N-QUEENS PROBLEM

1. Problem description

This problem is somewhat of a '*hello world*' for constraint programming and involves finding a non-attacking placement of n queens onto a $n \times n$ chessboard.

2. Z Models

We present two schemas for the usual model of having n rows (columns) as variables, whose domains are column (row) numbers. Let n be the number of queens, given as a non-zero number.

$$n : \mathbb{N}_1$$

Since the interval $1 \ldots n$ appears several times, we define it as the interval *fields*.

$$\mathit{fields} == 1 \mathrel{..} n$$

2.1. **Basic solution.** The definition of the n-queens schema below is based on a sequence *queens* of fields (columns or rows). In the domain of this sequence, again the same interval reappears.

NQueens

$\mathit{queens} : \operatorname{seq} \mathit{fields}$

$$\begin{aligned}
&\operatorname{dom} \mathit{queens} = \mathit{fields}\\
&\forall i, j : \operatorname{dom} \mathit{queens} \mid i \neq j \bullet\\
&\qquad \mathit{queens}(i) \neq \mathit{queens}(j) \land\\
&\qquad \mathit{queens}(i) + i \neq \mathit{queens}(j) + j \land\\
&\qquad \mathit{queens}(i) - i \neq \mathit{queens}(j) - j
\end{aligned}$$

The remainder of the predicate part contains the familiar horizontal and diagonal non-attack constraints.

2.2. **Advanced solution.** Noting that the domain and range of *queens* are identical and that this sequence is injective (i.e., contains no duplicate values), we can replace the first set of difference constraints by turning the declaration of *queens* into a bijection.[1] This allows the following succinct definition.

NQueens_Bijection

$\mathit{queens} : \mathit{fields} \rightarrowtail\!\!\!\!\!\twoheadrightarrow \mathit{fields}$

$$\forall i, j : \operatorname{dom} \mathit{queens} \mid i < j \bullet j - 1 \neq \mathsf{abs}(\mathit{queens}(j) - \mathit{queens}(i))$$

We use the absolute value function abs, which is defined in the Appendix.

Date: December 2003.

[1]The use of a bijection appeared first in the ALICE variant of the problem [5, p. 86]

3. References

Descriptions and examples can be found in nearly any tutorial on constraint problems, the bibliographic page of Kosters [4] currently mentions more than 88 literature references. A general algorithmic solution for the 8-queens problem is developed in the classical paper on program refinement [9], a PASCAL version later appeared in [10, sec. 3.5]. For a change, a functional programming approach to the problem is published in [8, pp. 241–243]. Due to the inherent symmetries, only a few solutions to the problem are non-isomorphic. This issue has been discussed in [2], where a modification of backtrack-search (symmetry-breaking) to obtain non-isomorphic solutions is also introduced.

Nadel investigated nine different representations of the problem with regard to performance in [6]. He is also named in [3, pp. 45–46] to have invented the *confused n-queens problem*, where the objective is to place n queens on an $n \times n$ board such that all pairs of queens *do* attack each other. Another variant is the *peaceably coexisting armies of queens problem*, an optimisation problem first posed in [1]. The aim is to place two equally sized but differently coloured armies of queens on a chessboard in such a way that no two queens from opposing armies can attack each other. The optimisation objective is to maximise the size of the armies. As noted in [7], there are 16 symmetries in the problem; the reference discusses 3 CSP models and shows that by remodelling, using SBDS and choosing an appropriate variable ordering a more than 100-fold reduction in solving time is possible.

References

[1] Robert A. Bosch. Peaceably Coexisting Armies of Queens. *Optima (Newsletter of the Mathematical Programming Society)*, 62:6–9, 1999.

[2] Paul Cull and Rajeev Pandey. Isomorphism and the N-Queens problem. *ACM SIGCSE Bulletin*, 26(3):29–36, September 1994.

[3] Rina Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[4] Walter Kosters. n-Queens Web Page. http://www.liacs.nl/~kosters/nqueens.html, 2002. Accessed July 2004.

[5] Jean-Louis Laurière. A Language and a Program for Stating and Solving Combinatorial Problems. *Artificial Intelligence*, 10(1):29–127, February 1978.

[6] Bernhard A. Nadel. Representation Selection for Constraint Satisfaction: A Case Study Using n-Queens. *IEEE Expert*, 5(3):16–23, June 1990.

[7] Barbara M. Smith, Karen E. Petrie, and Ian P. Gent. Models and Symmetry Breaking for Peaceable Armies of Queens. Presented at the ECAI 2002 Workshop W9 on Modelling and Solving Problems with Constraints, 2002.

[8] Åke Wikström. *Functional Programming Using Standard ML*. Prentice-Hall, 1987.

[9] Niklaus Wirth. Program Development by Stepwise Refinement. *Communications of the ACM*, 14(4):221–227, April 1971.

[10] Niklaus Wirth. *Algorithms + Data Structures = Programs*. Series in Automatic Computation. Prentice-Hall, 1976.