

ゲーム Hex における必勝手順の検証プログラム

櫻井 英俊, 三島 健, 吉元 昭裕, 野下 浩平

電気通信大学情報工学専攻

{sakura-h, mishim-k, yoshim-a, noshita}@igo.cs.uec.ac.jp

概要

ゲーム Hex は、先手必勝手順の存在が証明されているが、具体的な必勝手順を求めることは長年の研究課題である。これまでに示されている必勝手順は、ユニオン連結による盤面分割が多用されている。今後大きな盤面における必勝手順を導くためには、人の手で盤面分割を行い、コンピュータでその細部の検証を行うなどの共同作業が必要になる。そこで、本稿では、連結の判定を行うプログラム（プログラム A と B）を作成した。プログラム A は、通常の and-or 探索に Hex 特有の技法を導入し、探索効率を向上させたものである。プログラム B は、プログラム A に、仮想準連結に基づく連結判定法などを組み込んだものである。両プログラムの性能評価のために、野下の示した 7×7 と 8×8 の必勝手順の正しさを検証した。その結果、両方のプログラムにおいて、極めて短時間で検証を終えることができた。さらに、人間の熟練者と同じ程度に正確な盤面解析を行っていた。本稿のプログラムは、必勝手順の検証・発見に充分実用的な性能を持っているといえる。

Testing Connectivity for Verifying Winning Strategies in Hex

Hidetoshi Sakurai, Ken Mishima, Akihiro Yoshimoto and Kohei Noshita

Department of Computer Science, The University of Electro-Communications

Abstract

In the game of Hex several mathematical techniques have been proposed for constructing explicit winning strategies. If we try to construct them for relatively large boards of 8×8 , 9×9 or more, we want to combine those techniques with computer programs for human-computer cooperation. This paper presents two programs A and B for testing whether, for a given position, a desired union-connection can be attained. Program A is based on AND-OR searching with a transposition table, which is enhanced by Hex-dependent move-selection algorithms. Program B is further improved by incorporating, among others, virtual semi-connections. We evaluate the efficiency of those programs in terms of the running time as well as the number of searched nodes. For the set of benchmark problems, we have selected about forty hard positions from Noshita's winning strategies for 7×7 and 8×8 . Our computing experiment shows that the correctness of those positions has been verified very quickly in that our programs can be used for practical purposes. The quality of the move-selection (ordering and preclusion) can be evaluated to be as accurate as human expert players.

1 はじめに

ゲーム Hex は J. Nash より $N \times N$ の盤面において先手必勝手順の存在が証明された。しかし、具体的な手順を示すことは長年の研究課題である [5]。 7×7 以下の大きさの盤面においては、仮想連結 [1] による盤面分割の方法を用いて必勝手順が求められている [7]。2004 年野下は、ユニオン連結 (union-connection) の概念を導入し、これに基づいて 8×8 (初手 54) に対する完全な必勝手順を示した [6]。2006 年には三島 [8] が σ 連結と σ 拡張を提案し、これによって 8×8 (初手 63) と 9×9 (初手 55) の十分に簡潔な必勝手順を

示した。このように現在では、 9×9 に対する必勝手順が人の手で導き出せる範囲に入ってきている。

コンピュータを用いた必勝手順の研究も行われている。2000 年に B. Enderton がコンピュータによって 6×6 の全ての初手における勝利プレイヤーを示した [2]。2005 年には R. Hayward が 7×7 の全ての初手における勝利プレイヤーを示した [4]。

大きな盤面における必勝手順を導くためには、人の手で盤面分割を行い、コンピュータでその細部の検証を行うなどの共同作業が重要になってきている。2006 年に R. Hayward はコンピュータによる必勝手

順検証用のプログラムを作成し、2001年にJ. Yangが示した7×7の必勝手順 [7] の正しさを検証をしている [3]。

今後 (様々な初手における) 8×8や9×9の必勝手順や、10×10以上の大きな盤面において必勝手順が導かれていくことが予想される。その際、人間の手によって分割された盤面に対して、(コンピュータなどによって) 石同士の連結性を証明することは必須になってくる。また、この部分は、計算量という点において大きな部分を占める。そこで、本稿では、ユニオン連結の成否を出力する連結判定のプログラム (プログラムA、プログラムB) を作成した。さらに、プログラムAとBの性能を評価するため、野下 [6] の示した必勝手順とそこで用いられているユニオン連結の正しさの検証を行った。

プログラムAとBの基本的なアルゴリズムは、ハッシュ表を用いた and-or 探索であるが、Hex 特有の手法を取り入れることによって、探索の高速化を図っている。プログラムA、B共に、着手の順序付けのために、Hex に依存した多くの評価項目を線型結合した評価関数を用いている。各々の探索節点では、着手候補を限定する手法 (先行排除) を用いることによって、探索を行う必要のない着手を取り除いている。また、基本的な仮想連結の連結パターンを登録して、パターンとの照合により、探索の能率向上を図っている。さらに、プログラムBでは、仮想準連結 [1] に基づく連結判定法を組み込んでいる。

プログラムAの有用性を評価するため、実例として、野下の示した7×7の必勝手順の正しさの検証を行う。野下の示した証明木の (後手局面の) 節点には、ユニオン連結に基づいて分割された盤面が付与されている。これらの分割された部分盤面を入力として与えた。その結果、全てのユニオン連結においてその正しさを確かめることができた。入力として与えたユニオン連結の総数は23個であるが、1分未満の計算時間で全ての検証を行うことができた。

次にプログラムAとBの性能比較を行う。そのために、野下の示した8×8の盤面における必勝手順で、主要なユニオン連結の正しさを検証した。プログラムBの方がプログラムAに比べて、概ね短い実行時間で検証を終了している。実行時間の差が顕著に現れたものでは、200倍近い差が現れたものがあった。

計算時間について、本稿のプログラムは、必勝手順の検証・発見の実用に足ると判断できる。また、着手生成の良さについても、着手の順序付けと先行排除ともに、人間の熟練者と同じ程度に正確に行っている。

2 基本概念

この節では、Hex の基本的な事項を説明する。

Hex は2人で行う完全情報ゲームである。正六角形が敷き詰められた盤を用い、先手と後手のプレイヤーが交互に自分の色の石を盤上に置く。先手は黒石、後手は白石を用いる。図1のように、先手は盤の上下を黒石で連結させれば勝利となる。また、後手は盤の左右を白石で連結させれば勝利となる。

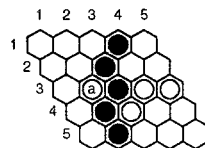


図1: 先手の勝利

本稿では盤上での座標を (x,y) と表現する。x は列の番号、y は行の番号である。例を挙げると、図1の白石 a の座標は $(2,3)$ と表現する。

仮想連結 (virtual connection) [1] とは、空セルの領域 S 内において、相手から着手を行っても、自分の石 x と他の自分の石 y、もしくは端との連結が保証されるという概念である。

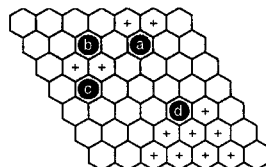


図2: 先手の仮想連結の例

図2は先手の仮想連結の例である。図2の例では石 a と上端、石 b と石 c、石 d と下端が仮想連結によって連結している。石 a と上端の仮想連結、石 b と石 c の仮想連結をいずれも一間跳びと呼ぶ。また、石 d と下端の仮想連結を台形レンマもしくは T-lemma と呼ぶ。図中の '+' で示されたセルは、それぞれの仮想連結に必要な空セルの領域である。

仮想準連結 [1] とは、空セルの領域 S 内において、自分から着手を行えば、自分の石 x と他の自分の石 y、もしくは端との連結が保証されるという概念である。

ユニオン連結 (union-connection) [6] とは、1つ以上の連結の候補を持ち、空セルの領域 S 内において相手から着手を行っても、その連結の候補のうち少なくとも1つが成り立つという概念である。仮想連結は、連結の候補が1つのユニオン連結である。

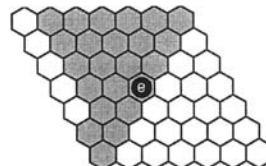


図3: 先手のユニオン連結の例

図3は先手のユニオン連結の例である。図3の例では色のついた領域において石eと上端、もしくは上端と下端のどちらかの連結が成り立つ。また、AB-property[6]や σ 連結と σ 拡張[8]など、ユニオン連結を拡張した概念が提案されている。

先行排除は must-play と呼ばれるプレイヤーの着手を限定する手法である (例えば [4])。その考え方は他のゲームでも古くから知られている。

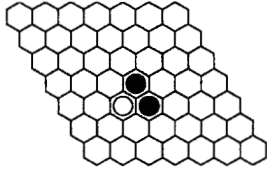


図4: 後手局面

図4を例にとって説明する。この盤面から後手(白)が着手を行うが、その際、探索の能率のために先手が即座に勝利できるような着手を省く。そこで、もう一手先手の石を盤に置き、先手が勝利できるような場合を考える。

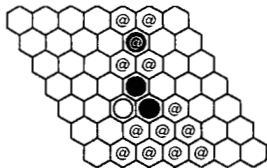


図5: 先手の勝利盤面1

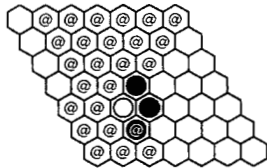


図6: 先手の勝利盤面2

先手が(5,2)に着手を行った場合、図5の '@' で示される領域を用い、先手が勝利する。よって、後手は図5の '@' で示された領域に着手を行わなければならない(それ以外の着手は先手に(5,2)に着手されることによって先手の勝利となる)。

先手が(3,6)に着手を行った場合、図6の '@' で示される領域を用い、先手が勝利することになる。こちらも後手は図6の '@' で示された領域に着手を行わなければならない。

後手は図5と図6の2つの場合に対応しなくてはならないため、これらの図で示された領域のうち、共通する領域のみを着手として残す。それが図7の '@' で示された領域である。

図4の盤面では、先行排除によって図7で示した

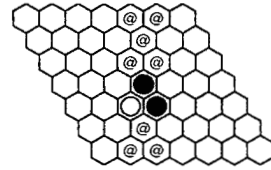


図7: 先行排除により残った後手の着手

ように着手を限定することができる。なお、図7の '@' すべてはもつと凝った先行排除で取り除くことができる。

3 連結判定プログラム

本稿では、2つのプログラムを使用している。それぞれをプログラムA、プログラムBと呼ぶ。

プログラムAとBは、入力として与えられた部分盤面においてユニオン連結が成立しているか否かを判定する。ユニオン連結は複数の連結の候補を持つ可能性があるため、連結目標が複数設定できる。また、連結目標を任意の石や端に設定することができる。

3.1 仮想連結パターンの利用

プログラムAとBでは基本的な仮想連結をあらかじめデータとして登録しておく。目標の連結の成立を判定する際、石が直接接続しているかどうかだけではなく、仮想連結を用いての接続の判定を行う。

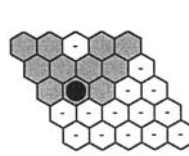


図8: 仮想連結1

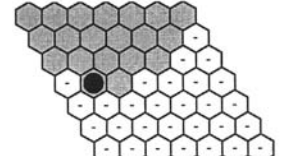


図9: 仮想連結2

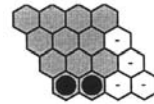


図10: 仮想連結3

図8、図9、図10はいずれも色のついた領域を用い、黒石と上端が連結する仮想連結である。プログラムAとBでは、一間跳びと T-lemma の他に、これらの仮想連結を平行移動、線対称、点対称した位置に対しても適用する。なお、計算時間との兼ね合いを考慮して、これらより大きいパターンは利用していない。

3.2 仮想準連結の利用

プログラムBはAと較べて、いくつかの能率向上の工夫を組み込んでいるが、特に重要なものは仮想準連結の利用である。石の群Xと、石の群Yがあるとす。この時、XとYをつなぐ仮想準連結が2つ

以上あり、それらの領域が重複しない場合、XとYが連結していると判定する。

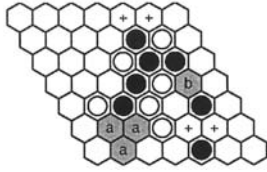


図 11: 仮想準連結を用いた連結判定

具体例を示す。図 11 は後手局面である。この盤面では、領域 a と領域 b の 2 つの仮想準連結が成立している。この後、後手が領域 a に着手を行うと、先手は領域 b に着手を行う。また、後手が領域 b に着手を行うと、先手は領域 a に着手を行う。どちらの場合も先手の勝利となる。よって、上端と下端は連結していると判定できる。

3.3 探索法

探索は and-or 探索によって行う。着手候補を評価関数によって評価し、評価値の高い手から順に着手する。着手候補は空セルに打つ全ての手に対し、先行排除による着手の限定を行ったものである。

Hex は本来引き分けの無いゲームである。しかし部分盤面を扱う場合、両プレイヤーが勝利しておらず石を打てるセルが全く無い状態、つまり引き分けが起りうる。そこで and-or 探索を行うに当たって、引き分けも後手の勝利とする。これにより、2 値の問題として取り扱うことができる。

3.4 評価関数

着手の順序付けを決める評価関数を説明する。着手の評価項目は次の 5 個である。

1. 着手の座標に対する静的および動的な評価。
2. 目標の連結を達成した着手の記憶による評価。キラーリスト技法の変形である。
3. 2 手連続着手して連結を達成できるか否かの評価。
4. 一間跳びの切断を守る着手の評価。
5. ランダムウォークによる着手の評価。例えば深度 5 のランダムウォークを 200 回行い、連結の達成回数を勘定する。

これらの評価項目の値を線形結合し、評価値によって着手の順序を決める。線形結合の係数は計算実験により定める。

4 必勝手順とその検証

プログラム A と B の性能評価のため、Hex の必勝手順の検証を行う。Hex の先手必勝手順を示す証明木 (解の木) とは次のようなものである。

- 後手局面では、着手可能な全ての手を着手する
- 先手局面では、1 つの手のみを着手する

Hex の必勝手順を求めた多くの論文では、必勝手順を簡潔に記述するため、次のような工夫を行っている。

- 先行排除による後手着手の限定
- ユニオン連結等による着手のパターン化

野下が示した 7×7 の盤面の必勝手順の木 (図 19) を例にして説明する。根節点の子には 7 つの先手局面がある。すなわち後手の着手候補が 7 手存在する。しかし根節点の盤面には 48 セルの空きがあるため、着手候補も 48 手存在する。根節点で先行排除を行うと、48 手中 41 手が排除できるため、着手候補から取り除かれている。その他の内部節点の後手局面でも先行排除により、後手の着手候補を限定している。

葉節点の盤面では、実際には上端と下端が連結することが必ずしも明示的に書かれていない。しかしユニオン連結が成立する複数の領域に対する (通常は簡単な) 論理計算により、先手の勝利を示している。

本稿のプログラムを用いた必勝手順の検証方法は次のとおりである。内部節点の後手局面では、先行排除により後手の着手を限定している。そこで、先行排除の正しさを検証する。排除された着手の集合によって定まる (一般には) 複数の盤面それぞれにおいて、後手から着手をはじめて先手勝ちを示す。

葉の節点は後手局面である。この局面では、盤面分割等を用いて先手必勝であることを示している。そこで、まずプログラムを用い、部分盤面内で目標の連結が成立することを確認する。次に部分盤面同士の連結を人手で統合して、その葉が先手必勝であることを示す。

5 性能評価実験

計算実験により、本稿のプログラムの性能評価を行った。使用したコンピュータの CPU は Intel(R) Pentium(R) 4 の 3.06GHz、メモリは 512Mbytes である。

5.1 プログラム A の性能評価

2005 年に野下が示した 7×7 の必勝手順 [6] 全体を、本稿のプログラム A を用いて検証する。この必勝手順 (図 19) では 20 個の葉を含む 23 個の節点に対して検証を行う必要がある。

実験の結果、ほとんどの盤面について、1 秒程度で連結の判定を行うことができた。必勝手順全体が正

しいことは合計1分未満で検証できた。このことはこの必勝手順の簡単さの程度を示していると解釈できる。

各盤面の空セル数と連結の判定にかかった時間(time)、探索節点数(nodes)は表1のとおりである。表の盤面の番号は、文献[6]のものである。表の空セル数とは、プログラムに入力として与えた部分盤面における空セルの数である。一般的な傾向としてはこの値が大きいほどプログラムによる計算量が大きくなる。nodesが0となっている盤面は、探索開始と同時に、仮想連結により勝利が確定した盤面である。

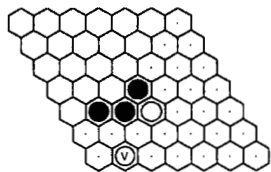


図 12: Figure 29[6]

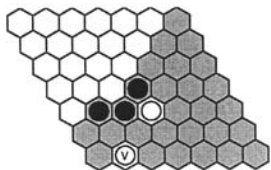


図 13: Figure 29-1

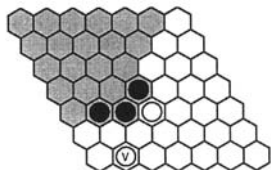


図 14: Figure 29-2

例として、図12における先手の勝利を検証する。この盤面では先手の勝利を示すために、図13と図14のような盤面分割を行っている。そこで、各部分盤面を入力として本稿のプログラムに与え、連結を判定した。その結果、図13、図14ともに1秒で目標の連結が確認できた。これにより、図12の上端と下端が連結することが確認できた。

5.2 プログラムAとプログラムBの性能比較

2005年に野下が示した8×8の必勝手順[6]に使われているユニオン連結の図の一部について、本稿のプログラムAとプログラムBを用いて連結を判定する。それにより、両プログラムの性能を比較する。これらのユニオン連結を選んだ理由は、必勝手順の中で特に難しく、性能評価に適しているからである。

実験の結果、ほとんどのユニオン連結は数分以内で連結の判定を行うことができた。計算時間等は表2

のとおりである。多くの盤面でプログラムBのほうが計算時間が短くなることがわかった。特にFigure 33では大きな差が出た。

プログラムAとBの差が大きかったFigure 33(図15)を例にあげて説明する。

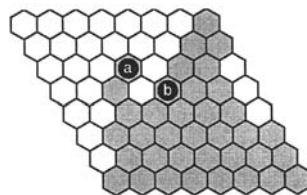


図 15: Figure 33[6]

図15は黒石aと下端、黒石bと下端、もしくは上端と下端が連結するユニオン連結である。この盤面に対し、両プログラムは先行排除によって図16のように後手着手を限定する。'@'で示したセルが先行排除後に残った着手である。この先行排除は人間の熟練プレイヤー(筆者の2人)と同じ結果である。

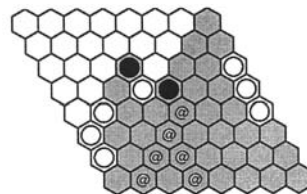


図 16: 先行排除後に残った着手

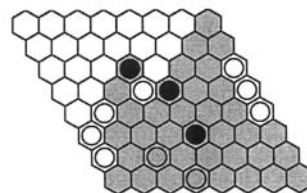


図 17: 図 15 から着手を進めた盤面

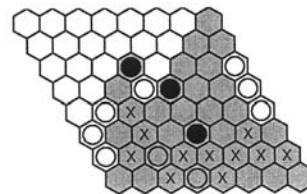


図 18: プログラムBの先行排除

図17は、図15から着手を進めた盤面である。白(3,7)に対して黒(5,6)、白(4,8)と進む。黒は(5,6)を1番目の着手に選択したが、実は正解の着手は(4,6)である。プログラムBでは2番目にこの正解を選んでいる。この盤面の状況においてプログラムAでは先行排除ができず、盤上の空セル全32手の着手が残る。プログラムBでは仮想準連結を組み込んだ部分

がきいて、多くのセルを先行排除できた。その結果、図 18 中の 'X' で示したセルへの着手、11 手のみが黒の着手として残る。この先行排除は熟練者と同等、もしくはそれ以上の結果である。これにより着手 (5,6) が短時間で不正解であると判定できた。図 15 全体では、プログラム A が 56685 秒かかったのに対し、プログラム B は 303 秒で連結を判定できた。このように仮想準連結を用いたアルゴリズムにより、計算時間の大幅な短縮ができた。

表 2 に示す通り、特に難しいと思われるユニオン連結の連結が実用的な計算時間で判定できたといえる。

6 おわりに

本稿では、部分盤面における連結性を判定する 2 つのプログラム、プログラム A と B を作成した。両プログラムでは、本稿でとりあげた規模の盤面に対して人間の熟練者と同程度の着手選択や先行排除ができた。また、計算時間については、8×8 の必勝手順に対して実用的な時間内で連結判定ができるようになった。

プログラム A を用いて、野下が文献 [6] で示した 7×7 の盤面における必勝手順を全て検証することができた。各々の部分盤面は多くても十数秒で検証が終わり、必勝手順全体の検証時間の合計は 1 分未満になった。8×8 の必勝手順は、一部のユニオン連結を検証した。それらのほとんどのユニオン連結は数分で連結を判定することができた。プログラム A では判定に十数時間かかるユニオン連結があったが、プログラム B では 5 分程度で判定することができた。

今後の課題は、もっと大きい盤面に適用するための探索法の改良である。また、人とコンピュータによる共同作業のために、対話的なシステムの開発も課題である。

参考文献

- [1] V. V. Anshelevich, "A Hierarchical Approach to Computer Hex", *Artificial Intelligence*, 134, 1-2 (2002), pp. 101-120.
- [2] B. Enderton, "Hex IAQ", <http://www.cs.cmu.edu/~hde/hex/hexfaq/>.
- [3] R. Hayward, B. Arneson, P. Henderson, "Verifying Hex strategies", *CG 2006* (2006), 10 pp.
- [4] R. Hayward, Y. Bjornsson, M. Johanson, M. Kan, N. Po and J. van Rijswijk, "Solving 7 x 7 Hex with Domination, Fill-in, and Virtual Connections", *Theoretical Computer Science*, 349, 2 (2005), pp. 123-139.
- [5] H.J. van den Herik, J.W.H.M. Uiterwijk and J. van Rijswijk, "Game Solved: Now and in the Future", *Artificial Intelligence*, 134, 1-2 (2002), pp. 277-311.

- [6] K. Noshita, "Union-Connections and Straightforward Winning Strategies in Hex", *ICGA Journal*, 28, 1 (2005), pp. 3-12.
- [7] J. Yang, S. Liao and M. Pawlak, "On a Decomposition Method for Finding Winning Strategy in Hex game", *International Conference on Application and Development of Computer Games in the 21st Century* (2001), pp. 96-111.
- [8] 三島 健, 櫻井 英俊, 野下 浩平, "Hex の必勝手順に対する新証明技法とその応用", *Proc. of 11th Game Programming Workshop* (2006), pp. 134-142.

表 1: 7×7 の検証結果

検証した盤面	空セル数	time(sec)	nodes
Figure 9	20	1	0
Figure 10	20	1	0
Figure 14	12	1	0
Figure 15	22	1	57
Figure 22-1 ¹	18	1	7755
Figure 22-2	18	1	7100
Figure 23	8	1	90
Figure 24	19	1	924
Figure 11	22	1	186
Figure 12	22	1	3293
Figure 13-L ²	18	1	133
Figure 13-R	8	1	9
Figure 17	24	1	9149
Figure 18	16	1	15006
Figure 19	24	1	85
Figure 20	18	1	0
Figure 21	10	1	10
Figure 26	25	16	182448
Figure 27	24	8	119246
Figure 28-L	14	1	609
Figure 28-R	14	1	649
Figure 29-L	19	1	919
Figure 29-R	25	1	279

¹ 盤面の番号の横に-1 や-2 とあるものは、一部の石の配置を変えて検証を行ったものである。

² 盤面の番号の横に-L や-R とあるものは、盤面分割を用いて検証を行ったものである。

表 2: 8×8 のユニオン連結の検証結果

検証した盤面	空セル数	プログラム A		プログラム B	
		time(sec)	nodes	time(sec)	nodes
Figure 32	32	20	38383	158	27284
Figure 33	35	56685 ³	38002570	303	47698
Figure 34	26	4	13225	4	103
Figure 36-L	22	21	22582	6	264
Figure 36-R	23	401	435543	8	3871
Figure 38-L-A ⁴	26	4	4217	8	103
Figure 38-R-A	10	1	595	0	3
Figure 38-L-B	26	3	3143	11	127
Figure 38-L-X	20	10	51181	0	1
Figure 38-R-X	12	1	26	0	3
Figure 39-L-A	25	2	796	2	413
Figure 39-R-A	20	175	679145	0	25
Figure 39-L-B	25	1	295	2	413
Figure 39-R-B	18	84	471356	0	3
Figure 39-L-C	25	3	1688	2	373
Figure 40	34	194	388078	38	708

³ プログラム A の評価関数を調整することにより、検証時間 6351 秒、探索節点数 3700794 で検証を行うことができる。

⁴ 盤面の番号の横に -A, -B, -C, -X とあるものは、AB-property[6] を展開して検証を行ったものである。例をあげると、Figure 38-L-A とは、Figure 38 の AB-property を展開した盤面 A を盤面分割したうちの片側の部分盤面である。

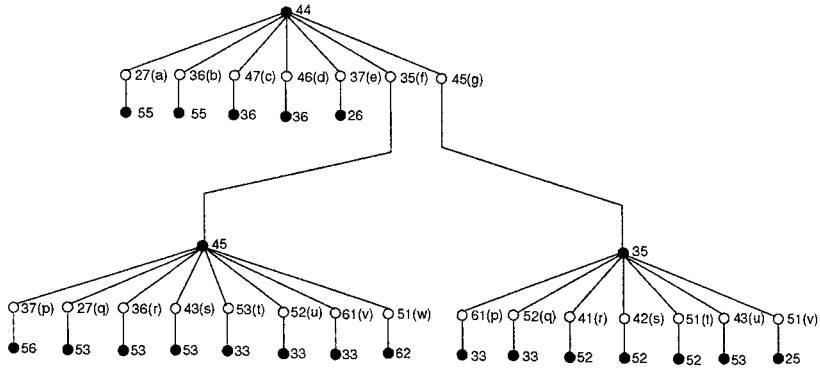


図 19: 野下が [6] で示した 7×7 の必勝手順