Supplement: Encoding the N-Queens Problem with Boolean Formulas*

Randal E. Bryant
Carnegie Mellon University

December 20, 2019

Notice

The material in this document is supplementary material to publication [2]. This document is being made available to the public, subject to copyright provisions. You are free to copy and distribute it, but you must give attribution for any use of this material.

This work was supported, in part, by NSF STARSS grant 1525527.

Overview

In this document, we show how to generate the Boolean formulas for the N-queens problem. The idea is to place N queens on an $N \times N$ chessboard, such that no two are located in the same row, column, or diagonal. To do so, we must both encode the position of the queens on a chessboard in terms of a set of Boolean variables, and then generate a set of formulas over those variables that constrain the positions of the queens.

As notation, for integer x, define $\mathrm{bits}(x) = \lceil \log_2 x \rceil$.

There are two choices for encoding the positions of queens on the board. A *one-hot* encoding uses a Boolean variable for each square. A *binary* encoding uses $\mathrm{bits}(n)$ variables for each row, encoding the position of the (unique) queen within the row. We use the abbreviation BIN to indicate binary encoding, and ONH to indicate one hot.

The naive approach to encoding constraints is to generate formulas for each square of the chessboard, indicating that if it is occupied, then none of the other squares in the same row, column, or either diagonal can be occupied [1]. This requires around $4N$ constraints per square, but since there are N^2 squares on the board, this yields a formula with $\theta(N^3)$ constraints. In this document, we show how to encode the problem with $\theta(N^2)$ constraints.

Preliminaries:

- Row index r numbers the rows from top to bottom with $1 \leq r \leq N$.
- Column index c numbers the columns from left to right with $1 \leq c \leq N$.

Boolean Variables

ONH $v_{r,c}$ for $1 \leq r, c, \leq N$.

BIN $x_{r,i}$ for $1 \leq r \leq N$ and $0 \leq i < \mathrm{bits}(N)$.

SELECT: Selector Expressions

Define $S_{r,c}$ to indicate the condition when a queen is occupying square r, c.

ONH $S_{r,c} = v_{r,c}$.

BIN Let $\overline{\oplus}$ denote the exclusive-nor operation and $\mathrm{bin}(c, i)$ denote the value of bit i in the binary representation of integer c.

$$S_{r,c} \quad = \quad \bigwedge_{i=0}^{\mathrm{bits}(N)-1} x_{r,i} \,\overline{\oplus}\, \mathrm{bin}(c, i) \qquad\qquad 1 \leq r, c \leq N$$

UNIQUE: One Queen Per Row

Define H_r to indicate that row r contains exactly one queen.

ONH Define $E_{r,c}$ to indicate that the row up through column c is empty, and $H_{r,c}$ to indicate that the row up through column c contains a single queen for $1 \leq r \leq N$:

$$\begin{array}{rcll} E_{r,0} & = & 1 & \\ H_{r,0} & = & 0 & \\ E_{r,c} & = & E_{r,c-1} \wedge \overline{S}_{r,c} & 1 \leq c \leq N \\ H_{r,c} & = & [S_{r,c} \wedge E_{r,c-1}] \vee \left[\overline{S}_{r,c} \wedge H_{r,c-1}\right] & 1 \leq c \leq N \\ H_r & = & H_{r,N} & \end{array}$$

BIN Define $H_{r,c}$ to indicate that the row up through column c contains a single queen for $1 \leq r \leq N$:

$$\begin{array}{rcll} H_{r,0} & = & 0 & \\ H_{r,c} & = & S_{r,c} \vee H_{r,c-1} & 1 \leq c \leq N \\ H_r & = & H_{r,N} & \end{array}$$

CONFLICT: Conflict Avoidance

We build up the following expressions for each position r, c:

$L_{r,c}$	The downward-left diagonal starting at position r, c is unoccupied.
$D_{r,c}$	The downward column starting at position r, c is unoccupied.
$R_{r,c}$	The downward-right diagonal starting at position r, c is unoccupied.

Boundary conditions:

$$\begin{array}{rcll} L_{N+1,c} & = & 1 & 0 \leq c \leq N-1 \\ D_{N+1,c} & = & 1 & 1 \leq c \leq N \\ R_{N+1,c} & = & 1 & 2 \leq c \leq N+1 \\ L_{r,0} & = & 1 & 2 \leq r \leq N \\ R_{r,N+1} & = & 1 & 2 \leq r \leq N \end{array}$$

Recurrence, for all $1 \leq c \leq N$, starting with $r = N$ and working up to $r = 2$:

$$\begin{array}{rcl} L_{r,c} & = & L_{r+1,c-1} \wedge \overline{S}_{r,c} \\ D_{r,c} & = & D_{r+1,c} \wedge \overline{S}_{r,c} \\ R_{r,c} & = & R_{r+1,c+1} \wedge \overline{S}_{r,c} \end{array}$$

ROW: Each Row is Valid

Expression V_r specifies that row r has one queen, and that it has no conflicts in the rows below, for $1 \leq r \leq N$:

$$V_r \quad = \quad H_r \wedge \bigwedge_{c=1}^{N} \left[\overline{S}_{r,c} \vee (L_{r+1,c-1} \wedge D_{r+1,c} \wedge R_{r+1,c+1}) \right]$$

Overall Correctness

The overall validity requires that every row is valid:

$$V \quad = \quad \bigwedge_{r=1}^{N} V_r$$

Table. 1 shows an accounting for the number of Boolean operations in the expressions. These only show the terms that are at least quadratic in N.

With a one-hot encoding, we see that the formula has size $\theta(N^2)$. With a binary encoding, the asymptotic formula size is $\theta(N^2 \log N)$. Note, however that the largest benchmark we have ever successfully completed has $N = 16$, and that $N = 27$ is the largest value for which the total number of solutions is known [3]. It is therefore reasonable to assume that $\log_2 N$ will not exceed 5 for the forseeable future, and hence the total number of operations is effectively $O(N^2)$ for both one-hot and binary encodings.

Constraint	Binary	One-Hot
SELECT	$\mathrm{bits}(N) \cdot N^2$	0
UNIQUE	N^2	$4N^2$
CONFLICT	$3N^2$	$3N^2$
ROW	$4N^2$	$4N^2$
Total	$(8 + \mathrm{bits}(N)) \cdot N^2$	$11N^2$

Table 1: Operations counts for encoding N-queens problem

References

[1] Henrik Reif Andersen. An introduction to binary decision diagrams. Technical report, Technical University of Denmark, October 1997.

[2] R. E. Bryant. Chain reduction for binary and zero-suppressed decision diagrams. *Journal of Automated Reasoning*, 2020.

[3] Thomas B. Preußer and Matthias R. Engelhardt. Putting queens in carry chains, No. 27. *Journal of Signal Processing Systems*, 88(2):185–201, 2017.