5F−3　将棋の先読みをしない１手詰めの計算法

山崎　二三雄，瀬野　訓啓，飯田　弘之，小谷　善行
東京農工大学　工学部　電子情報工学科

１．はじめに

一般に、コンピュータ将棋の詰めルーチンは、先読みを用いたゲーム木の全探索を用いて詰めを読んでいる。普通は１手詰めについても、

①王手を生成し局面を進める
②王側の受け手を生成し局面を進める
③王が取られる位置にあるか判定をする

という枝を作る。しかし、このように１手先読みを行うごとに局面を進めたり戻したりすることは、全体の実行時間を大きくする原因となる。そこで、先読みをしない１手詰めの方法を考察し、詰め将棋システムとして実現をしたので報告する。

２．１手詰め

１手詰めを次の４とおりに分けて考える。

①王の周り８マスのうちの１マスに駒を移動させて（打って）詰める。
②王から２マス以上離れた距離から飛び駒（飛車、角、香）で王手をかけて詰める。
③空き王手をかけて詰める。
④桂馬を移動させて（打って）詰める。

詰んだというのは、王の退路がなく、王手をかけている駒を取ることができない状態のことである（②③の場合、合い駒ができない）。

３．先読みをしない１手詰めの方法

図１のように王の周りのマスをビットアドレスに割り振り、王の退路を１バイトのデータで持つことにする。（ビットが１のとき王はそのマスに移動可能）

例として図２の場合、それぞれ歩と金にはききがないとすると移動できるビットアドレスは、7,6,3,2,0で、退路のデータは11001101となる。

5	3	0
6	王	1
7	4	2

図１．王の周りのﾋﾞｯﾄｱﾄﾞﾚｽ

歩	王	
		金

図２．退路データの例

(1)①の場合

王の周りのマスで守りの駒（王以外の駒）のききがなく、攻めの駒（王手をする駒以外）が１つ以上きいているマスがなければ詰まない。以下、このマスを◎で表すことにする。

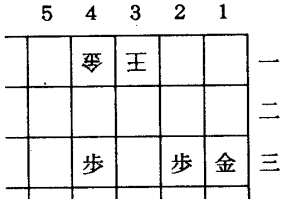
例えば、図１のビットアドレス２の位置で王手をかけることができる駒の持つビットパターンは表１のようになる。

ビットパターンとは、王手をかける駒のききのあるビットアドレスを０、それ以外のビットアドレスを１としたものである。

実際の局面での王の退路のビットパターンと王手をかける駒の持つビットパターンのＡＮＤをとって０ならば詰んだことになる。

表１．◎の位置と王手駒のビットパターンの例

◎の位置	王手駒	ﾋﾞｯﾄﾊﾟﾀｰﾝ
5 3 0 / 6 王 1 / 7 4 ◎	金・成り	11101001
	銀	11111001
	角	11011011
	龍	11101000
	馬	11001001



①◎の位置は座標２二。つまりビットアドレスの２
②王手をかけることのできる駒は金か歩成り（と金）
③王手をかける駒の持つﾋﾞｯﾄﾊﾟﾀｰﾝは11101001
④王の退路のビットは、00010010
⑤11101001 AND 00010010 = 00000000
⑥この局面では２二金か２二歩成りで詰みである

図３．①の１手詰めの例

(2)②の場合

考え方としては(1)とほぼ同じであるが、王の受け手に合い駒があると詰まない。また、龍で王から２マス離れたところで王手をかける場合もほかの場合とは違ってくる。一例を、表２に記載する。表中の●は、駒がなく守りの駒のききのないマスである（◎で駒のないマス）。

(3)③の場合

空き王手を大別すると以下のようになる。

(a)両王手がかかる（飛び駒＋普通の駒）
(b)　〃　（飛び駒＋桂馬）
(c)　〃　（飛び駒＋飛び駒）
(d)飛び駒で王手のみの空き王手

(a)(b)の場合、(1)(2)の組み合わせで詰めを読むことができる。つまり、飛び駒のビットパターン、もう１つの王手駒のビットパターン、王の退路のビットパターンの３つのＡＮＤをとり０の手を生成する。

(c)の場合、合い駒の計算などを省き、(a)(b)と同様に求めることができる。

(d)の場合、間の駒の移動で王の退路のビットパターンが変化する場合、そのための規則を追加することで(2)と同様に詰め手を生成することができる。

Computation of One Move Checkmate without Looking Ahead in Shougi Game

Fumio YAMAZAKI, Kunihiro SENO, Hiroyuki IIDA, Yoshiyuki KOTANI
Tokyo University of Agriculture and Technology

表2．飛び駒による1手詰めのビットパターン

王手の形	ﾋﾞｯﾄﾊﾟﾀｰﾝ
5 3 0 6 王 1 7 ● 2 龍	01100011
・飛、龍で王とX座標が一致したとき	11100111
・飛、龍で王とY座標が一致したとき ・香	10111101

(4)④の場合

④の場合、桂を移動する（打つ）位置に守りの駒のききがなく王の退路がないとき詰みである。

また、桂を取ると飛び駒で王手がかかってしまうとき（図4）は、飛び駒と王の間にある守りの駒は、移動すると王手のかかってしまうマスへのききを消す、という処理をすれば上と同様に詰みを読むことができる。図4の場合、銀のききは3三と1一である。

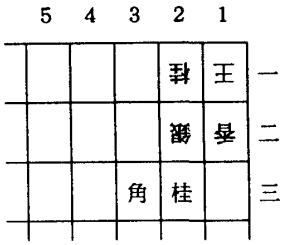


図4．桂馬による詰みの例

4．詰め将棋システムとしての効果

3手詰め以上の詰めを読む場合にも、先読みを行うノードのかなりの数を1手詰めルーチンが読むことになる。

例えば、図5の3手詰めを縦型の全探索による詰めルーチンで解いたとき、図6のようなゲーム木が生成されたとする。

このとき、詰みを得るまでに読む末端ノードの数は、多いとき28、少ないとき（()のノードを一番最初に読んだとき）18である。また、仮に1三銀を最初に読んだとしても末端ノードの数は、多いとき12、少ないとき3である。先読みを行わない1手詰めルーチンでは、この末端ノードを生成しないため、全体としてかなりの実行時間を短縮することができるわけである。

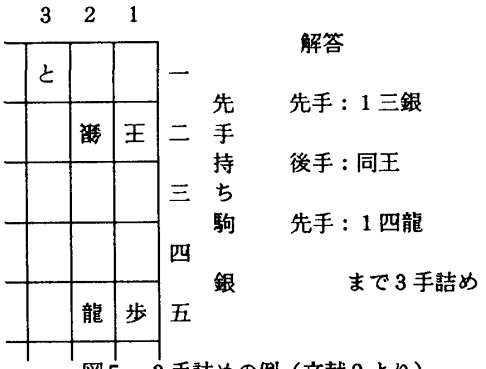
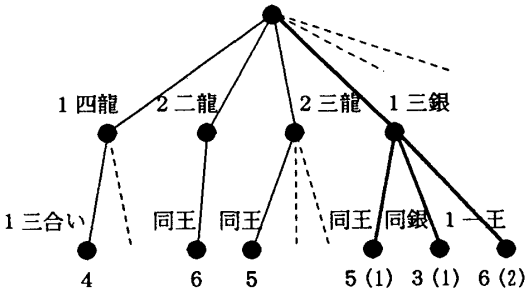


図5．3手詰めの例（文献2より）



（下の数字が末端ノード数。()の数は詰みのノード数）

図6．図5の問題のゲーム木

5．評価

縦型全探索による詰めルーチンと、末端ノードを読む部分をこの方法に置き換えた詰めルーチンとの比較評価を行った。使用機種はＳｕｎ４、システムの記述言語はPrologである。また、評価は参考文献2の中から3手詰めの問題23問、5手詰めの問題25問について行った。結果を表3に記載する。

表3．平均実行時間（秒）

末端ﾉｰﾄﾞの読み	3手詰め	5手詰め
先読みあり(全探索)	4．30	13．25
先読みなし(本方法)	2．56	8．80

6．まとめ

先読みを行わず、局面を静的に評価することで1手詰めを計算することができた。また、その実行速度が期待どおりのものであることが証明された。

謝辞

本原稿執筆にあたり貴重なご意見を頂いた本学滝口伸雄助手、本学野瀬隆技官に感謝します。

参考文献

1．小谷善行，吉川竹四郎，柿木義一，森田和郎，　著
コンピュータ将棋，サイエンス社，1990
2．塚田正夫　著
塚田（九段）の詰将棋，高橋書店，1977
3．小谷善行　（編）
コンピュータ将棋協会資料集　vol.1〜vol.3
コンピュータ将棋協会(1988〜1990)