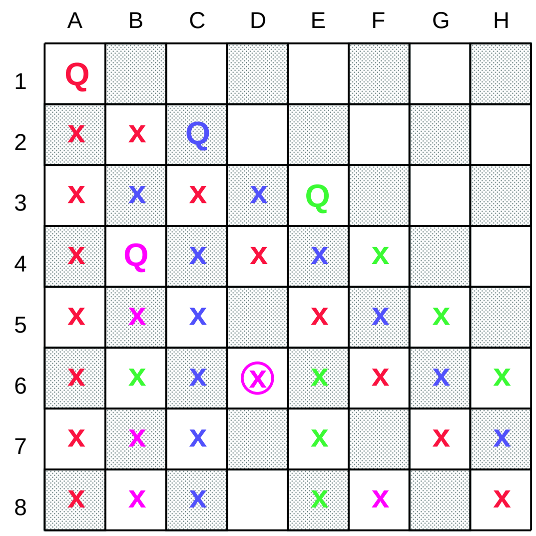
n-Queens Problem

Place n queens in an $n \times n$ chessboard such that no two queens threaten each other.

- *Variables* $x_i, i = 1, \dots, n$ with domain $D_i = \{1, \dots, n\}$ indicating the column of the queen in line i.
- *Constraints*
  - $x_i \neq x_j$, for $1 \leq i < j \leq n$ (vertical)
  - $x_i \neq x_j + (j - i)$, for $1 \leq i < j \leq n$ (diagonal 1)
  - $x_i \neq x_j - (j - i)$, for $1 \leq i < j \leq n$ (diagonal 2)

Forward Checking (2)

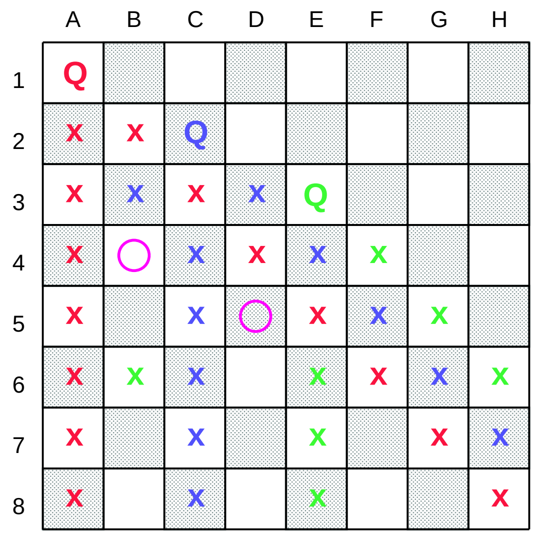
Forward Checking



1A
2C
3E
4BG
5B
6D
5D
4H
5B
6D
7F
6 (no more value)
5D
4 (no more value)
3F

Partial Lookahead (3)

Partial Lookahead

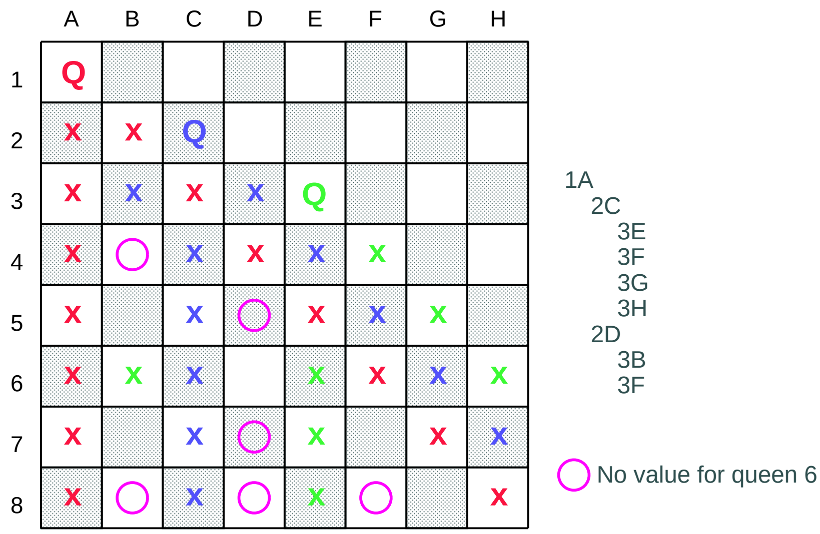


1A
2C
3E (delete 4B and 5D)
4GH
5B (no value left for 6)
3F (delete 6D and 6E)
4BH (failed, backtrack to 4)
3G (delete 5D and 7E)
4B

○ No value for queen 6

Full Lookahead (4)

Full Lookahead



Typical structure of a constraint program

- Declare the variables and their domains
- State the constraints
- Enumeration (`labeling`)

The constraint solver achieves only local consistency.

In order to get global consistency, the domains have to be enumerated.

Labeling

- Assigning to the variables their possible values and constructing the corresponding search tree.
- *Important questions*
  1. In which order should the variables be instantiated (variable selection) ?
  2. In which order should the values be assigned to a selected variable (value selection) ?
- Static vs. dynamic orderings
- *Heuristics*

Dynamic variable/value orderings

- Variable orderings
  - Choose the variable with the smallest domain *"first fail"*
  - Choose the variable with the smallest domain that occurs in most of the constraints *"most constrained"*
  - Choose the variable which has the smallest/largest lower/upper bound on its domain.

- Value orderings
  - Try first the minimal value in the current domain.
  - Try first the maximal value in the current domain.
  - Try first some value in the middle of the current domain.

Some constraint programming systems

System	Avail.	Constraints	Language	Web site
B-prolog	comm.	FinDom	Prolog	www.probp.com
CHIP	comm.	FinDom, Boolean, Linear $\mathbb{Q}$	Prolog, C, C++	www.cosytec.com
Choco	free	FinDom	Java	choco.emn.fr
Eclipse	free non-profit	FinDom, Hybrid	Prolog	eclipseclp.org
Gecode	free	FinDom	C++	www.gecode.org
GNU Prolog	free	FinDom	Prolog	gnu-prolog.inria.fr
ILOG	comm.	FinDom, Hybrid	C++, Java	www-01.ibm.com/software/ integration/optimization/cplex-cp-optimizer/
JaCoP	free	FinDom	Java	jacop.osolpro.com
MiniZinc	free	FinDom Arithmetic		g12.cs.mu.oz.au/minizinc
Mozart	free	FinDom	Oz	www.mozart-oz.org
NCL	comm.	FinDom		www.enginest.com
Prolog IV	free	FinDom, Arithmetic	Prolog	prolog-heritage.org
SCIP	free	Hybrid		scip.zib.de
Sicstus	comm.	FinDom, Boolean, linear $\mathbb{R}/\mathbb{Q}$	Prolog	www.sics.se/sicstus/