



CSC 384

Introduction to Artificial Intelligence

CSP 2

Alice Gao and Randy Hickey
Winter 2023

Learning Goals

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

Arc Consistency

- Determine whether a binary constraint is arc-consistent.
- Explain why the arc consistency of a binary constraint is not symmetric.

Forward Checking

- Explain the idea of Forward Checking and
- Explain how to combine Backtracking Search and Forward Checking.
- Explain how Forward Checking can reduce the size of the search tree.
- Trace the execution of Backtracking Search and Forward Checking.



Outline

1. Arc Consistency
2. Backtracking Search with Forward Checking

ARC CONSISTENCY

Motivating Arc Consistency




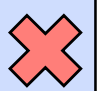



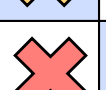

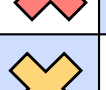

- $x_0 = 0$ and $x_1 = 2$ do not lead to a solution. Why?

	0	1	2	3
0				
1				
2				
3				

Motivating Arc Consistency

- $x_0 = 0$ and $x_1 = 2$ do not lead to a solution. Why?

No possible value for x_2 😭

	0	1	2	3
0				
1				
2				
3				

Notation for an Arc

- X and Y are two random variables.
- D_X and D_Y are their respective domains.
- $c(X, Y)$ is a binary constraint.



Arc Consistency Definition

$\langle X, c(X, Y) \rangle$ is arc-consistent if and only if
for every value v in D_X ,
there exists a value w in D_Y ,
such that (v, w) satisfies the constraint $c(X, Y)$.

Question 1: Checking Arc Consistency

Assume that $D_X = \{1\}$ and $D_Y = \{1,2\}$.

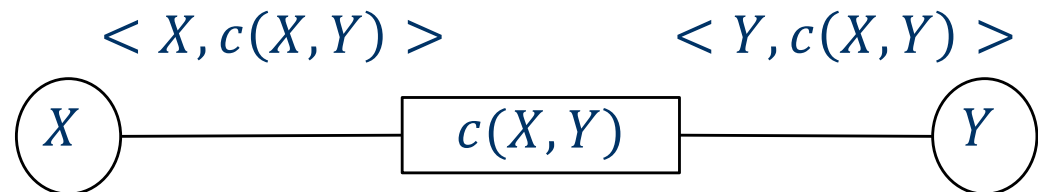
Consider the constraint $c(X, Y): X < Y$.

Is the arc $< X, c(X, Y) >$ consistent?

A. Yes.

B. No.

C. I don't know.



Answer 1: Checking Arc Consistency

Assume that $D_X = \{1\}$ and $D_Y = \{1,2\}$.

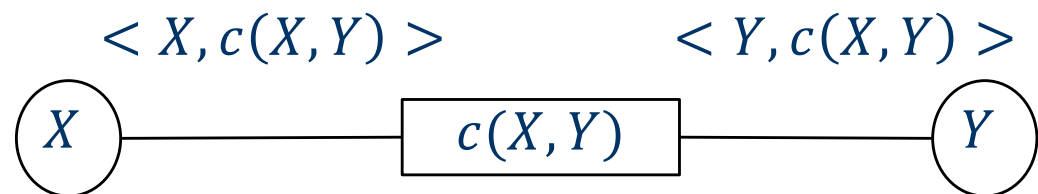
Consider the constraint $c(X, Y): X < Y$.

Is the arc $\langle X, c(X, Y) \rangle$ consistent?

A. **Yes.**

B. No.

C. I don't know.



If $X = 1$, then $Y = 2$ satisfies the constraint.

Question 2: Checking Arc Consistency

Assume that $D_X = \{1\}$ and $D_Y = \{1,2\}$.

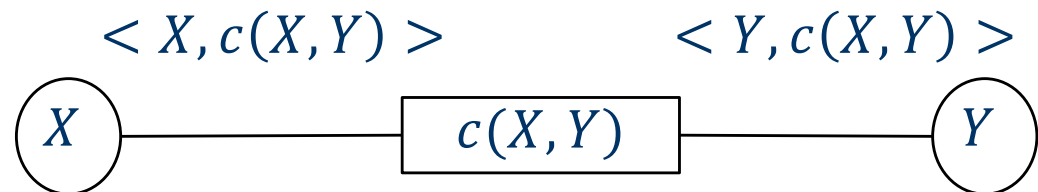
Consider the constraint $c(X, Y): X < Y$.

Is the arc $\langle Y, c(X, Y) \rangle$ consistent?

A. Yes.

B. No.

C. I don't know.



Answer 2: Checking Arc Consistency

Assume that $D_X = \{1\}$ and $D_Y = \{1,2\}$.

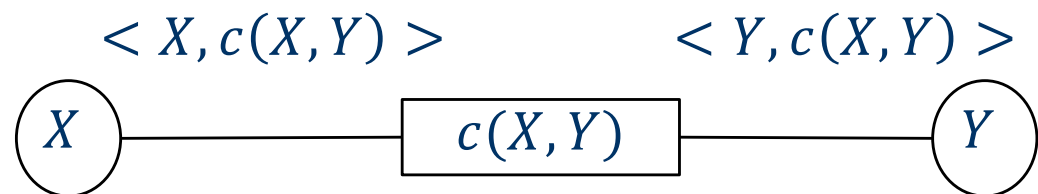
Consider the constraint $c(X, Y): X < Y$.

Is the arc $\langle Y, c(X, Y) \rangle$ consistent?

A. Yes.

B. **No.**

C. I don't know.



If $Y = 1$, then no value of X satisfies the constraint.

Arc Consistency is not Symmetric

$\langle X, c(X, Y) \rangle$ is consistent

\nLeftrightarrow

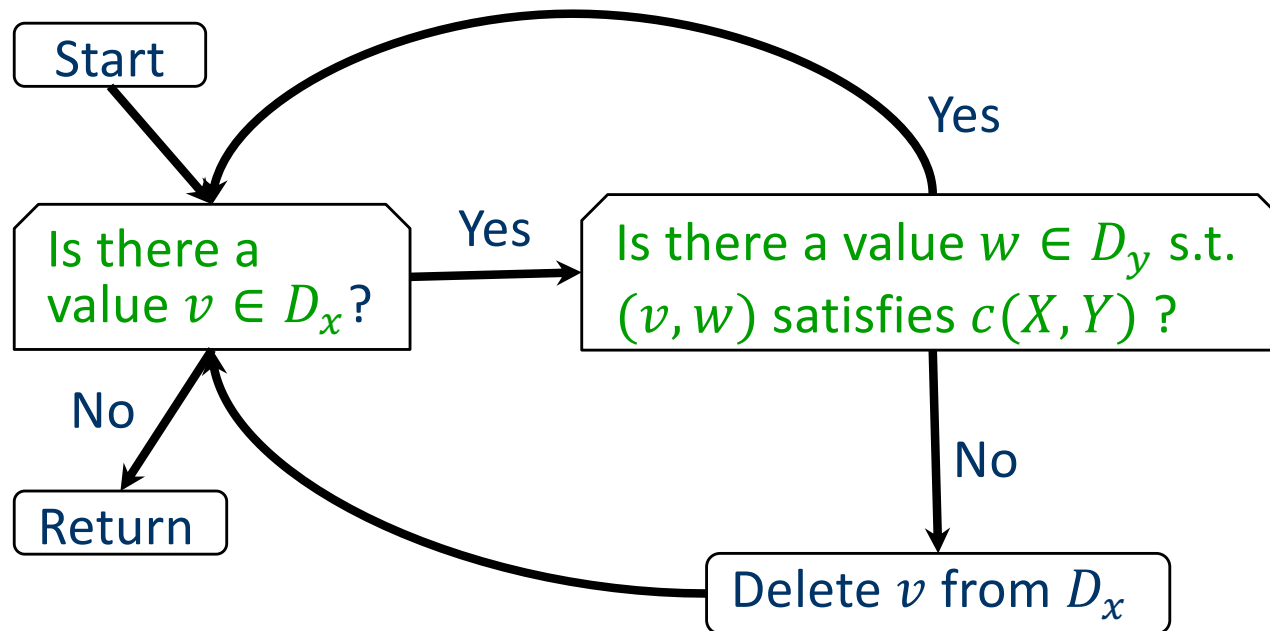
$\langle Y, c(X, Y) \rangle$ is consistent



Restore Arc Consistency

- Suppose that $\langle X, c(X, Y) \rangle$ is not consistent.
- At least one value $v \in D_x$ is causing the inconsistency.
 - For every $w \in D_y$, (v, w) violates the constraint $c(X, Y)$.
- Restore arc consistency by removing every value in D_x causing the inconsistency.

Revise Domain to Restore Arc-Consistency



1. function REVISE(csp, X , Y)
2. for each v in D_x do
3. if no value w in D_y allows (v, w) to satisfy the constraint between X and Y then
4. delete v from D_x

BACKTRACKING SEARCH WITH FORWARD CHECKING

Forward Checking (for binary constraints)

After assigning variable X to a value

For each unassigned variable Y connected to X
by a constraint $c(X, Y)$

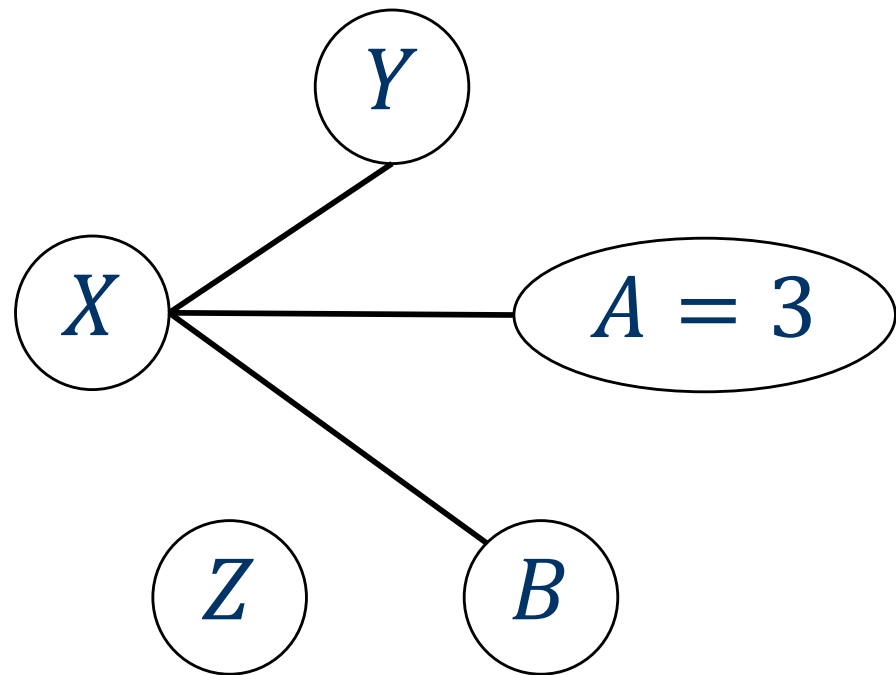
make $\langle Y, c(X, Y) \rangle$ arc-consistent.

(remove any value w in D_y violating the constraint c .)

Q1: Forward Checking for Binary Constraints

After assigning $X = 0$,
which arcs do we need to check for consistency?

- (A) $\langle Y, c(X, Y) \rangle$
- (B) $\langle Z, c(X, Z) \rangle$
- (C) $\langle A, c(X, A) \rangle$
- (D) $\langle B, c(X, B) \rangle$



Answer 1: Forward Checking for Binary Constraints

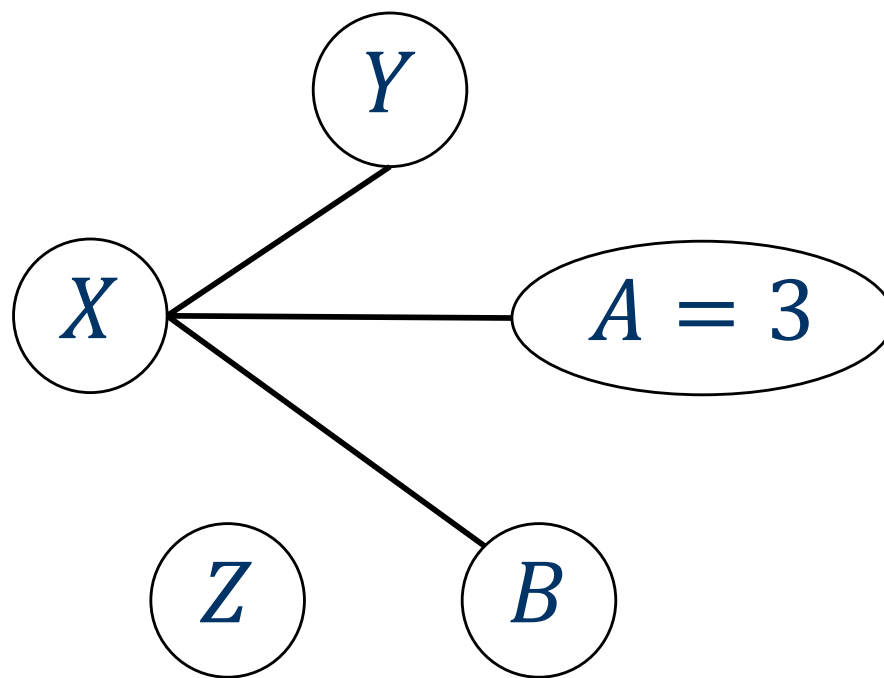
After assigning $X = 0$,
which arcs do we need to check for consistency?

(A) $\langle Y, c(X, Y) \rangle$

(B) $\langle Z, c(X, Z) \rangle$

(C) $\langle A, c(X, A) \rangle$

(D) $\langle B, c(X, B) \rangle$



Forward Checking (for all constraints)

After assigning variable X to a value

For every constraint c involving X

If c has exactly one unassigned variable Y in its scope

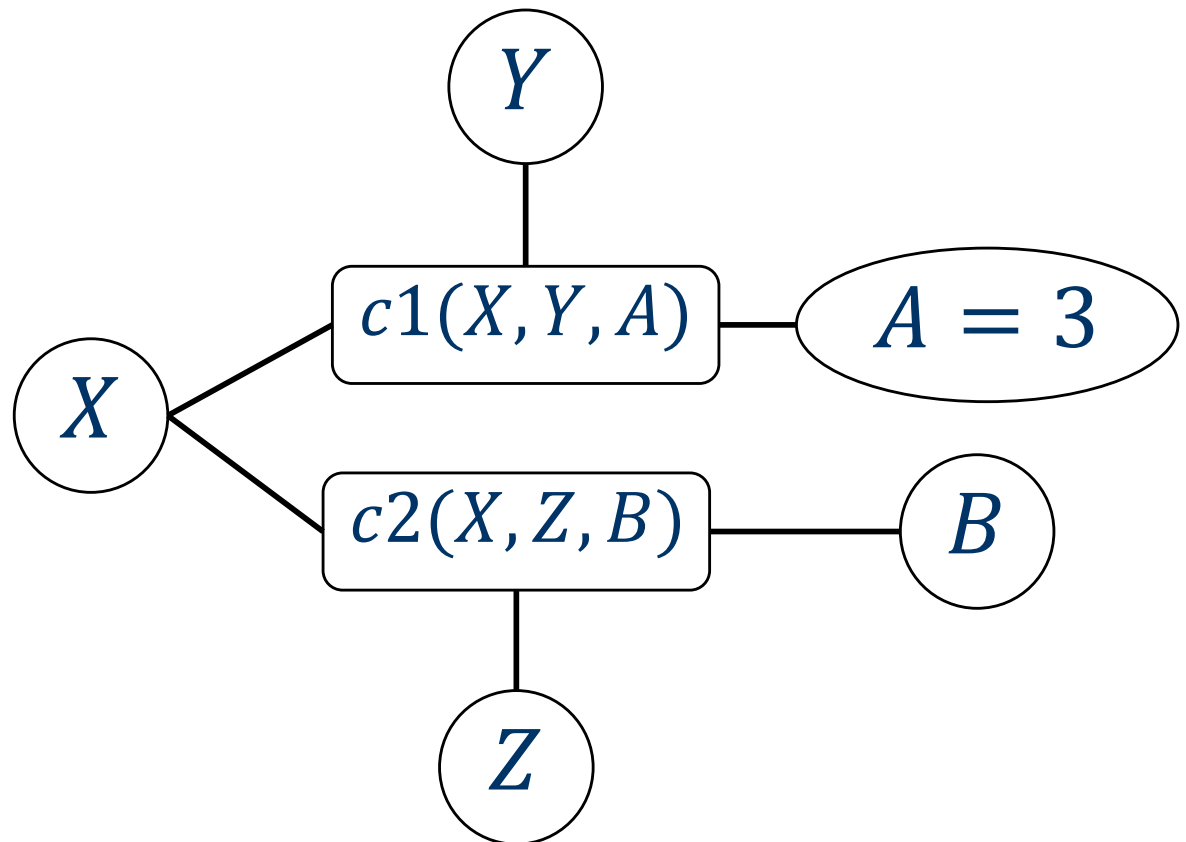
make $\langle Y, c \rangle$ arc-consistent.

(remove any value w in D_y violating the constraint c .)

Q2: Forward Checking for All Constraints

After assigning $X = 0$,
which arcs do we need to check for consistency?

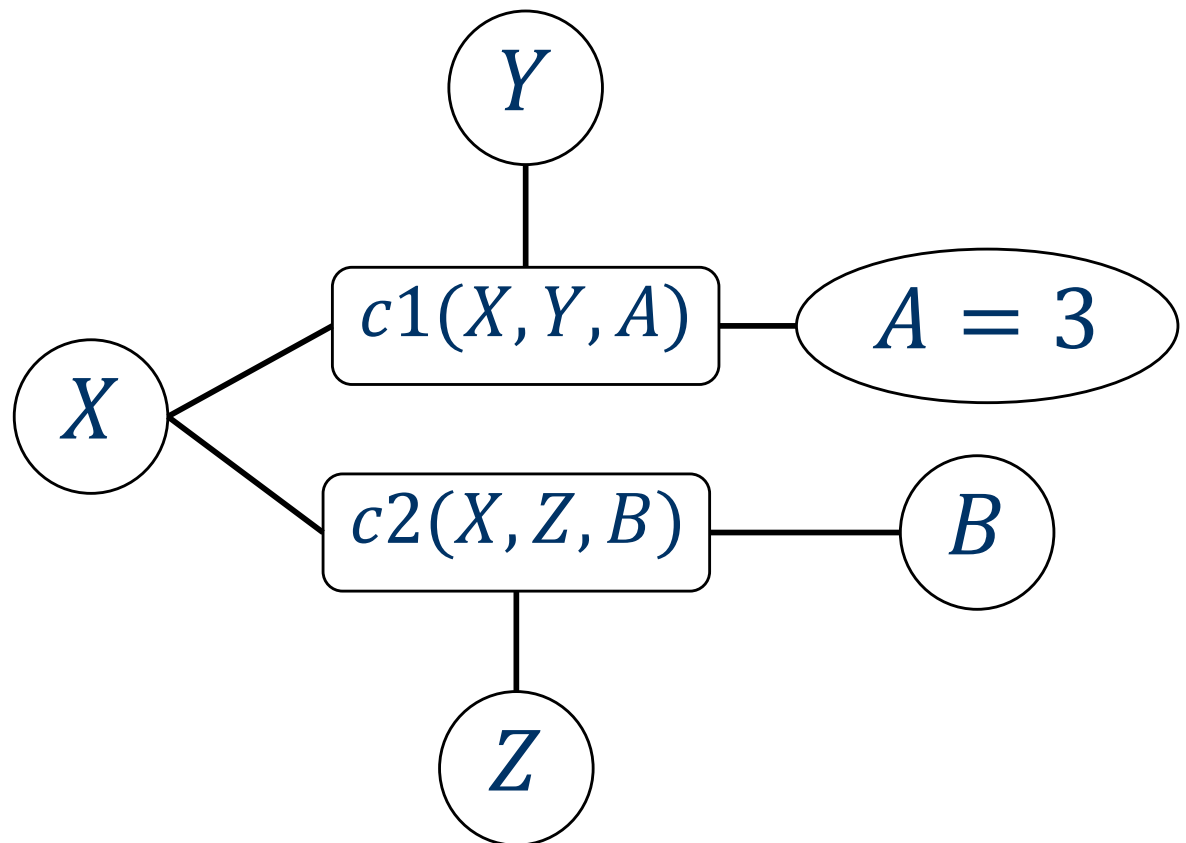
- (A) $\langle Y, c1 \rangle$
- (B) $\langle A, c1 \rangle$
- (C) $\langle Z, c2 \rangle$
- (D) $\langle B, c2 \rangle$



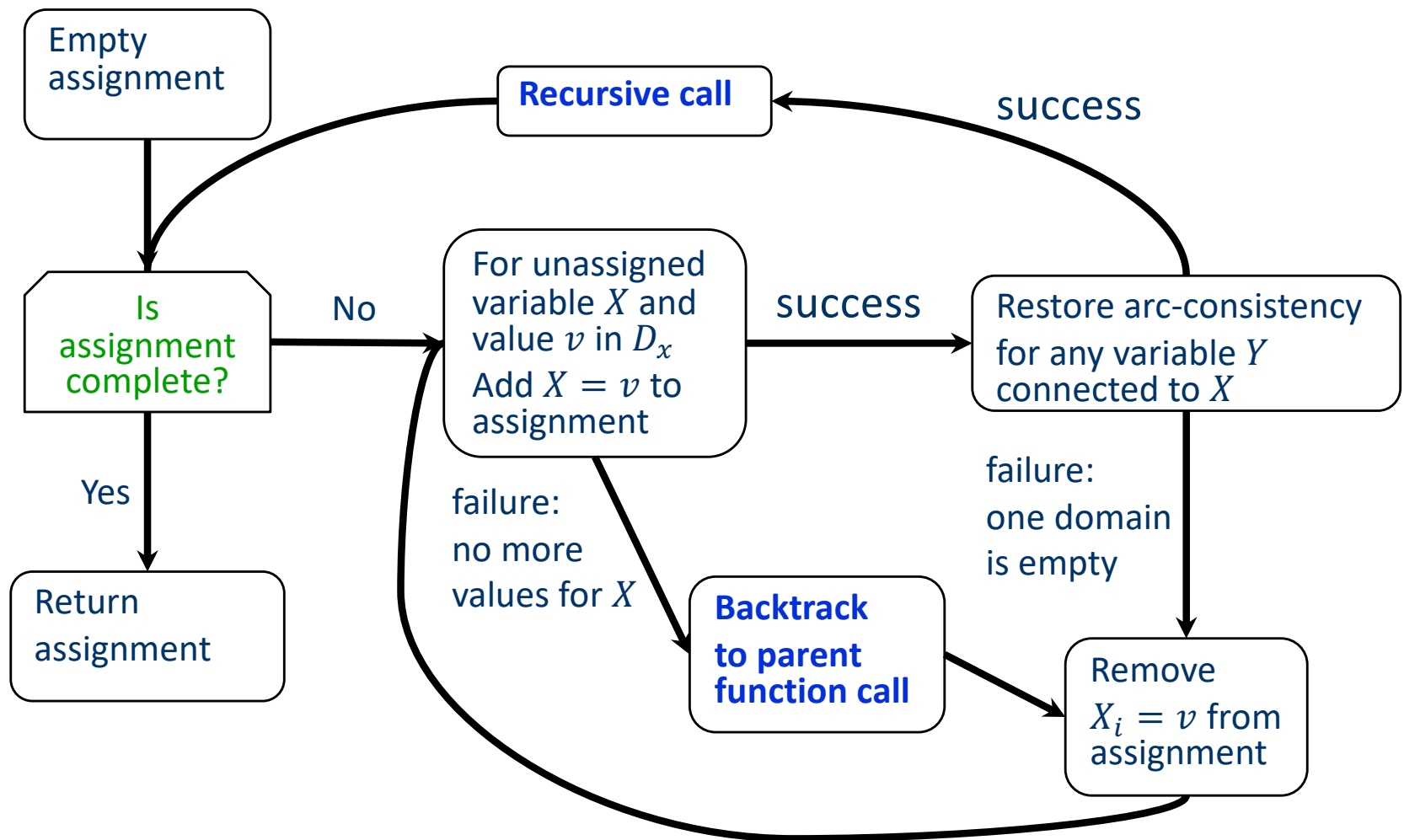
Answer 2: Forward Checking for Binary Constraints

After assigning $X = 0$,
which arcs do we need to check for consistency?

- (A) $\langle Y, c1 \rangle$
- (B) $\langle A, c1 \rangle$
- (C) $\langle Z, c2 \rangle$
- (D) $\langle B, c2 \rangle$




Backtracking Search w/ Forward Checking

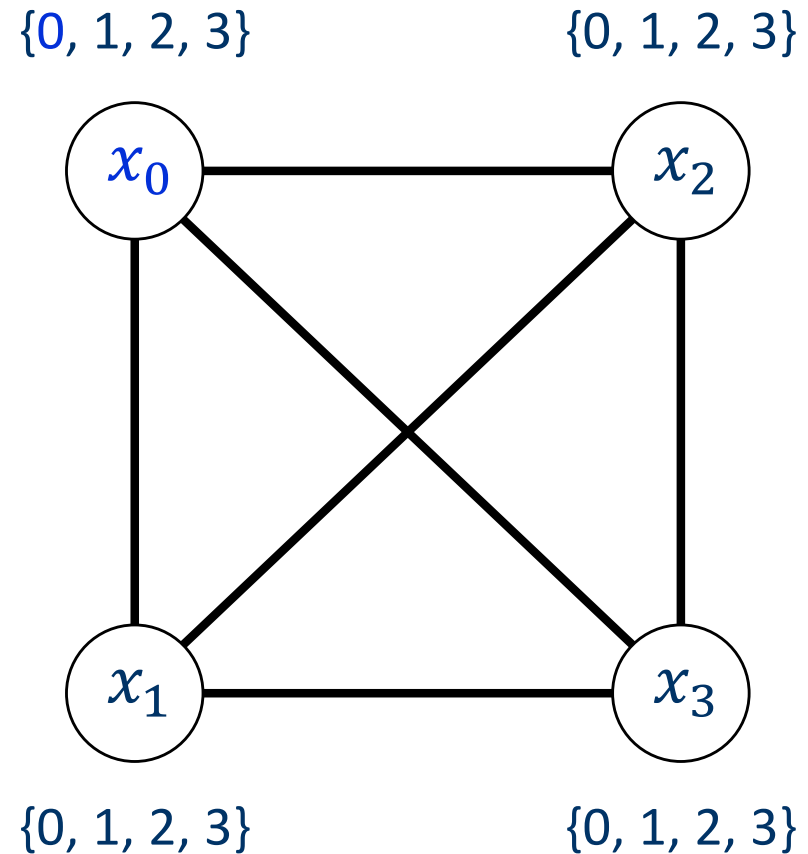


Backtracking Search w/ Arc Consistency








```
1. function BACKTRACKING-SEARCH(csp)
2.   return BACKTRACK({}, CSP)
3.
4. function BACKTRACK(assignment, csp)
5.   if assignment is complete then return assignment
6.   var <- SELECT-UNASSIGNED-VARIABLE(csp)
7.   for each value in ORDER-DOMAIN-VALUES(var, assignment, csp) do
8.     add {var = value} to assignment
9.     inferences <- INFERENCE(csp, var, value)
10.    if inferences ≠ failure then
11.      add inferences to assignment
12.      result <- BACKTRACK(assignment, csp)
13.      if result ≠ no solution then
14.        return result
15.      remove {var = value} and inferences from assignment
16.    return no solution
```

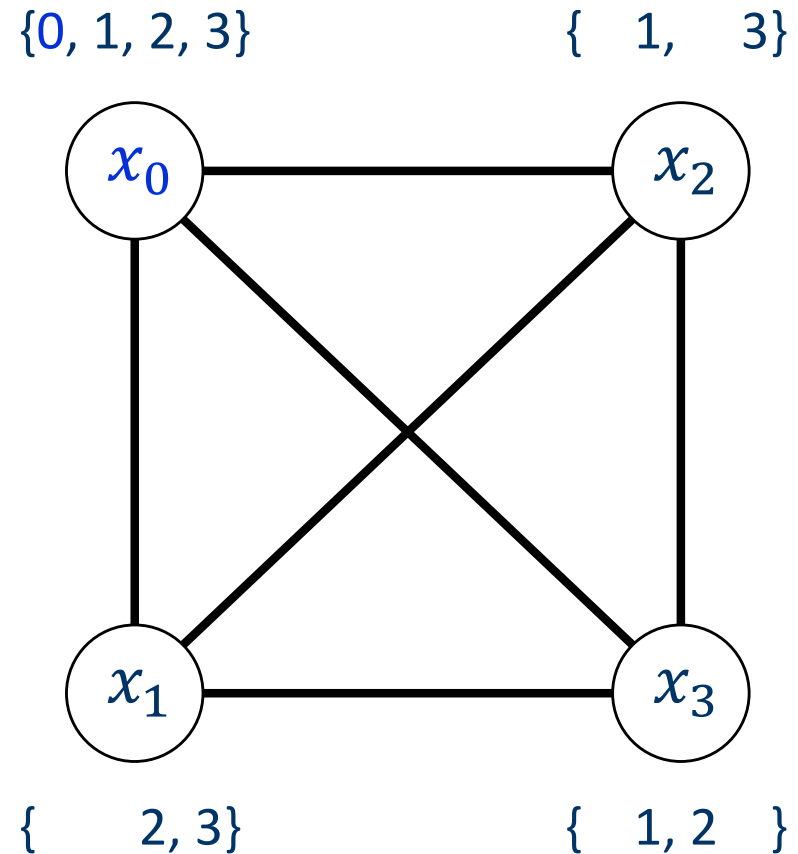

Solving 4-Queens using Forward Checking

	0	1	2	3
0				
1				
2				
3				



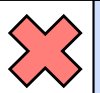







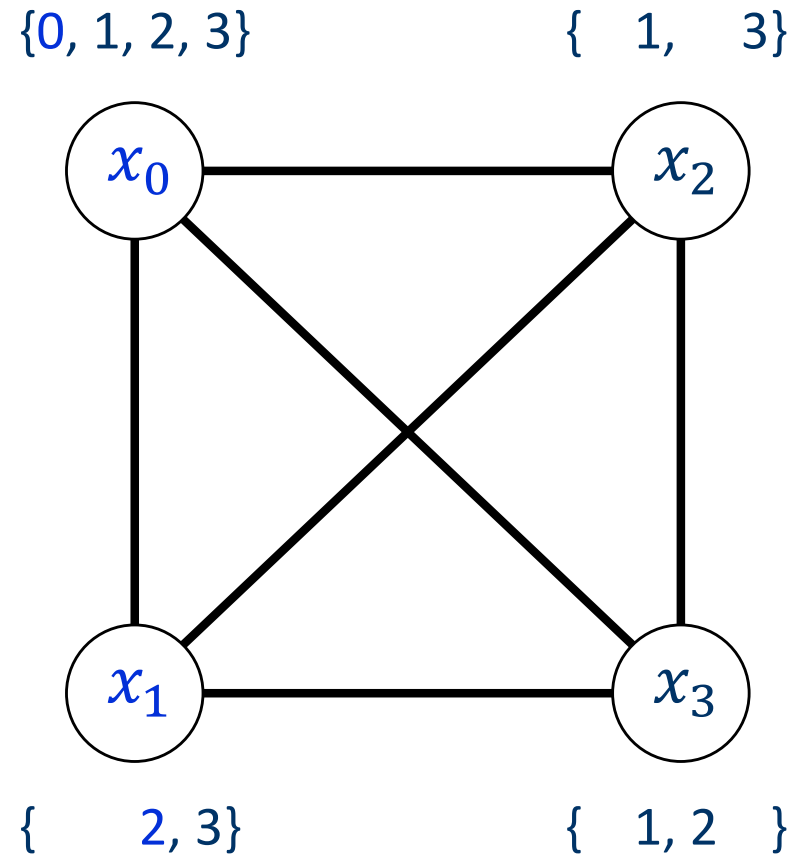
Solving 4-Queens using Forward Checking

	0	1	2	3
0				
1				
2				
3				



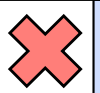










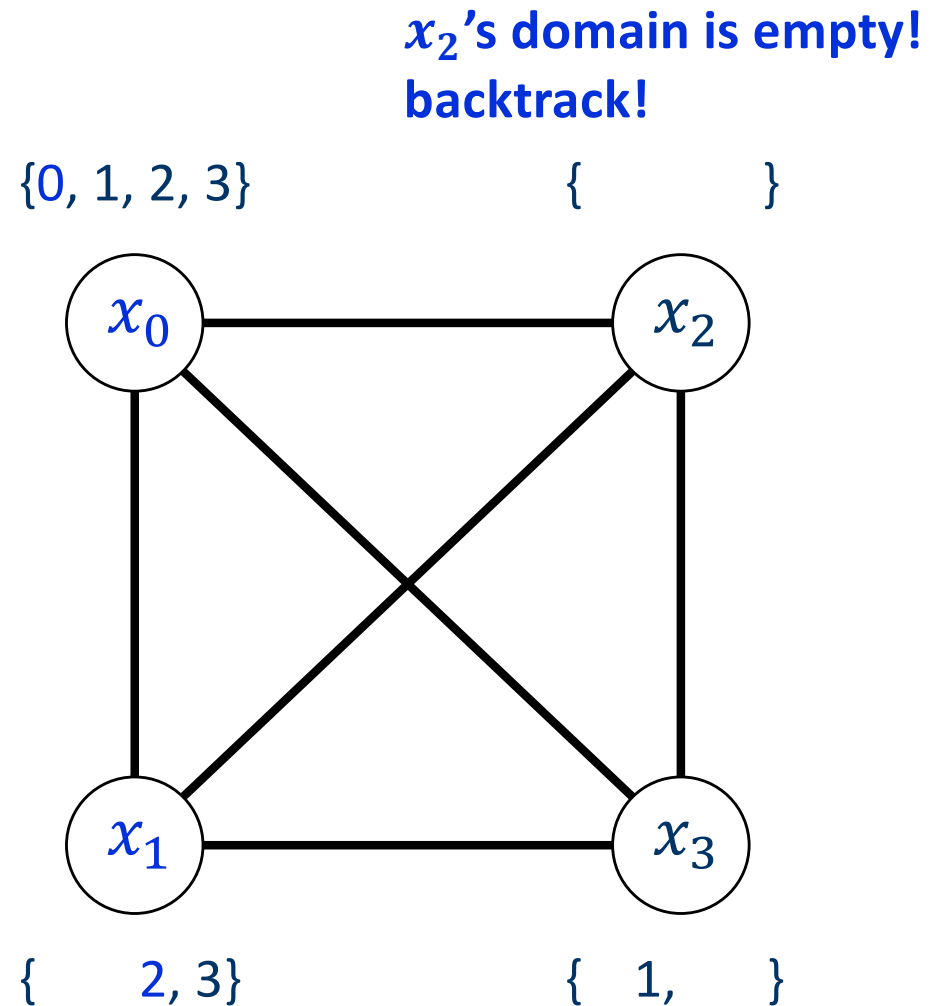
Solving 4-Queens using Forward Checking

	0	1	2	3
0				
1				
2				
3				












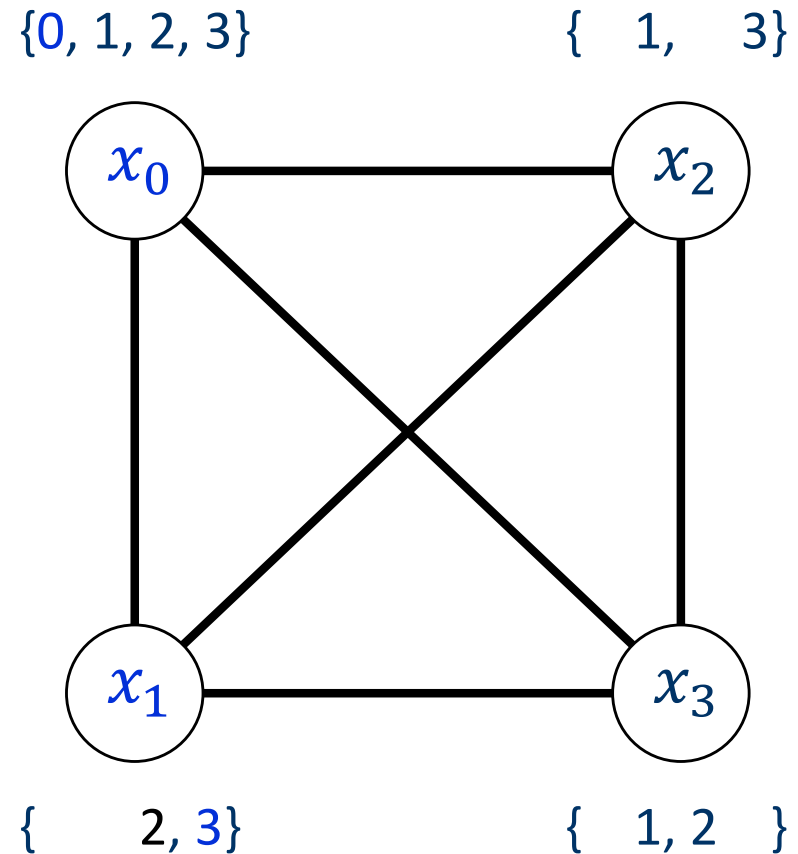
Solving 4-Queens using Forward Checking

	0	1	2	3
0				
1				
2				
3				



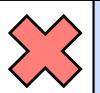




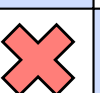





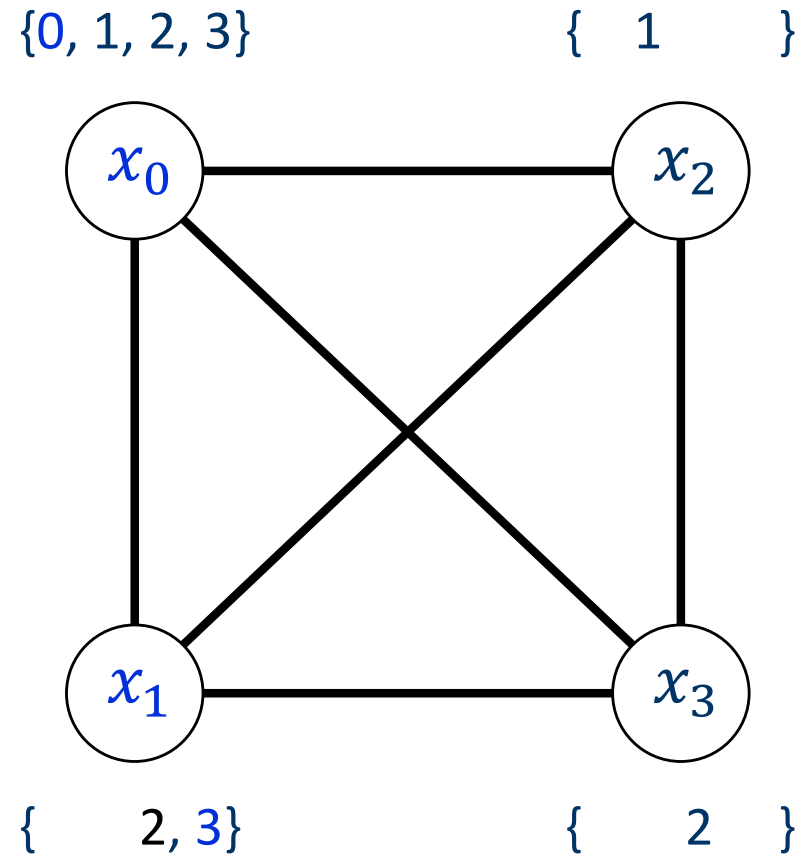
Solving 4-Queens using Forward Checking

	0	1	2	3
0				
1				
2				
3				















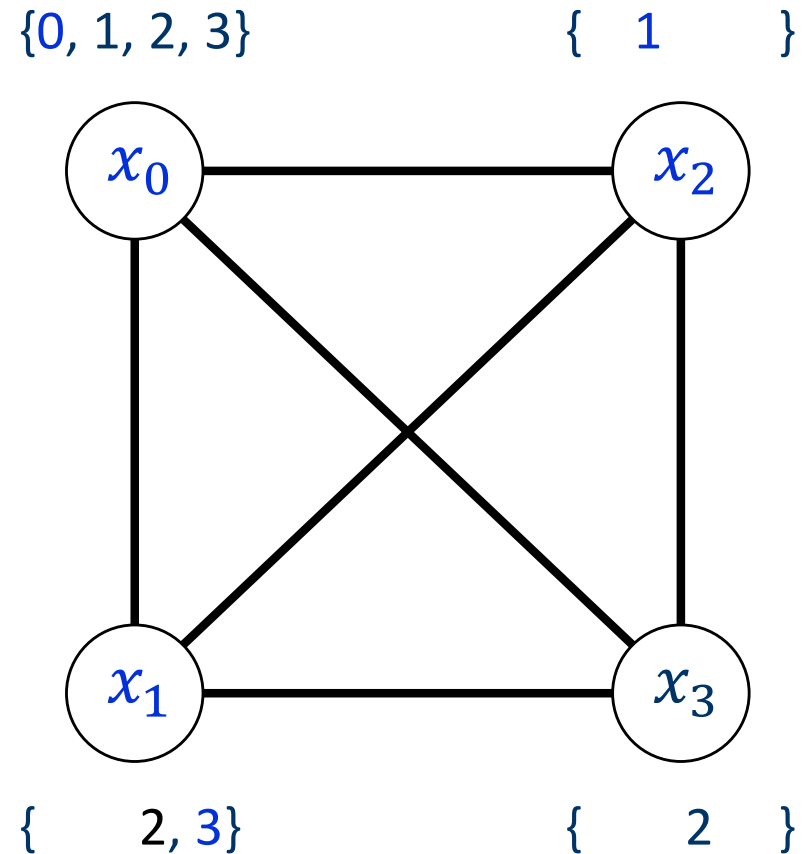
Solving 4-Queens using Forward Checking

	0	1	2	3
0				
1				
2				
3				



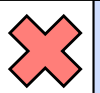












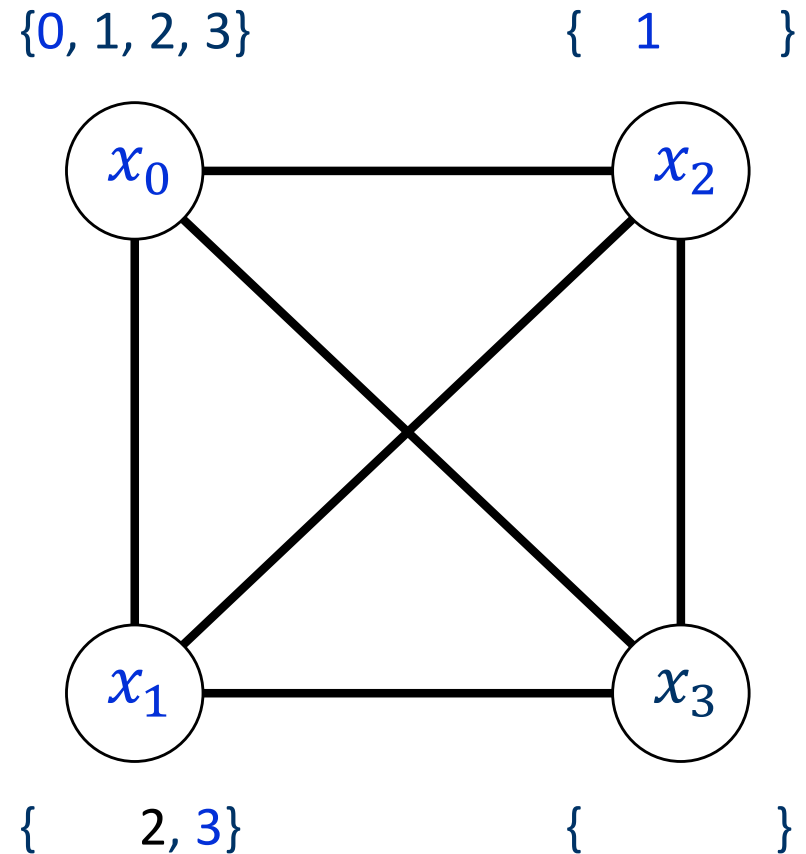
Solving 4-Queens using Forward Checking

	0	1	2	3
0				
1				
2				
3				



Solving 4-Queens using Forward Checking

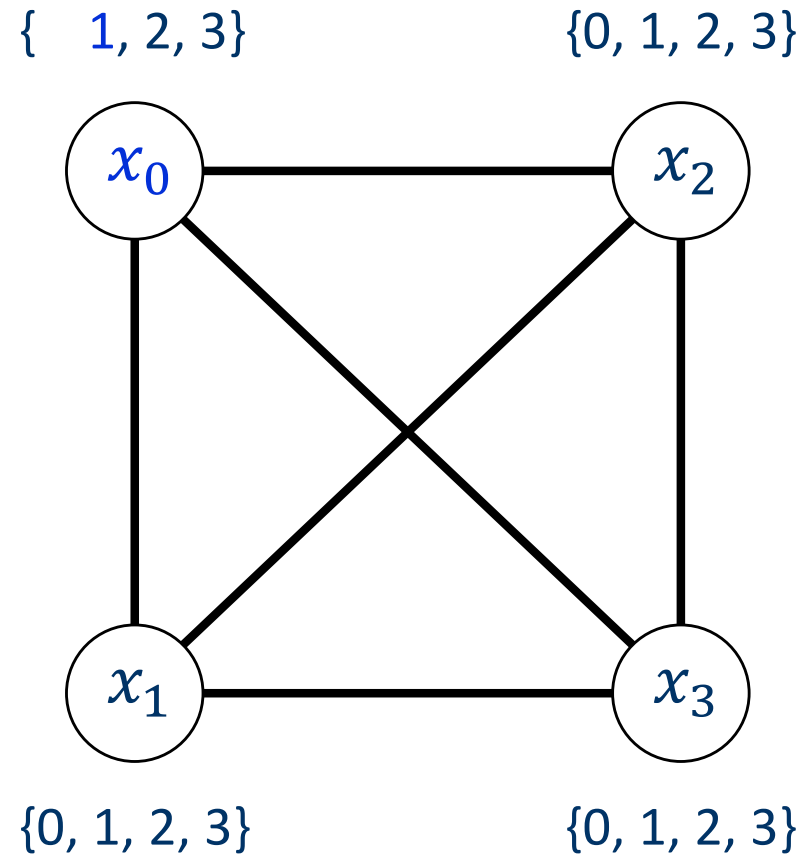
	0	1	2	3
0				
1				
2				
3				



**x_3 's domain is empty!
backtrack!**

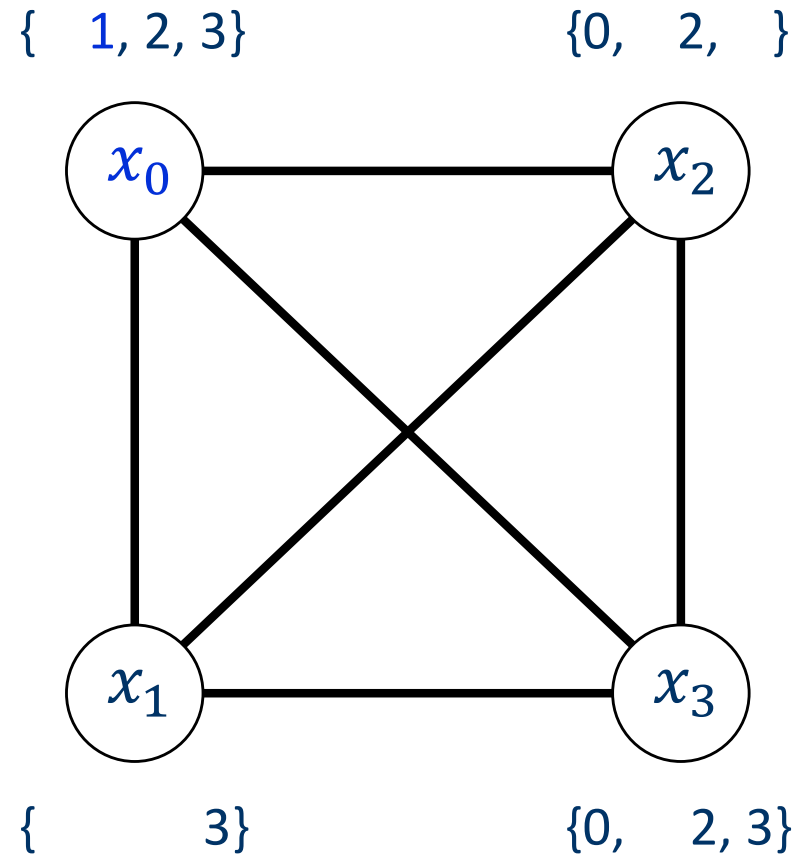
Solving 4-Queens using Forward Checking

	0	1	2	3
0	✖			
1	♔			
2				
3				



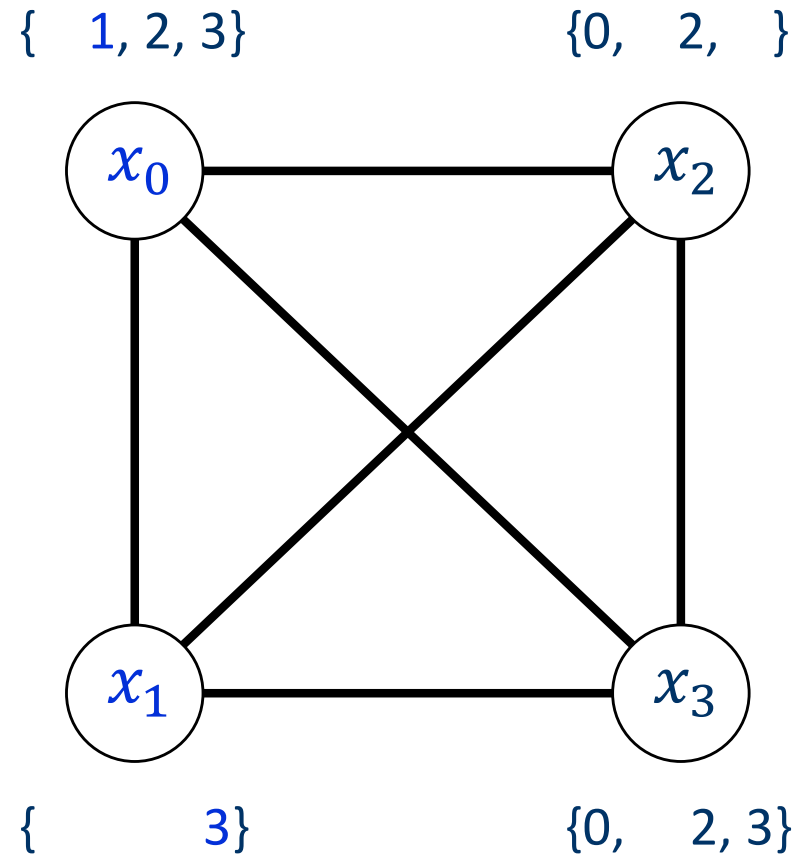
Solving 4-Queens using Forward Checking

	0	1	2	3
0	✖	✖		
1	♔	✖	✖	✖
2		✖		
3			✖	



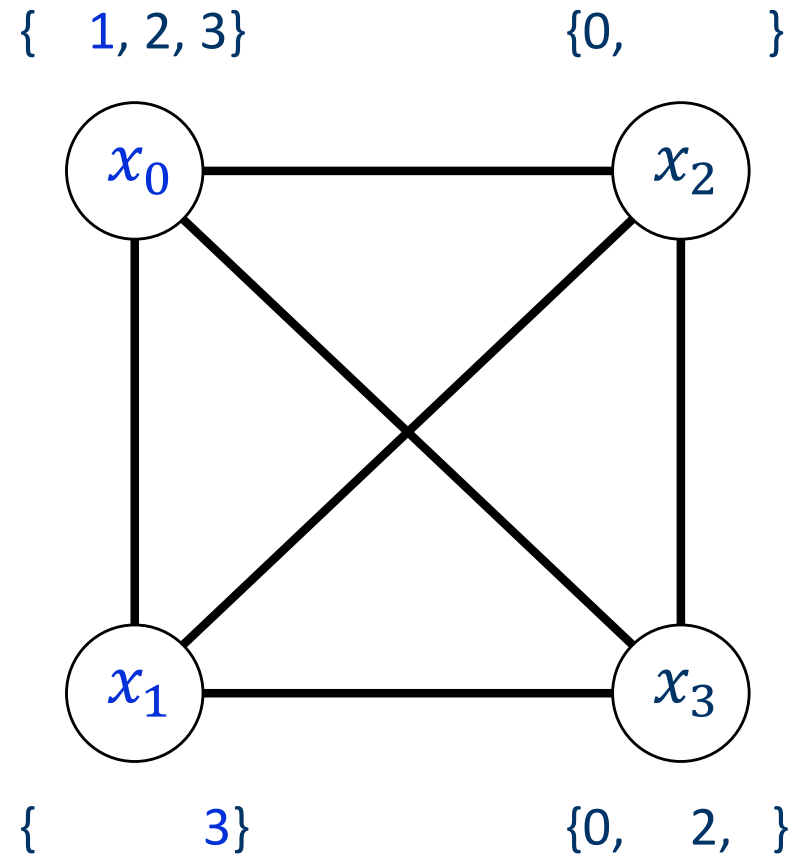
Solving 4-Queens using Forward Checking

	0	1	2	3
0	✖	✖		
1	♔	✖	✖	✖
2		✖		
3		♔	✖	



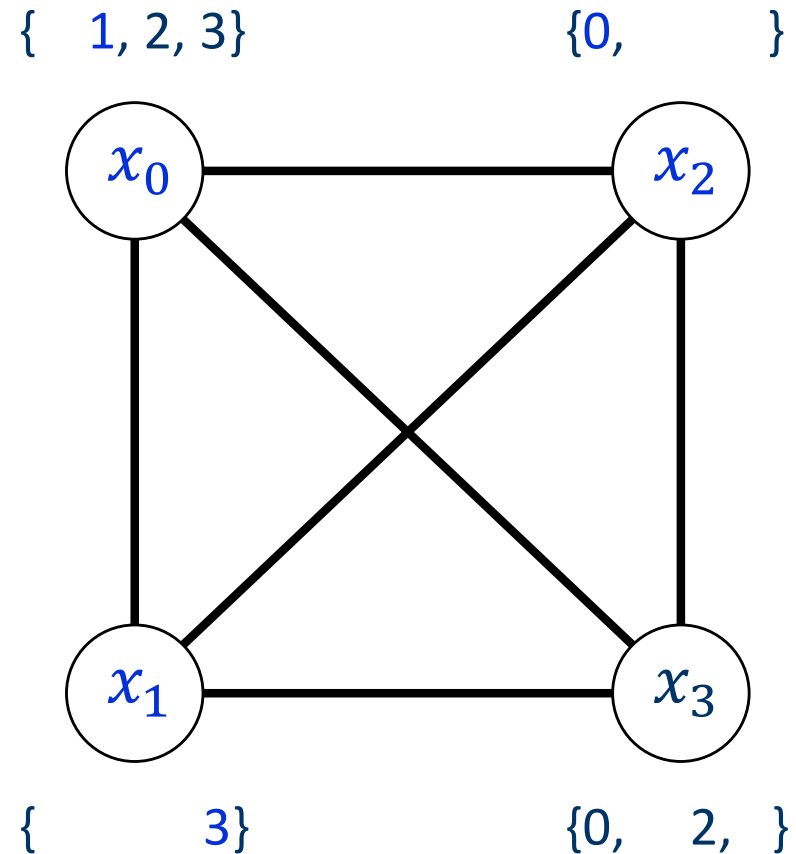
Solving 4-Queens using Forward Checking

	0	1	2	3
0	✖	✖		
1	♔	✖	✖	✖
2		✖	✖	
3		♔	✖	✖



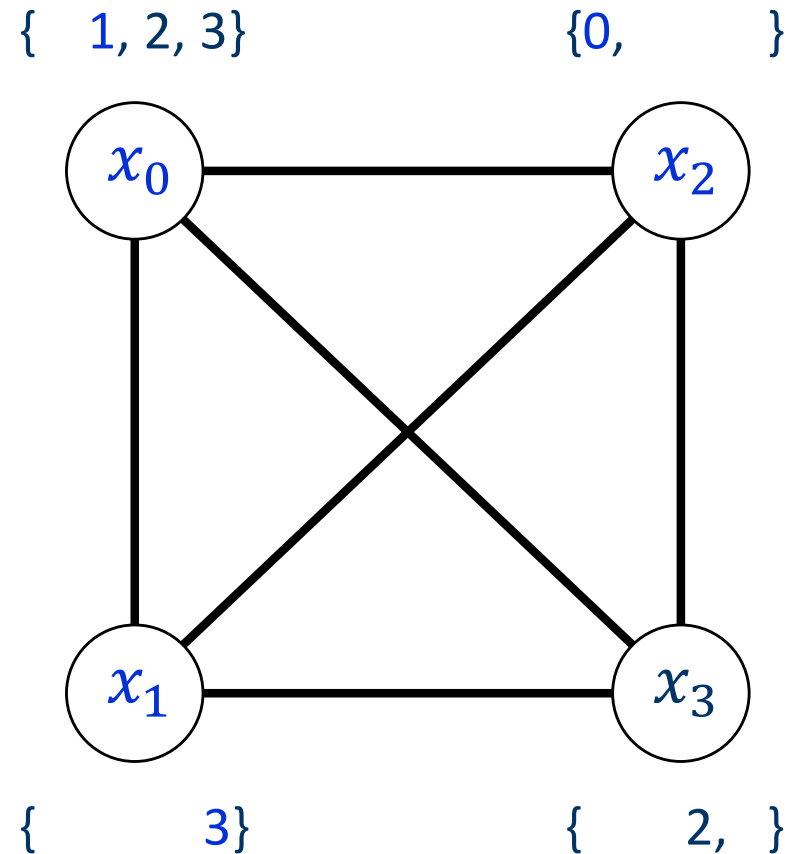
Solving 4-Queens using Forward Checking

	0	1	2	3
0	❌	❌	♔	
1	♔	❌	❌	❌
2		❌	❌	
3		♔	❌	❌



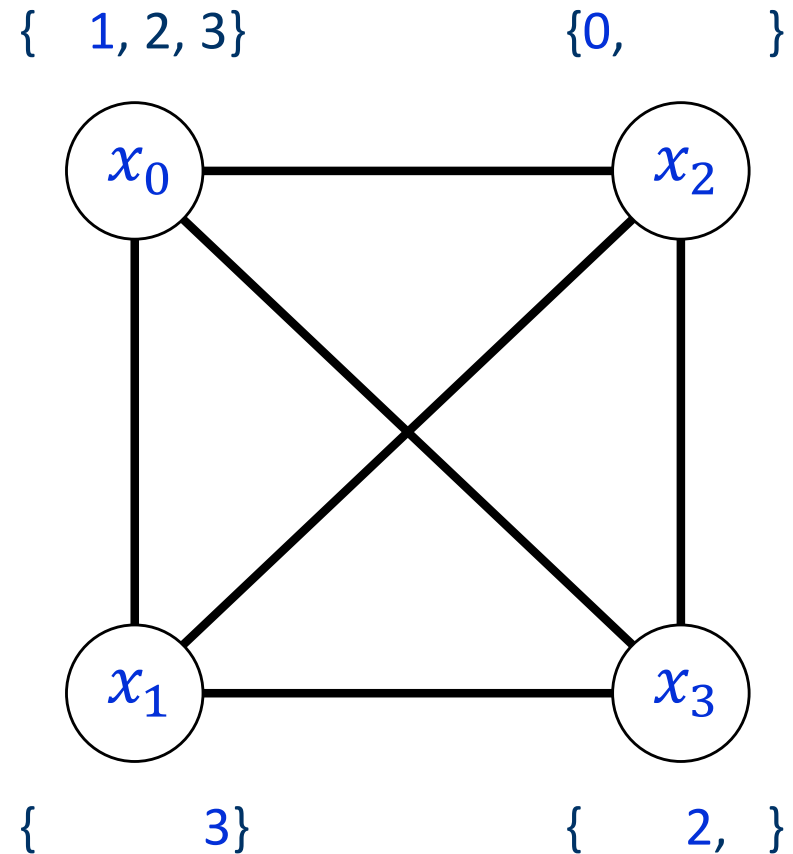
Solving 4-Queens using Forward Checking

	0	1	2	3
0	❌	❌	♔	❌
1	♔	❌	❌	❌
2		❌	❌	
3		♔	❌	❌

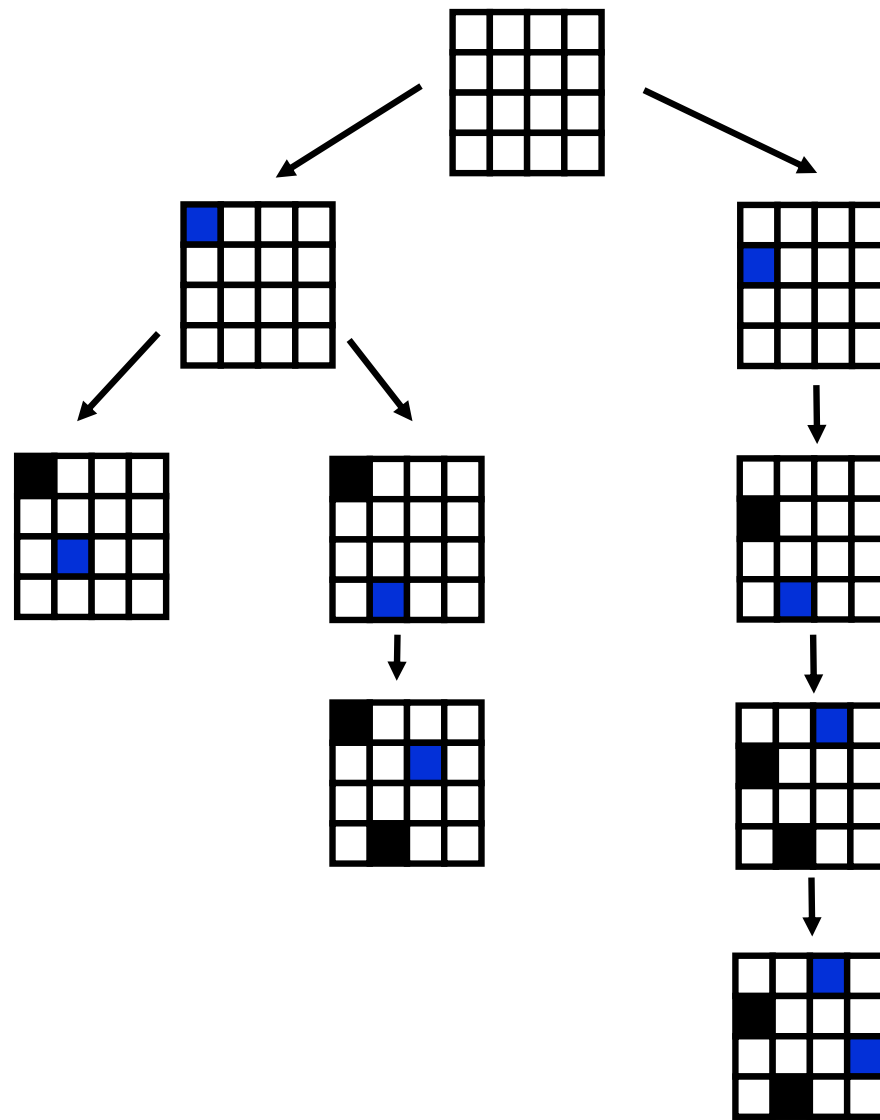


Solving 4-Queens using Forward Checking

	0	1	2	3
0	❌	❌	♔	❌
1	♔	❌	❌	❌
2		❌	❌	♔
3		♔	❌	❌



Solving 4-Queens using Forward Checking

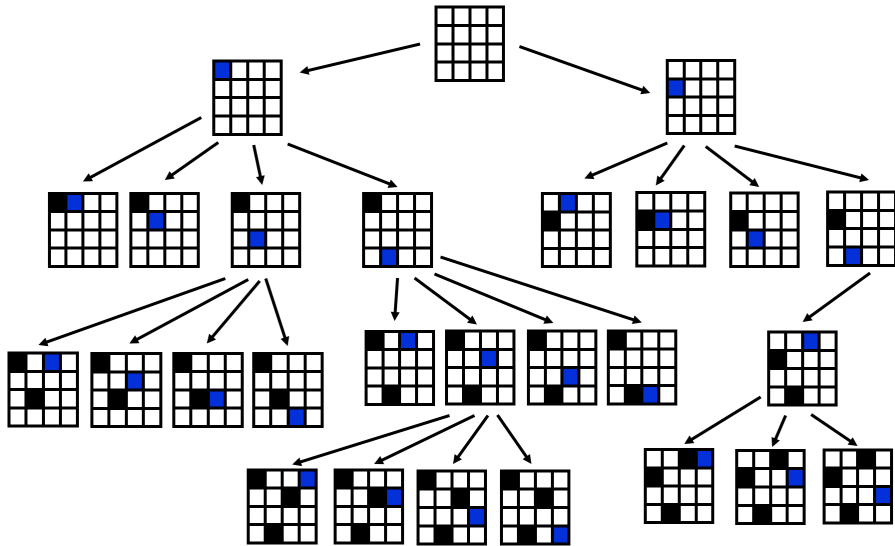


Solving 4-Queens w/ Forward Checking

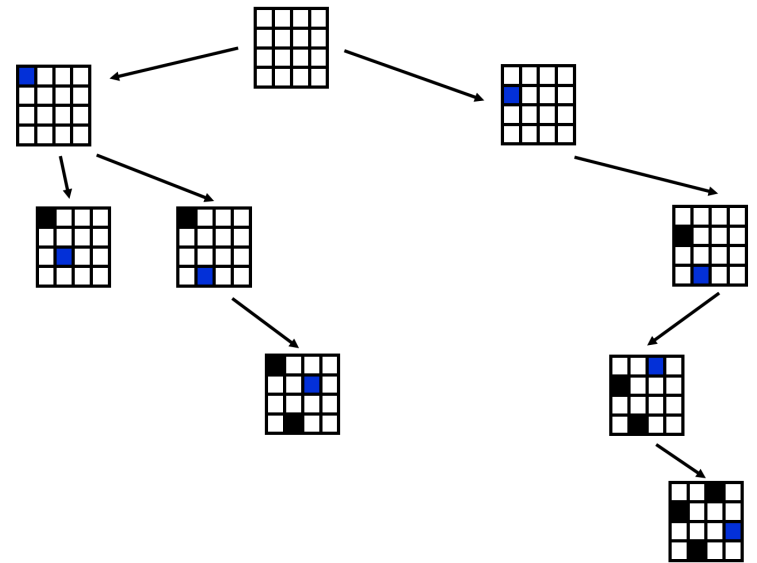
Step	Assigning a Value and Revising Domains				What Next?
	x_0	x_1	x_2	x_3	
1	$x_0 = 0$	2, 3	1, 3	1, 2	Continue
2		$x_1 = 2$	empty	1	Backtrack
3		$x_1 = 3$	1	2	Continue
4			$x_2 = 1$	empty	Backtrack
5	$x_0 = 1$	3	0, 2	0, 2, 3	Continue
6		$x_1 = 3$	0	0, 2	Continue
7			$x_2 = 0$	2	Continue
8				$x_3 = 2$	Solution Found!

Backtracking versus Forward Checking

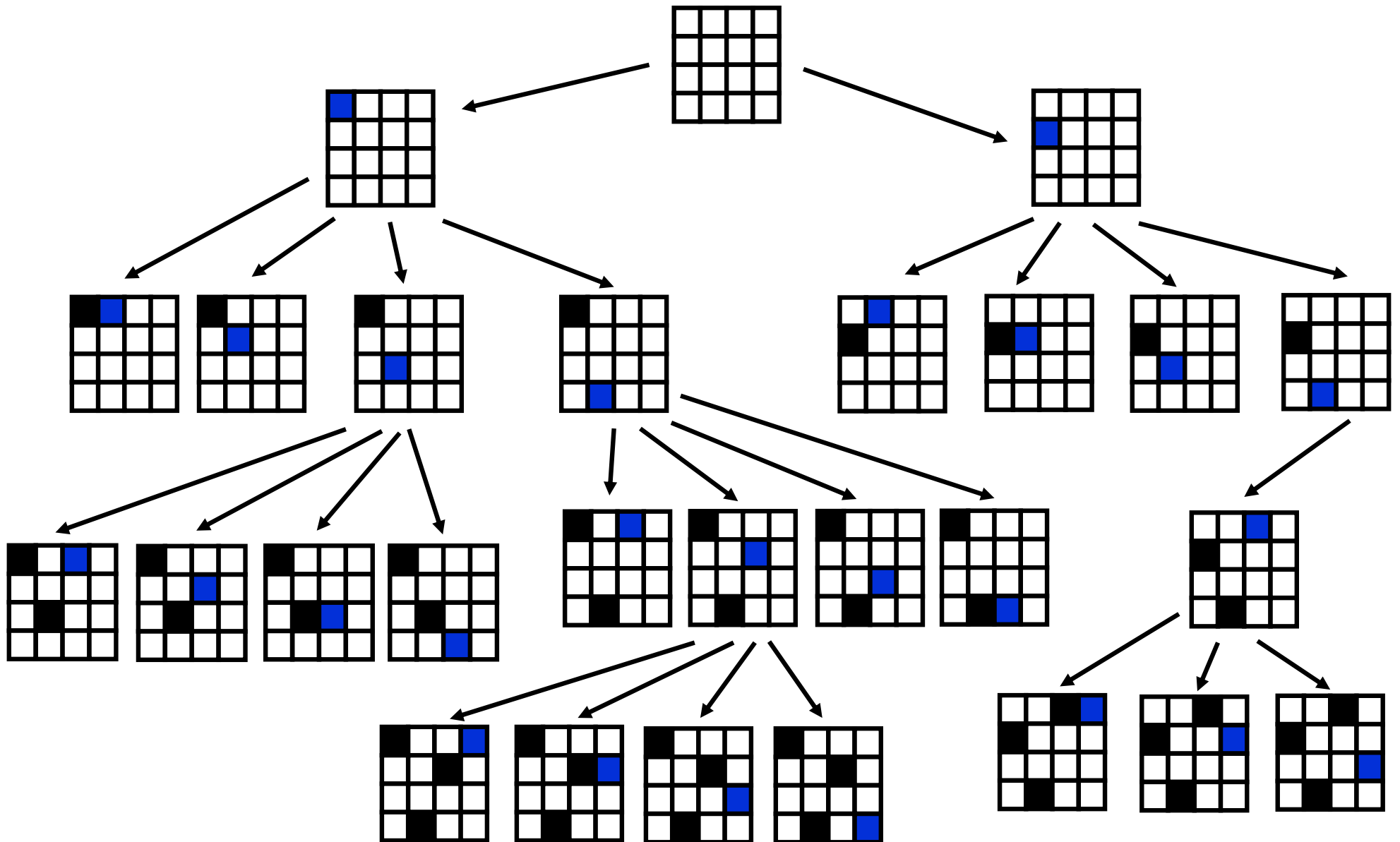
Backtracking Search Space



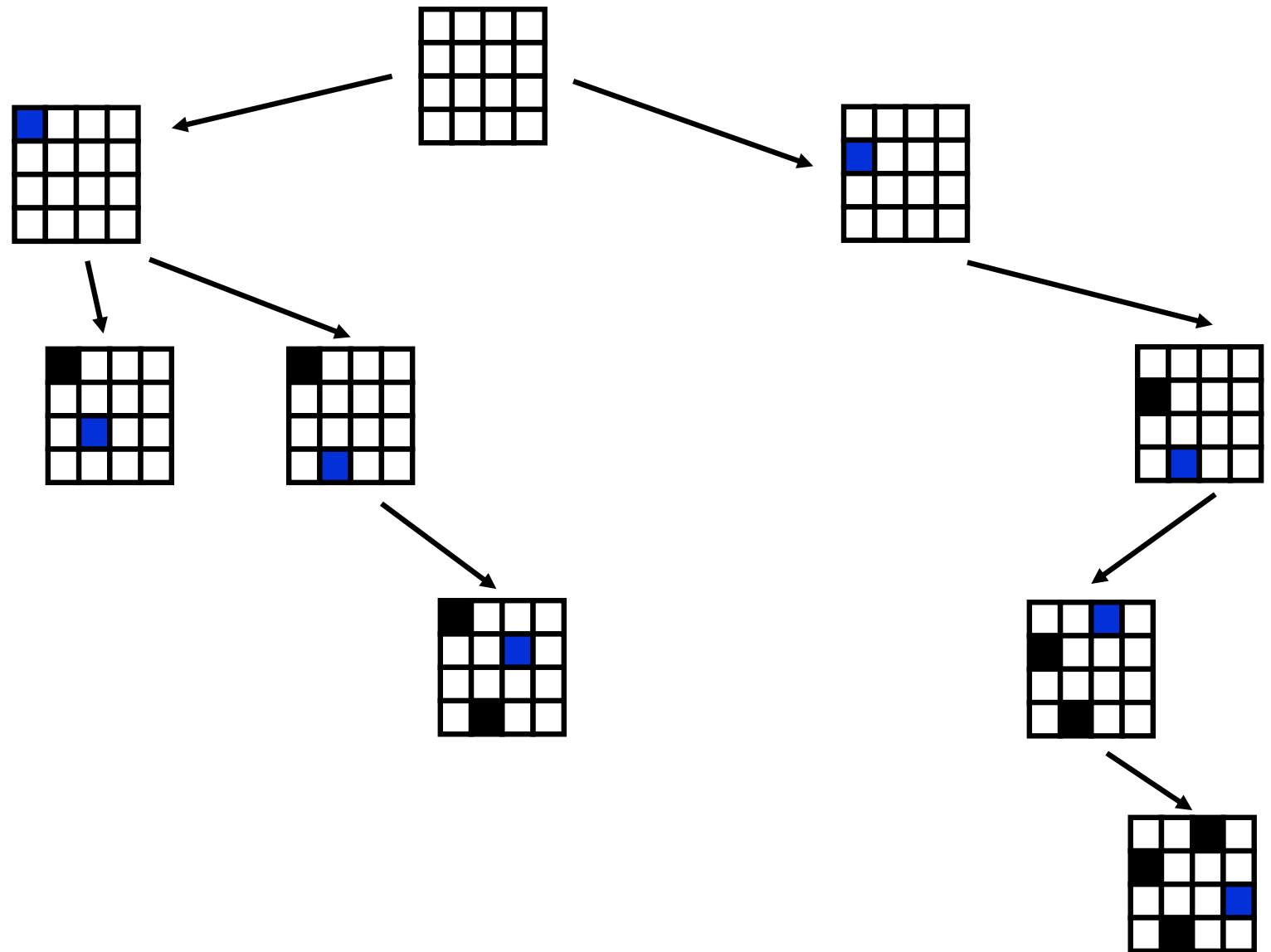
Forward Checking Search Space



Backtracking Search Tree

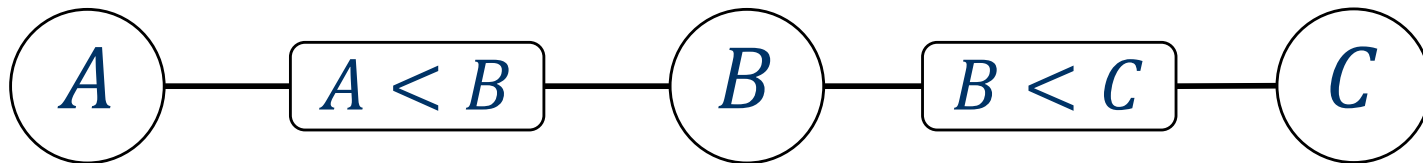


Forward Checking Search Tree



Extra Example 1

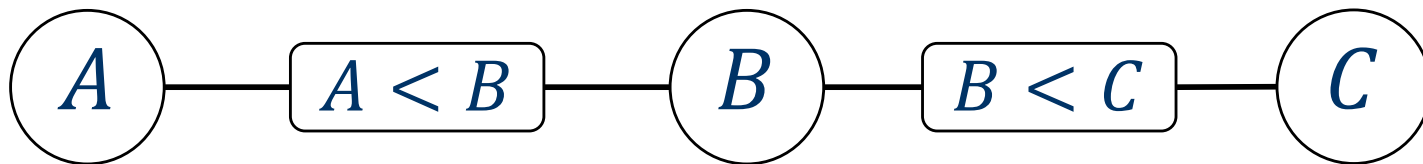
Consider the CSP below. Each variable's domain is $\{1, 2, 3, 4\}$.
Solve the CSP using Backtracking Search and Forward Checking.
For each variable, consider the values in **increasing** order.



Step	Assigning a Value and Revising Domains			What Next?
	A	B	C	
1				
2				
3				

Extra Example 1

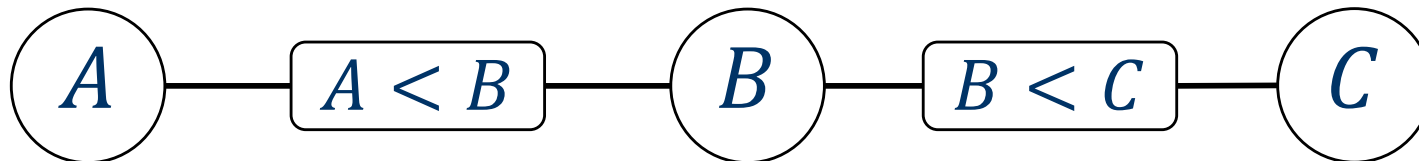
Consider the CSP below. There are three variables A , B , C . The domain of each variable is $\{1, 2, 3, 4\}$. Solve the CSP using Backtracking Search and Forward Checking. For each variable, consider the values in **increasing** order.



Step	Assigning a Value and Revising Domains			What Next?
	A	B	C	
1	$A = 1$	2, 3, 4	1, 2, 3, 4	Continue
2		$B = 2$	3, 4	Continue
3			$C = 3$	Solution Found!

Extra Example 2

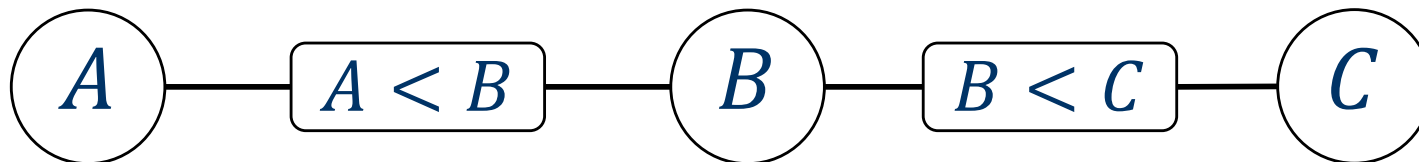
Consider the CSP below. Each variable's domain is $\{1, 2, 3, 4\}$.
Solve the CSP using Backtracking Search and Forward Checking.
For each variable, consider the values in **decreasing** order.



Step	Assigning a Value and Revising Domains			What Next?
	A	B	C	
1				
2				
3				
4				
5				
6				
7				

Extra Example 2

Consider the CSP below. Each variable's domain is $\{1, 2, 3, 4\}$.
Solve the CSP using Backtracking Search and Forward Checking.
For each variable, consider the values in **decreasing** order.



Step	Assigning a Value and Revising Domains			What Next?
	A	B	C	
1	$A = 4$	empty	1, 2, 3, 4	Backtrack
2	$A = 3$	4	1, 2, 3, 4	Continue
3		$B = 4$	empty	Backtrack
4	$A = 2$	3, 4	1, 2, 3, 4	Continue
5		$B = 4$	empty	Backtrack
6		$B = 3$	4	Continue
7			$C = 4$	Solution Found!