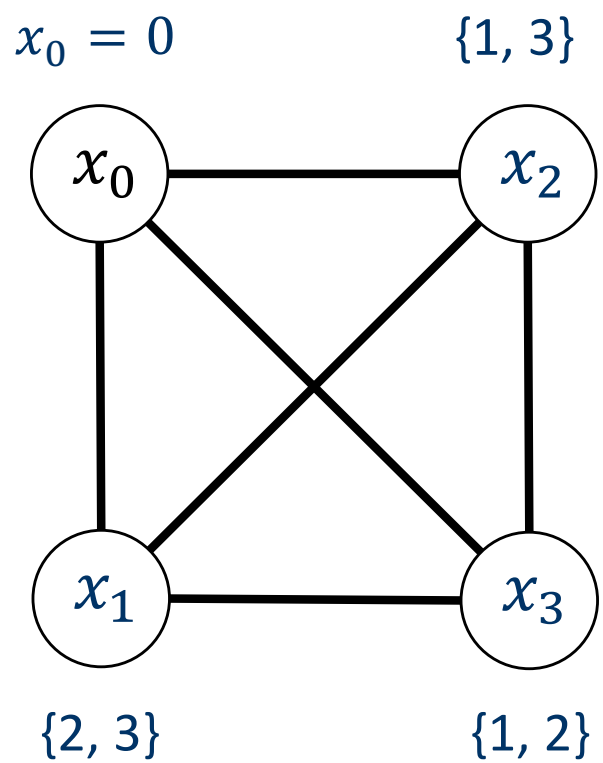

```
function REVISE(csp, X, Y)
    revised ← false
    for each v in D_x do
        if no value w in D_y allows (v,w) to satisfy the
constraint between X and Y then
            delete v from D_x
            revised ← true
    return revised
```

Example 1: Execute AC-3 Algorithm

Queue: $< x_1, x_2 >, < x_2, x_1 >, < x_3, x_1 >, < x_2, x_3 >, < x_1, x_3 >, < x_3, x_2 >$

Step 1	
Arc removed	
Values deleted	
Arc(s) added	

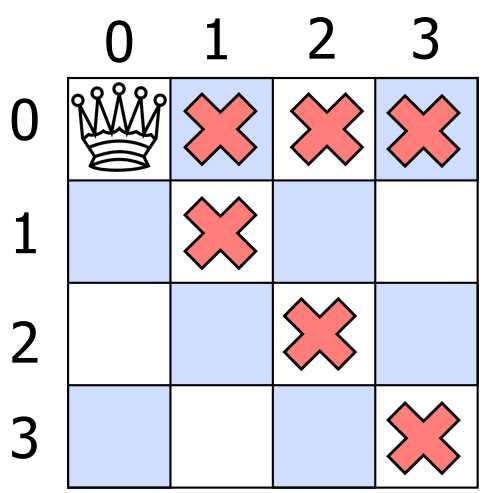


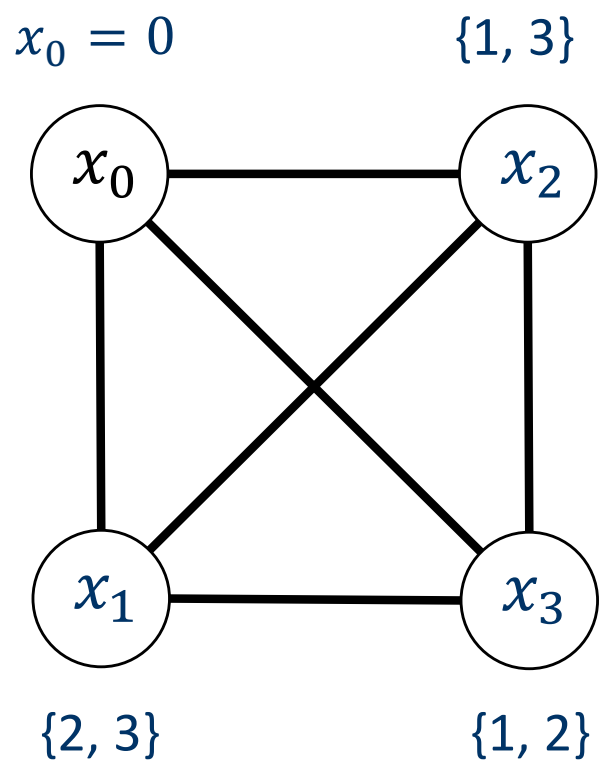


Example 1: Execute AC-3 Algorithm

Queue: $\cancel{< x_1, x_2 >}$, $< x_2, x_1 >$, $< x_3, x_1 >$, $< x_2, x_3 >$, $< x_1, x_3 >$, $< x_3, x_2 >$

Step 1	
Arc removed	$< x_1, x_2 >$
Values deleted	
Arc(s) added	



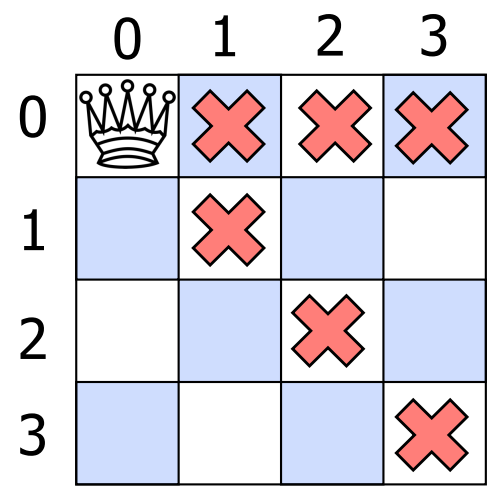


Step 1: Restore Arc Consistency

How can we make the arc below consistent?

$$< x_1, x_2 > (or \ < x_1, c(x_1, x_2) >)$$

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.



Example 1: Execute AC-3 Algorithm

Queue: $< x_1, x_2 >, < x_2, x_1 >, < x_3, x_1 >, < x_2, x_3 >, < x_1, x_3 >, < x_3, x_2 >$

$x_0 = 0$, $D_1 = \{2, 3\}$, $D_2 = \{1, 3\}$, $D_3 = \{1, 2\}$

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

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 $< X_i, X_j >$,

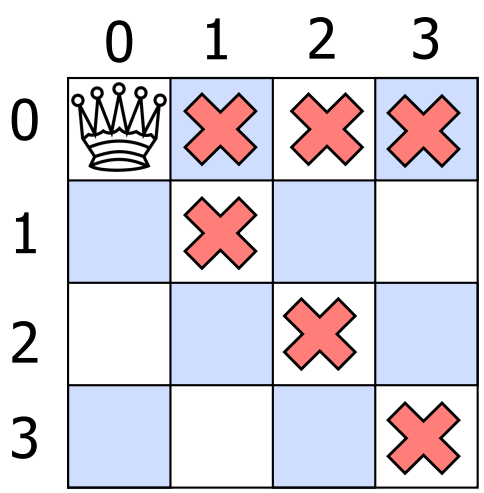
Add any arc $< X_k, X_i >$ to the queue where

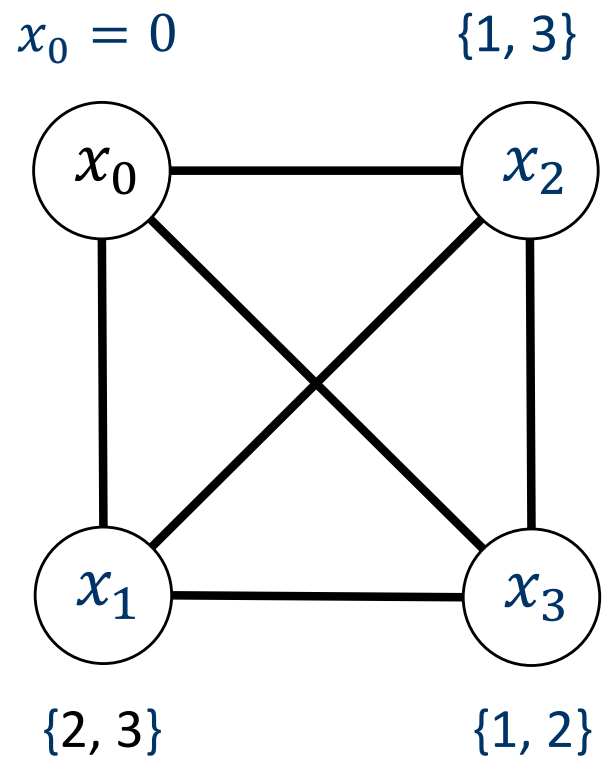
- X_k is a neighbour of X_i, and
- $X_k \neq X_j$.

Example 2: Execute AC-3 Algorithm

Queue: $< x_1, x_3 >, < x_2, x_3 >, < x_3, x_2 >, < x_2, x_1 >, < x_3, x_1 >, < x_1, x_2 >$

Step 1	
Arc removed	
Values deleted	
Arc(s) added	

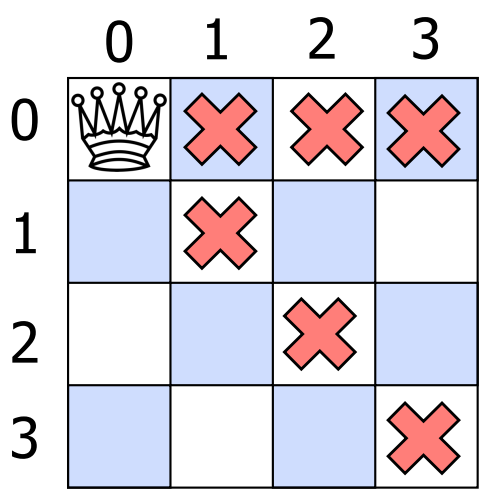


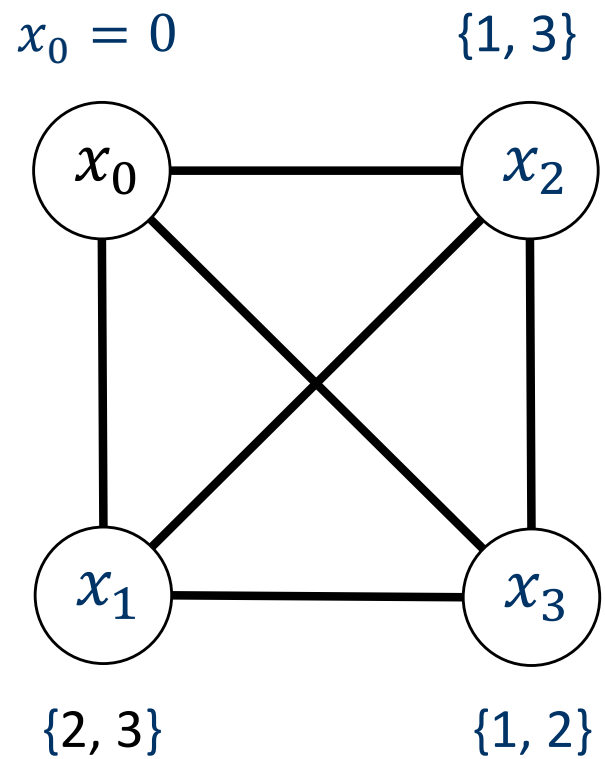


Example 2: Execute AC-3 Algorithm

Queue: $< x_2, x_3 >, < x_3, x_2 >, < x_2, x_1 >, < x_3, x_1 >, < x_1, x_2 >$

Step 2	
Arc removed	
Values deleted	
Arc(s) added	

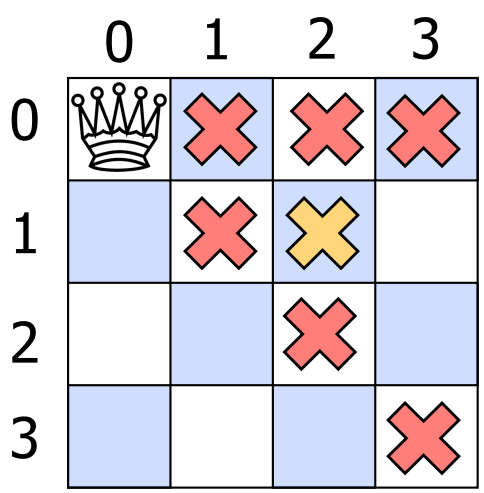


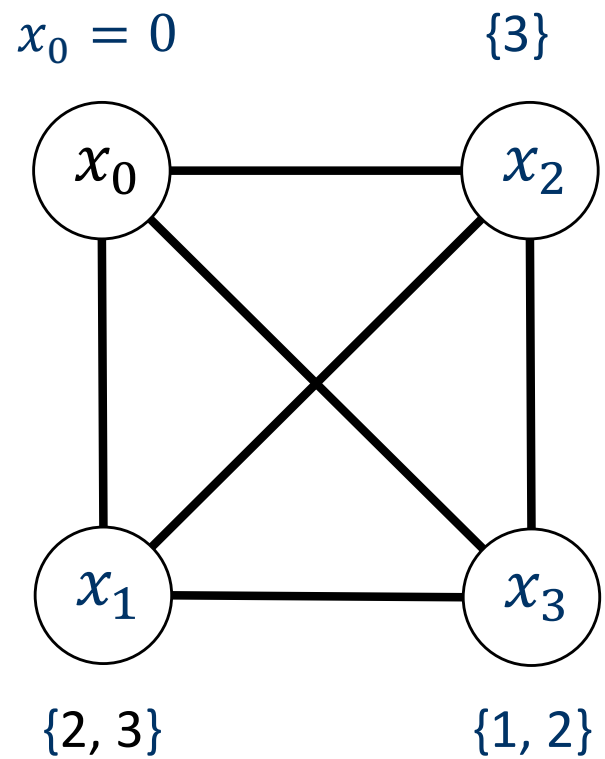


Example 2: Execute AC-3 Algorithm

Queue: $< x_3, x_2 >, < x_2, x_1 >, < x_3, x_1 >, < x_1, x_2 >$

Step 3	
Arc removed	
Values deleted	
Arc(s) added	

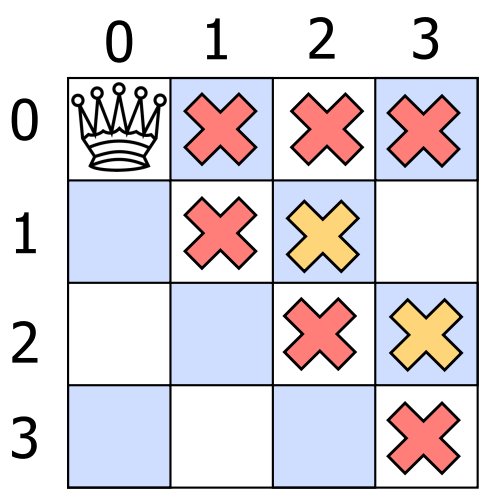


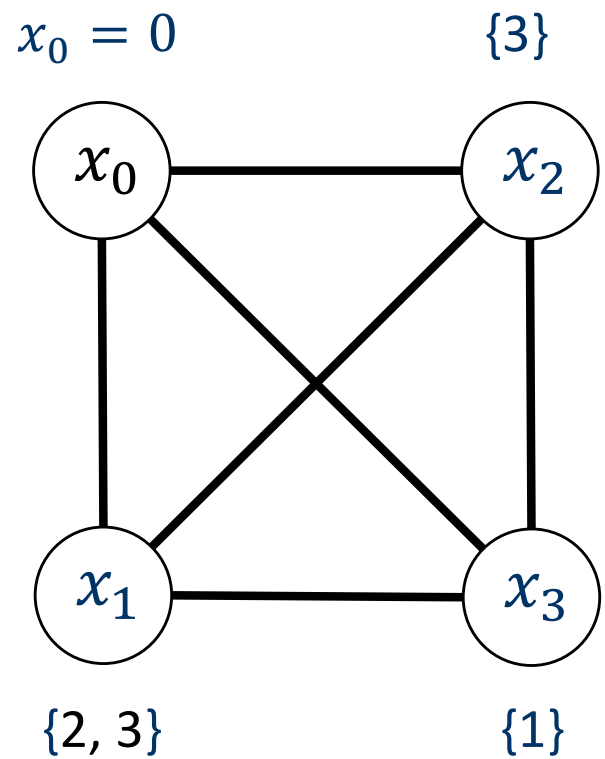


Example 2: Execute AC-3 Algorithm

Queue: $< x_2, x_1 >, < x_3, x_1 >, < x_1, x_2 >$

Step 4	
Arc removed	
Values deleted	
Arc(s) added	





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.

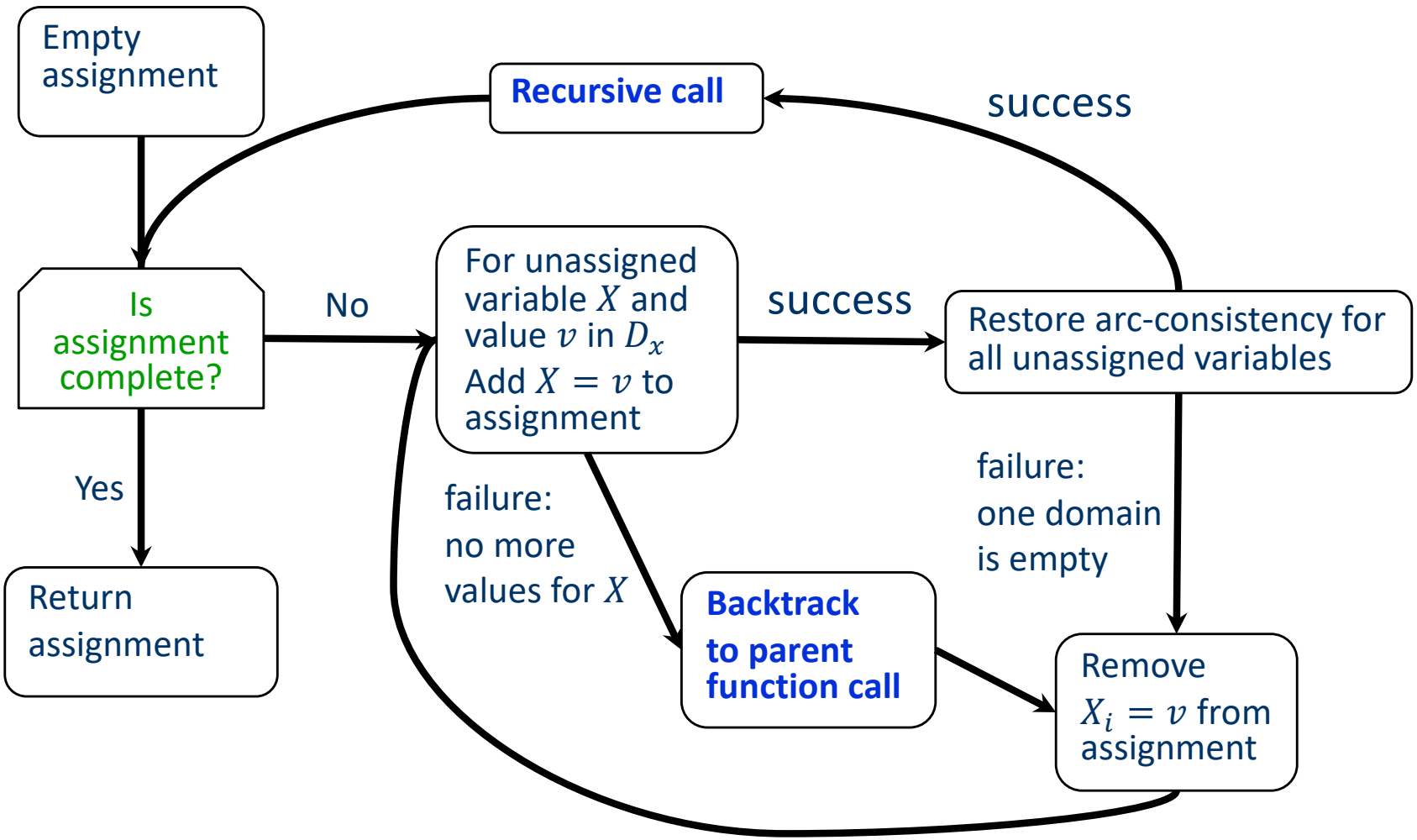
Is AC-3 Guaranteed to Terminate?

A. Yes.
B. No.

Why or why not?

BACKTRACKING SEARCH WITH AC-3

Backtracking Search w/ AC-3



Extra Example 1: Execute AC-3 Algorithm

Queue: $< x_2, x_1 >$, $< x_3, x_1 >$, $< x_3, x_2 >$, $< x_1, x_2 >$, $< x_1, x_3 >$, $< x_2, x_3 >$

$x_0 = 1$, $D_1 = \{3\}$, $D_2 = \{0, 2\}$, $D_3 = \{0, 2, 3\}$

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