NOTES

KBBK SQUEEZED INTO A MICRO

Hans Zellner
Mallersdorf, West Germany

Further to my previous note in Vol. 9, No. 4, pp. 199-200 on compressing KRK, KQK and KPK databases so as to fit a Commodore C64, I now report on a similarly successful compression of the 4-piece endgame KBBK onto a non-extended version of this machine, effectively having only 50 Kbytes of available memory. This note deals with the various measures adopted to economize storage space.

1. Reducing the number of positions to be stored

Up to 64^4 (= 16,777,216) positions may result from a 4-piece endgame. When trying to reduce this number, one first remarks that positions are drawn whenever the two Bishops are of the same colour. Bishops of different colours are each limited to 32 squares. Moreover, by symmetry, a 10-square triangle constrains the BK, leaving $64 \times 32 \times 32 \times 10 = 655,360$ positions to store.

Even so, there is still some redundancy. Let the constraining triangle be h1-h4-e4; then, whenever the BK is on the h1-e4 diagonal, at least one additional symmetry constraint may be imposed. For instance, the position BK: h1, WK: h3, WWB: h5, WBB: h4 is symmetry-equivalent to BK: h1, WK: f1, WWB: d1, WBB: e1. We therefore confine, e.g., the WK to be *on or above* the diagonal, saving some 20% percent and reducing the number of positions to be stored to around 525,000 positions.

I refrained on purpose from exploiting further fortuitous symmetries as leading to undue complexities as against meagre savings in storage. For instance, the WK also being on the h1-e4 diagonal (possibly extended to a8) has not been singled out as a special case. The same consideration applies even more strongly to the extreme case of three men finding themselves on that diagonal.

2. Streamlining Database Accesses

As a further space-saving device, the familiar multidimensional-array structure of the database was abandonded for a one-dimensional structure, discarding all illegal positions. Instead of a four-dimensional access method, the single index is now given by a position generator exhaustively enumerating all legal positions in KBBK such that if, primed with the n^{th} position, it will uniquely produce the $n+1^{th}$, $n+2^{th}$, ... legal positions.

While this seems laborious, it eliminates all illegal positions, leaving only 320,680 positions in the database, all defined for White to move (WTM). If Black-to-move (BTM) positions are needed, they are generated by a one-ply search.

3. Reducing the Number of Stored Bits

It has been arranged that no illegal position shall ever be looked up in the database. If one should record mating distances (≤ 19 for KBBK), five bits would suffice. But for an analysis there is no need to differentiate among mating distances: indeed during the analysis no more than one bit is needed (or a 320,680 - sized bit map) initialized to zero. When analyzing at the level mate-in-*n*, the appropriate bit is set to 1, thus distinguishing between positions analyzed and those not analyzed yet.

4. Construction of the Database

4.1. Initialization

All bits in the database are set to 0. The position generator successively creates all legal WTM KBBK positions. Each position gives rise to a two-ply forward search. If a white move leads to a position without a legal move for BK *and* BK is in check, mate-in-1 is concluded. All these positions are stored into a buffer.

4.2. Updating the Database

Every position in the buffer gets a "1" in its corresponding database bit.

4.3. Main Loop

Repeat for *n* from 2
Consult the database:
If the position corresponds to a "0", perform a two-ply forward search and consult the database again for the positions so resulting.
If a "1" is found, it indicates a win for White in *n* - 1 moves at most; if a "0" is found it indicates a win for White in more than *n* - 1 moves or a draw.
If a white move is found whereupon no move for BK is possible avoiding a "1"-position then the position under analysis is a mate-in-*n*. (Captures are no problem because the resulting positions, not being found in the database are rated "0".) All positions so found are stored into a buffer.
If the buffer contains no such positions, 4.3 is terminated.
If the buffer is not empty, proceed as per 4.2 and implicitly increment *n*.

At the conclusion of this process, for which in practice *n* runs up to 19, any database position of value 0 represents a drawn position.

5. Program and Database Sizes

The program was written in assembler language and consists of fewer than 2 Kbytes. The database, i.e. the bitmap, being in memory, requires 41 Kbytes. (= 328,000 bits). This leaves some 7 Kbytes for the buffer which, whenever full, was transferred to the floppy disc and retrieved therefrom as necessary.

6. Playing from the database

For a database to play from (as opposed to the previous bitmap), the recording of mating distances is required. The mating distance is known to be ≤ 19 and thus appears to require five bits per position. As a further space-saving measure, I record only four bits which if ≠ 0000 represent mating distances from 4 to 19 inclusive. If the mating distance so recorded is 0000 it denotes either a draw or a mate in 1 to 3 (inclusive) moves. The latter classes are stored as a bitmap, which is economical because they contain a rather limited number of initial positions.

Thus, the playing database compresses to some 160 Kbytes. Allowing for bootstrap, etc., 120 Kbytes on the floppy are free for other uses. The playing program consults the floppy, addressing it by block, and displays the message 'drawn' or 'mate-in-n', followed, if won, by the optimal white moves and their optimal black countermoves.

THE KBBKN STATISTICS: NEW DATA FROM KEN THOMPSON

H.J. van den Herik and I.S. Herschberg

Delft University of Technology
The Netherlands

Elsewhere in this issue (p. 4), Michie and Bratko exhibit a histogram showing their count of the KBBKN positions by distance-to-win. While their histogram is enlightening, a database result by Ken Thompson allows even stronger quantification. In Table 2 below, *wfreq* counts the number of legal WTM positions starting from which a mate is obtained *or* conversion into a proper subgame is enforced. Each proper subgame (KBBK or KBKN) is known to be won, provided it has arisen from KBBKN. [Note that the general KBKN endgame is not won, but instances arising by optimal play out of a won KBBKN *do* enjoy this property.]

In the subsidiary Table 1, *mfreq* counts the number of positions with the stated distance-to-mate for all proper subgames in the sense defined above. It is not directly comparable to statistics published elsewhere, because it runs over both proper subgames only and therefore fails to be complete for KBBK. Moreover, it also is suspected of containing an unknown proportion of KBKN positions. However, Ken Thompson assures us that the latter are isolated positions not leading to an extended sequence of moves. Furthermore, we note that the results of Table 2, when referring to conversion rather than to direct mate, are silent on the point of entry into Table 1. Specifically, by way of example, a distance of m in Table 2 implies, if conversion is its endpoint, that the distance-to-mate is $m + n$ with $n \leq 19$, but unspecified otherwise.

Finally, we beg readers to note that Table 1 requires most careful interpretation : apart from being possibly contaminated with KBKN, Table 1 is known to be deficient, because it fails to enumerate all KBBK positions, merely counting those which legally and optimally arise out of KBBKN.

Proper subgames of KBBKN won in # of moves	# of positions *mfreq*	Proper subgames of KBBKN won in # of moves	# of positions *mfreq*
1	1580	11	62414
2	652	12	73646
3	3734	13	88690
4	4608	14	90134
5	7668	15	79014
6	9378	16	61518
7	14202	17	27340
8	21554	18	3362
9	31066	19	30
10	45314		

Table 1: *mfreq* counts the number of positions in the proper subgames of KBBKN with their distance-to-mate