

パズルロジックを用いたドット絵画像の二値変換

システム科学技術学部 情報工学科

1 年 金子 和樹

1 年 浜野 伸一

指導教員 システム科学技術学部 情報工学科

准教授 草薙 良至

1. 研究目的

今回私たちはプログラミングとパズルの関係性について興味を持った。プログラミングにはパズルの要素が多く含まれている。この二つにはどちらも論理的に順序立てて考えることが必要とされる。例としてはナンバープレースなどのパズルが挙げられる。パズルの解法アルゴリズムを用いて自動解法プログラムをつくり、そこから役に立つようなプログラムを作ることはできないかと思い研究を行った。

2. ライツアウト

今回の研究で目を付けたのはライツアウトというパズルである。ライツアウトとはタカラから 1995 年に発売されたパズルゲームである。このパズルは 5×5 や 6×6 で点灯・消灯をするライトで構成された正方形の盤面に対して最初に点灯している各ライトをすべて消灯させることを目的としたパズルである。点灯と消灯にはルールがあり、点灯・消灯を反転させるライト(操作するライト)を中心としてその上下左右のライトも同時に反転させる。

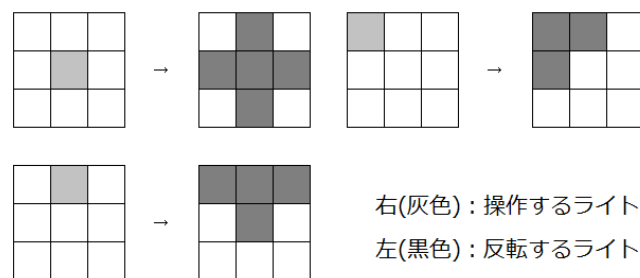


図 1 盤面の点灯、消灯ルール

ライツアウトの特徴としては、同じライトなら二回押しても押したライトは状態が戻るだけで意味がないということと、ライトの押す順番は関係がないということである。これらの理由は、ライツアウトにおいて重要なのは一つ一つのライトの反転の回数であるからである。したがって、ある一つのライトの反転回数が偶数、奇数によって元の状態から変わる、変わらないが決まる。中心ライトの状態変化の条件は中心ライトとそのライトへ影響のある上下左右のライトをいくつ操作したかのみである。よって状態の変化は、各ライトの操作回数の偶奇性で決まり、どの順番で押してもライトの最終的な状態は変わらない。

図 2 にライツアウトの動作例を示す。操作ライトはカッコで示す。ライツアウトは初期盤面, 操作ライトの盤面, 最終盤面をそれぞれ二値行列で表すことができる。ここでの最終盤面はすべての行列の値が消灯(0)となる盤面でありその盤面までに操作したライトの盤面を解行列とする。この研究では解行列を求めること目標とした。

なお、盤面の一边が 5×5 や 9×9 などの特定の時、解行列が複数ある場合や解行列の存在しない場合がある。したがって、この研究では解が一意に定まる盤面について考える。

押す順番を変えることで結果が変わるかの検証 (0 : 消灯 1 : 点灯)											
1	0	0		0	[1]	[1]					
0	0	0		0	0	0					
1	0	0		0	[1]	[1]					
初期盤面				解行列							
解行列適用例 1											
1	[0]	0		0	1	[1]		0	0	0	
0	0	0	→	0	1	0	→	0	1	1	→
1	0	0		1	0	0		1	[0]	0	
									0	1	[1]
									0	0	0
解行列適用例 2											
1	[0]	0		0	1	1		0	1	[1]	
0	0	0	→	0	1	0	→	0	1	1	→
1	0	0		1	0	[0]		1	1	1	
								1	[1]	1	
									0	0	0
結局、操作の開始から終了までそれぞれのライトが何回反転しているかによって最終的なライトの状態が決まるため、反転するライトの位置さえ決まれば操作する過程は問われないのである。											

図 2 ライトを操作する順番について

3. ライツアウトを解くアルゴリズム

まず解行列の性質について述べる。解行列の一行目のみを適用すると最上列のライトの状態が決まる。すべてのライトに解行列を適用する必要があるが、解行列の二列目は最上列のライトを消すために一意に定まる。以下同様にして解行列の一行目が決まるとそれ以降の解行列が一意に定まる。これにより解行列の一行目が決まるまでの過程ですべてのライトについて操作する、しないを検討したこととなる。これを利用して、最下列以外のライトを反復的に消灯させるアルゴリズムを設定した。

まず、最上列で反転させるライトの決め方を説明する。1つ1つのライトについてライトの操作の有無を独立に決め、最上列のライト n 個の操作の全てのパターン、つまり 2^n 通りを総当たりで探索する。以降、 n は盤面の一边のライトの数を示すものとする。

次に、最上列以外で反転させるライトの決め方を説明する。1列目から $(n-1)$ 列目まで順に、「対象の列で点灯している全てのライトの1つ下のライトを反転する」という操作を繰り返すことで、上からライトを消灯させていく。このことで、ライツアウトのルールより対象の列の全てのライトが消灯する。

1 列目から $(n-1)$ 列目まで通して操作することで、最下列以外の全てのライトが消灯する。最下列のライトが全て消灯していた時、その反転操作をライツアウトの解とする。

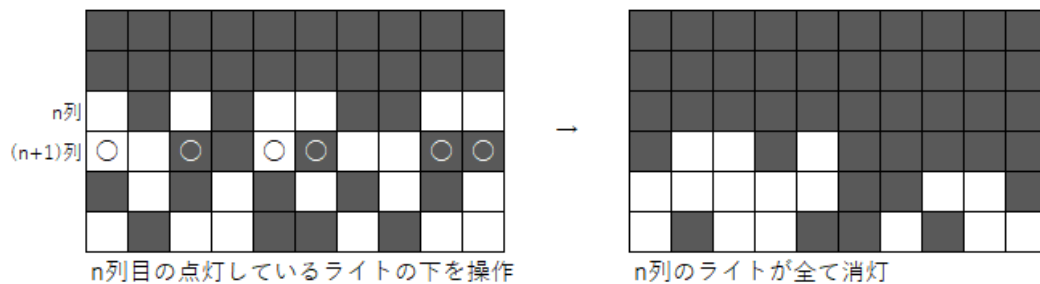


図3 最上列以外の反転させるライトの決め方の例

4. アルゴリズムを応用した画像変換と実行例

ここまでは、一般的な「ライトを全て消灯させる」ルールによるライツアウトについて述べてきた。この手法を応用し、ライツアウトの反転ルールに従って「ライトの点灯、消灯状態を任意に変化させる」ことができる。そのためには、指定された初期盤面と目標盤面について、同じ場所で点灯状態が異なるライトに対して上で述べたような操作を行えばよい。具体的な手順を以下に示す。

まず、最上列で反転させるライトのパターンを決める。ライトの操作パターン 2^n 通りをしらみつぶしに探索する。この手順は、全消灯を目指すときと同じである。

次に、最上列以外のライトを上列から反転させる。1 列目から $(n-1)$ 列目まで順に、「対象の列で、現在の盤面と目標盤面を比較して点灯状態が異なる全てのライトの 1 つ下のライトを反転する」という操作を繰り返す。このことで、対象の列のライトの点灯状態が目標盤面と一致する。1 列目から $(n-1)$ 列目まで通して操作することで、最下列以外の全てのライトが目標盤面と同じ状態になる。最下列も含めて全てのライトが目標盤面と同じ状態になったときに、その反転操作をライツアウトの解とする。これにより、ライツアウトの反転ルールによって、盤面を二値画像（2 種類の色によって構成される画像）と見ることでこれを別の二値画像に変化させることが可能となる。

二値画像の例として、QR コードがある。ライツアウトの反転ルールに従い、ある QR コードを別の QR コードに変化させることを試みた。このプログラムでは最上列のライトの操作をしらみつぶしに決めるため、盤面の一边のライトの数によって探索回数が指数関数的に増加してしまう。そのため、QR コードのサイズは最小規格の 21×21 とした。また、2 つの QR コードの持つ情報を "example_1" "example_2" とした。一边のライトの数が 21 であるライツアウトは、解が一意に定まる。

QR コードの黒ライトを 1、白ライトを 0 としたときのプログラムの入力と実行結果を次のページに示す。（左：QR コード入力 右：解の出力）

