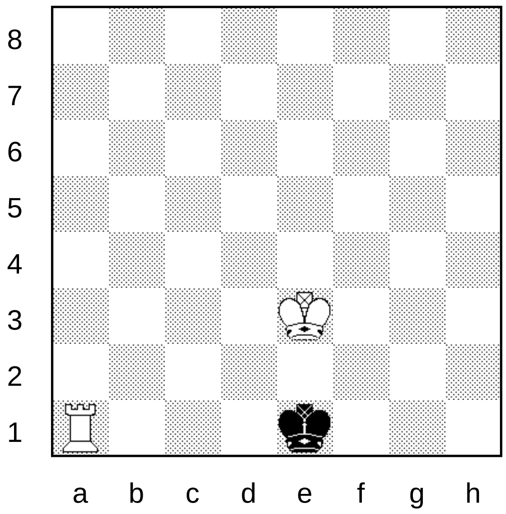
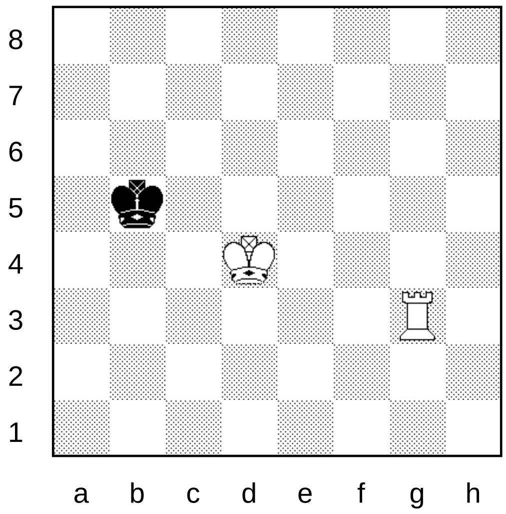
Learning rules from chess databases

Early work on propositional learning from logic databases such as those provided by chess endgames played a key role in highlighting the easy-inverse trick which was later used in developing systems such as Kardio [2]. These databases are good sources of examples for both learning and testing rules. In fact relatively simple chess problems have in the past defeated Machine Learning systems which have otherwise proved rather successful on a range of real-world tasks. The data here concerns one such simple problem: that of the KRK endgame. In this, there are three pieces left on the chess board: White King, White Rook and Black King. The problem of learning rules for recognising illegal positions when it is white's turn to move (WTM) was first proposed in [3]. This has since become a widely accepted test-bed for ILP systems. Examples of a position that is illegal (left) and legal (right) are shown below



illegal(e,3,a,1,e,1).



not(illegal(d,4,g,3,b,5)).

The position on the left is illegal since BK:e1 is in check with WTM. ILP systems usually use the following information to learn the concept:

- Examples are represented by the predicate **illegal/6**. The arguments of this predicate stand for the file (column) and rank (row) coordinates for the WK, WR and BK.
- The background knowledge provided is simply the ordering of rows or columns on a chess board. The predicate **lt/2** tabulates pairs row or column values where one is less than the other. The predicate **adj/2** tabulates row or column values that are adjacent.

Another, more difficult problem is to do with predicting optimal depth of win (that is, the number of moves to checkmate). In the data here, the exhaustive database for the KRK domain acts as a source of example positions, each of which has associated with it optimal depth of win information. Here this is represented by the predicate **krk/7**. The arguments for this predicate stand for depth of win, and the file (column) and rank (row) coordinates for the WK, WR and BK. The typical ILP task is to learn the sub-concept 'black-to-move KRK position won optimally for white in N moves". Preliminary results of classifiers for "won in 0 moves" up to "won in 5 moves", can be found in [1]. The background knowledge used consisted of ground instances of the predicates "symmetric difference" between files and ranks and "strictly less than" ($<$) for all unordered pairs of the file values $\{a, \ldots, h\}$ and all unordered pairs of the rank values $\{1, \ldots, 8\}$.

The data files are held in compressed TAR format. Within the TAR file, they are as used in the original *Golem* [4] experiments. That is, background knowledge files have a ".b" suffix, positive example files have a ".f" suffix, and negative example files have a ".n" suffix.

References

[1] Bain M. and Srinivasan A. (1995). Inductive Logic Programming With Large-Scale Unstructured Data. In D. Michie K. Furukawa and S. Muggleton, editors, *Machine Intelligence 14*. Oxford University Press.

[2] Bratko I., Mozetic I. and Lavrac N. (1989). *Kardio: a Study in Deep and Qualitative Knowledge for Expert Systems*. MIT Press, Cambridge Mass.

[3] Muggleton S.H., Bain M.E., Hayes-Michie J. and Michie D. (1989). An experimental comparison of human and machine learning formalisms. *Proceedings of the Sixth International Workshop on Machine Learning*

[4] Muggleton S.H. and Feng C. (1990). Efficient induction of logic programs. *Proceedings of the First Conference on Algorithmic Learning Theory*, Tokyo.