

卒業研究報告書

題目

変形倉庫番ゲームの計算量

指導教員

石水 隆 講師

報告者

14-1-037-0224

西村 遼

近畿大学工学部情報学科

平成 31 年 1 月 24 日提出

概要

本研究では、倉庫番ゲームに新たな制約を加えることで既存ゲームの可能性を広げることを目指す。「倉庫番 (Sokoban)」とは、1982年12月に有限会社シンキングラビットから発売されたコンピュータパズルゲームである。押し物系パズルである倉庫番は、 1×1 ブロック「荷物」および「壁」の配置と、目標位置の集合および操作キャラ「番人 (pusher)」の位置が与えられたときに、全ての荷物を目標位置に移動させることを目的とするパズルゲームである。番人が垂直方向か水平方向に1単位動くのがゲームの1手である。番人の移動先に荷物があつたとき、先が空いていれば、荷物は隣のスペースに押されて移動する。一方、先が空いていなければ番人はそちらに動けない。すべての目標位置に荷物を置く一連の手順を見つけることが目標である。

倉庫番は、手数制限のない1人ゲームである。手数制限のない1人ゲームとは打つ手の回数に制限のないパズルであり、移動が可逆なものが典型的である。例えば15人パズルや箱入り娘に代表されるスライディングブロックパズルでは、箱の中でピースをスライドする回数に制限がなく、スライドしたばかりのブロックは、いつでもすぐに元の位置に戻ることができる。

パズルを解く際の移動回数に対する多項式上界がないため、もしかしたら解は指数関数的な回数の移動を要するかもしれない。したがって提示された解の正当性を多項式時間でチェックすることは、もはや不可能である。実際手数制限のないパズルは、しばしば PSPACE 完全になる。こうしたパズルで解の計算に必要なのは、現在の配置と、その時点での移動だけである。そしてパズルの解となる移動列は非決定的に予測すればよいので、明らかに非決定的多項式領域 (NSPACE) で解くことができる。ここでサビッチの定理 [1] より $PSPACE = NPSPACE$ なので、これらのパズルは決定的に多項式領域で解くことができる。

本研究では代表的な PSPACE 完全問題である非決定制約論理問題 (NCL) を倉庫番に帰着させることを目指す。すなわち、任意の NCL 問題に対し、それが解を持つときのみ解を持つような荷物の並び方が作成できるかを検証する。

目次

1	序論	1
1.1	本研究の背景	1
1.2	計算量	1
1.3	計算量の大きいパズルゲーム	1
1.4	既知の結果	1
1.5	本研究の目的	2
1.6	本報告書の構成	2
2	倉庫番	2
2.1	倉庫番とは	2
2.2	変形倉庫番	2
3	制約論理と制約グラフ	3
3.1	制約グラフ	3
3.2	非決定性制約論理 (NCL)	3
3.3	PSPACE 完全性の証明の戦略	4
4	証明	4
4.1	二人倉庫番	4
4.2	摩擦の無い倉庫番	6
5	結論・今後の課題	8
	謝辞	9

1 序論

1.1 本研究の背景

押し物系パズルの一つである倉庫番 (Sokoban) は, 1×1 ブロック「荷物」および「壁」の配置と, 目標位置の集合および操作キャラ「番人」の位置が与えられたときに, 全ての荷物を目標位置に移動させることを目的とするパズルゲームである. 番人の移動先に荷物があったとき, 先が空いていれば, 荷物は隣のスペースに押されて移動する. 一方, 先が空いていなければ番人はそちらに動けない. すべての目標位置に荷物を置く一連の手順を見つけることがこのゲームの目標である.

1.2 計算量

NP とは, 非決定性多項式時間 (Non-deterministic Polynomial time) である. 定義は, 「非決定性チューリングマシンによって多項式時間で解くことができる問題」かつ「yes となる証拠が与えられたとき, その証拠が本当に正しいかどうかを多項式時間で判定できる問題」である.

非決定性チューリングマシンとは, データ x に対して決まった (同じ) y が一意付けられておらず, 状態が確定しないことを指す. 現在の状態から次の状態への遷移選択において, 解に近づく選択肢を選べる (解を求めるのに効率的な選択が毎回可能な) マシンである. 具体的には, 量子コンピュータ, 並列処理できる強力なコンピュータ, 普通のコンピュータである.

量子コンピュータは, 指数時間で解く問題を多項式時間で解けるようになったりすると言われている. 普通のコンピュータも非決定性チューリングマシンに属しているが, 突き詰めると決定性チューリングマシンでもある.

PSPACE とは, 計算複雑性理論における複雑性クラスの一つ (Polynomial Space) である. 定義は, 「非決定性チューリングマシンに多項式領域で解くことができる問題」である.

PSPACE 完全とは, NP 完全 [17] と同様に PSPACE に属する全ての問題から多項式時間還元可能であり, 自らも PSPACE に属する問題のことである.

1.3 計算量の大きいパズルゲーム

パズルゲームには, 計算量の観点から一般的な解放が存在しないと思われるものもある. 例えば数独カックロ, スリザーリンク, テトリス, ぷよぷよは NP 完全であることが示されている [9, 10, 11, 12, 13] また, ラッシュアワー, 橋渡しゲーム等は P-SPACE 完全であることが示されている [14, 2].

これらのパズルゲームの多くは, 手数制限のあるものは NP 完全に, 手数制限の無いものは P-SPACE 完全になる [2] このような計算量の大きいパズルゲームには一般的な解放を求めることは不可能であると考えられており, サイズの小さい部分問題に対する解き方やヒューリスティックな手法が用いられる.

1.4 既知の結果

本節では, 倉庫番に関する既知の結果について述べる.

Culberson は, 領域限定のチューリング機械に対応する倉庫番の配置を構成する方法を示して, 倉庫番が PSPACE 完全であることを示した [4, 18].

倉庫番ゲームに対し、村瀬らは倉庫番の問題の自動生成プログラムを生成した [3]。村瀬らのプログラムは、小さい領域と大きい領域に分け、(1) 効率よく問題を作成する、(2) 作成されたものを解き、解のデータを作る、(3) 得られたデータを基に、問題のおもしろさを評価するという3つの部分から成る、倉庫番のゲーム性のおもしろさを追求した研究である。

一方、倉庫番の解法を求める手法は、小田原らが部分マップの手詰まり判定法を提案している [6] が、前述のように倉庫番自体は本質的に PSPACE 完全であり、荷物の個数が大きくなると解くことができない。また、複数の荷物を同時に押せたり、押した荷物が滑ったりといった条件を加えたいいくつかの変形倉庫番に対する計算量も求められている [6]。

1.5 本研究の目的

本研究では代表的な PSPACE 完全問題である非決定性制約論理問題 (NCL) を倉庫番に帰着させることを目指す。すなわち、任意の NCL 問題に対し、それが解を持つときのみ解を持つような荷物の並び方が作成できるかを検証する。

1.6 本報告書の構成

本報告書の構成は以下の通りに述べる。第2章は倉庫番の説明を踏まえて、AND ガジェット・OR ガジェットの作成の仕方と非決定性制約論理 (NCL) の研究内容を述べる。第4章では NCL を作成した二人倉庫番ガジェットと摩擦の無い倉庫番に帰着させ、解を持つような荷物の並び方が作成できているかの考察を述べ、今後の課題を述べる。

2 倉庫番

2.1 倉庫番とは

本節では、本研究の対象である倉庫番について述べる。

本研究では、倉庫番ゲームに新たな制約を加えることで既存ゲームの可能性を広げることを目指す。「倉庫番 (Sokoban)」とは、1982年12月に有限会社シンキングラビットから発売されたコンピュータパズルゲームである。押し物系パズルである倉庫番は、 1×1 ブロック「荷物」および「壁」の配置と、目標位置の集合および操作キャラ「番人 (pusher)」の位置が与えられたときに、全ての荷物を目標位置に移動させることを目的とするパズルゲームである。番人が垂直方向か水平方向に1単位動くのがゲームの1手である。番人の移動先に荷物があつたとき、先が空いていれば、荷物は隣のスペースに押されて移動する。一方、先が空いていなければ番人はそちらに動けない。すべての目標位置に荷物を置く一連の手順を見つけることが目標である。

2.2 変形倉庫番

本節では、変形倉庫番について述べる。

本研究では、番人が2人、2人が協力すれば同時に2つ荷物を押せるという条件を加えた2人倉庫番、および床との摩擦が無く一度荷物を押すと障害物が無い限り延々滑るという条件を加えた滑る倉庫番に対してその計算量を検証する。

3 制約論理と制約グラフ

3.1 制約グラフ

R.A.Hearn らは、様々なゲームの計算量を証明するために制約グラフ (constraint graph) を用いている [2] . [2] の 2.1 節では制約グラフを以下のように定義している .

制約グラフ (constraint graph) とは、辺に 1 か 2 の重みが付けられた有向グラフである . それぞれの辺は、重みに応じて赤か青の色が付いている . 各頂点を指している辺の重みの和を、その頂点の流入 (inflow) と呼び、各頂点は非負の最小流入 (mimimum inflow) を持つ . 制約グラフの各頂点において、最小流入以上の流入があるとき、その状態を正当 (legal) であるといい、これがこのグラフにおける制約である . 制約グラフ上の正当な移動 (legal move) とは、正当な状態を保持したままで、一つの辺の向きを反転することである . 一般に、正当な移動を順に実行して、指定された辺の向きを反転することが、そのゲームにおけるゴールである . プレイヤーが複数いるゲームでは、それぞれの辺は別のプレイヤーによって制御され、各プレイヤーはそれぞれ固有のゴール辺を持つ . 決定性のゲームでは、一意的な反転の列が強制的に与えられる . また、手数が制限されたゲームでは、それぞれの辺は一度しか反転できない .

R.A.Hearn らは、制約グラフを用いてゲームの特性を表すために、AND、OR、FANOUT、CHOICE の基本頂点を定義している . 図 1 に制約グラフの基本頂点を示す .

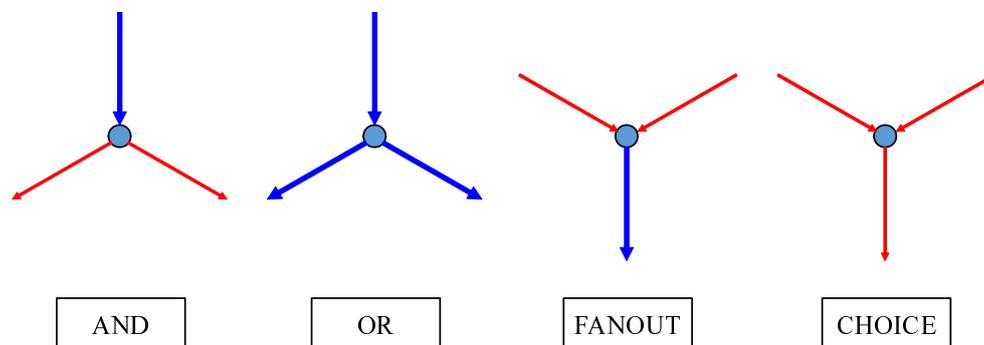


図 1 制約グラフの基本頂点

3.2 非決定性制約論理 (NCL)

本節では、非決定性制約論理 (Nondeterministic Constraint Logic, NCL) について述べる .

NCL には手数制限あり非決定性制約論理 (Bounded Nondeterministic Constraint Logic (Bounded NCL)) と手数制限なし非決定性制約論理 (Nondeterministic Constraint Logic, NCL) が存在する .

Bounded NCL および NCL は以下で定義される .

[Bounded NCL]

入力： 制約グラフ G , G の辺 e .

質問： それぞれの辺が高々 1 回しか反転できないとき、辺 e を反転する G 上の移動が存在するか？

[NCL]

入力： 制約グラフ G , G の辺 e .

質問： G の移動の列で、辺 e を反転するものが存在するか？

Bounded NCL は手数制限のあるゲームに、NCL は手数制限の無いゲームに対応する .

R.A.Hearn らは、Bounded NCL が NP 完全、NCL が PSPACE であること、また、制約グラフが図 1 の基本頂点から成る平面グラフに限定しても Bounded NCL が NP 完全、NCL が PSPACE であることを示した [2] .

3.3 PSPACE 完全性の証明の戦略

本節では、変形倉庫番の計算量を示すために本研究で用いる戦略について述べる .

変形倉庫番は手数制限の無いゲームである . よって、NCL 問題から変形倉庫番に帰着させることを目指す . そのためには、制約グラフの基本頂点に対して、対応する変形倉庫番の配置を考えればよい .

倉庫番の証明では、取り返しのつかない配置 (unrecoverable configuration) というアイデアが極めて重要である . ここで構成する倉庫番の問題では、パズルが解けるなら、任意の解状態から、すべての「押し」を逆回しにすればもとの配置に戻せる . ところが逆回しできない「押し」をしてしまうと、取り返しのつかない配置になる . 例えば図 2 の部分的な配置でブロック A を左に押すと、ブロック D と並んでくっついてしまって、取り返しがつかない . これを再び動かせるような番人の立ち位置がなくなってしまう . これから構成するパズルは、こうした移動を含む解は存在しない . こうした移動でできる配置は厳密には正当な配置であるが、今後は禁止されている、もしくは不可能であるという .

本研究では NCL 問題を倉庫番に帰着させるために、OR 演算に相当する荷物の並び「OR ガジェット」および AND 演算に相当する荷物の並び「AND ガジェット」を作成する . 各ガジェットには複数の「入口」と「出口」があり、OR ガジェットは入口のどれか一つから番人が来るとガジェット内の荷物を手詰まり状態にすることなく出口に抜けることができ、AND ガジェットは全ての入口から番人が入ったときのみガジェット内の荷物を手詰まり状態にすることなくガジェットを通り抜けられるようにする .

4 証明

本章では、変形倉庫番の計算量を証明する .

4.1 二人倉庫番

二人倉庫番の計算量について以下の定理が成り立つ .

[定理] 二人倉庫番は PSPACE 完全である .

以下にその証明を述べる .

図 3 に二人倉庫番の AND ガジェット、図 4 に二人倉庫番の OR ガジェットを示す .

図 3 の左半分は、F および G が動かさせたときのみ手詰まり状態にならずに C を動かせるようにする AND 論理を実現する配置であり、右半分は番人が二人いるときのみ荷物を動かせるようにする配置である . まず左

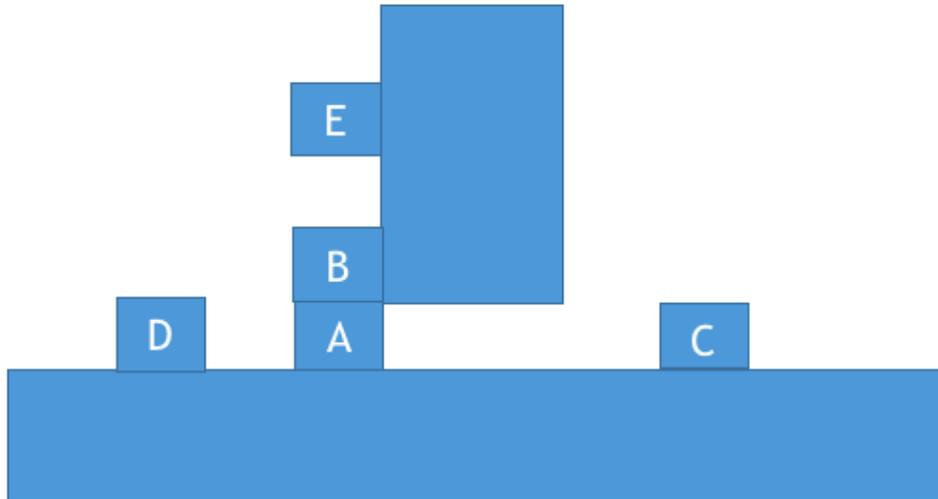


図2 倉庫番の AND ガジェット

半分は、手詰まり状態にならずに C を左に動かすためには、あらかじめ F, D を左, G, E を上に動かすことで、C を手詰まり状態になることなく左に動かすことができる。一方、右半分では C のある領域に入るためには B を左に動かす必要があるが、そのためには A を下に動かしてしまうと A は身動きがとれなくなってしまうので、A を上に動かす必要がある。しかし、番人が 1 人しかいない場合、A を上に動かしてしまうとその後 A を動かすことができず、手詰まり状態になってしまう。一方、番人が 2 人いる場合、番人の 1 人が A の上の隙間に待機しておくことで、A を上に動かした後再び A を元の位置に戻すことができる。以上のことから、AND ガジェットは番人が 2 人おり、かつ F との両方を動かせる場合のみ C を動かすことができる。

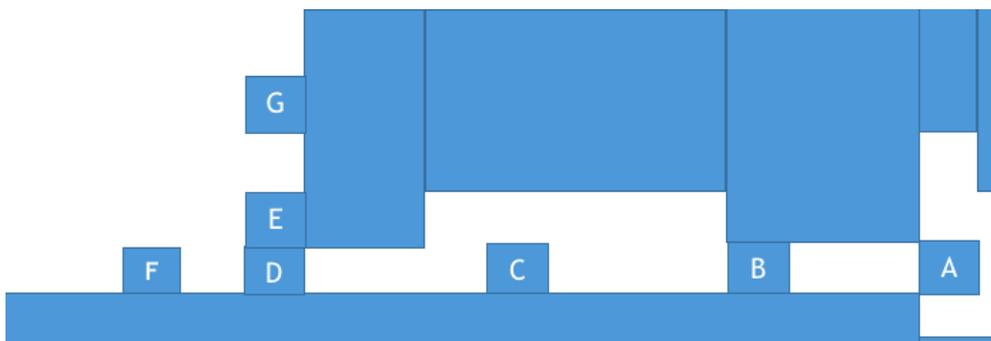


図3 二人倉庫番の AND ガジェット

図 4 の左半分は、F または G が動かさせた時のみ手詰まり状態にならずに C を動かせるようにする OR 論理を実現する配置であり、右半分は番人が二人いる時のみ荷物を動かせるようにする配置である。まず左半分は、手詰まり状態にならずに C を左に動かすためには、まず、D が左に動くか、または E が上に動くこと

であることを示す．そのためにはまず F が左に動いて G が上に動かなければ成らない．そうでないと D や E を動かしたときに，取り返しのつかない配置になってしまう．だから，D か E を動かす前に，F か G をどかさなければならない．そして，もし D が左に動いたら，C を左に動かしても良い．D を動かしたことで開いた隙間を使えば，あとで C を元に戻すために番人が入り込むことができる．B についても同様である．しかし，D や E を左や上に移動しないで C を左に押ししてしまうと，取り返しのつかない配置に追い込まれてしまう．

図 4 の右半分は，AND ガジェットと同様に番人が二人いるときのみ荷物を動かせるようにするための配置である．

AND ガジェットおよび OR ガジェットを用いて，任意の NCL 問題を二人倉庫番に還元することができる．よって，二人倉庫番は PSPACE 完全であることが示される．

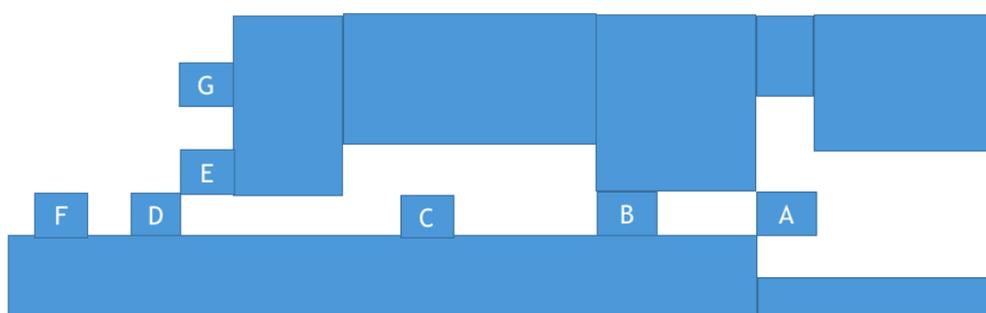


図 4 二人倉庫番の OR ガジェット

4.2 摩擦の無い倉庫番

摩擦の無い倉庫番の計算量について以下の定理が成り立つ．

[定理] 摩擦の無い倉庫番は PSPACE 完全である．

以下にその証明を述べる．

図 5 に摩擦の無い倉庫番の AND ガジェット，図 6 に摩擦の無い倉庫番の OR ガジェットを示す．

図 5 は，摩擦のない倉庫番の AND パターンのガジェットである．A を（目的地）へと動かせるようにする AND 論理を実現する配置である．A を目的地へと動かすためには，B および D を動かさなければならない．まず，A を右に動かすと摩擦が無いので C にぶつかってしまう．その場合，B および C の位置によって取り返しのつかない配置になってしまうので，A を右に動かす前に D を下に動かしてそれを避けなければならない．もしも，D を元通りにしたいならば，下，右，上，左，下と順に動かせば元に戻すことができる．そして，A を に向かわせるために A を上に動かす．先程と同じように，A を上に動かすと摩擦が無いので E にぶつかってしまう．その場合，D および E の位置によって取り返しのつかない配置になってしまうので，A を上に動かす前に G を右に動かしてそれを避けなければならない．もしも，G を元に戻したいならば，右，上，左，下と順に動かせば元に戻すことができる．A を元に戻したいならば，左，下と順に動かせば元に戻すことができる．

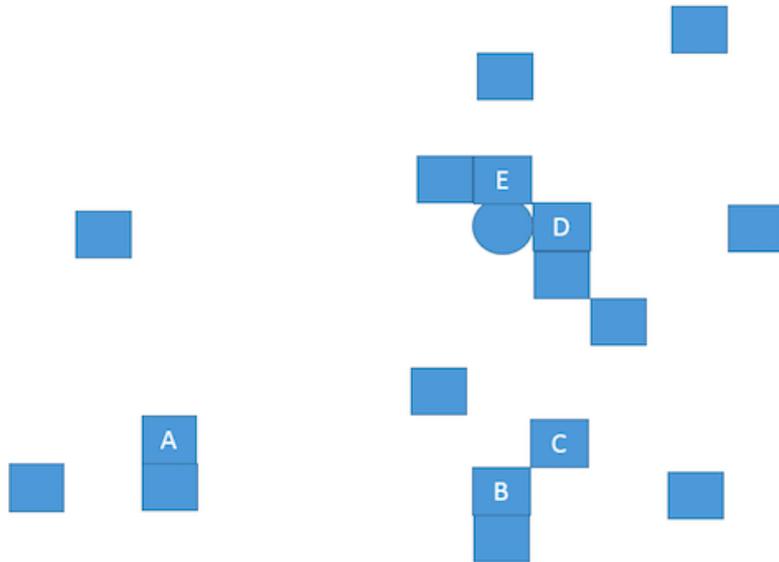


図 5 摩擦の無い AND ガジェット

図 6 は、摩擦のない倉庫番の OR パターンのガジェットである。A を元の位置へと戻せるようにする OR 論理を実現する配置である。手詰まり状態にならずに A を元の位置に動かすためには、まず、F が上に動くか、または H が上に動くことであることを示す。そうしなければ、A を E および I に動かした場合取り返しのつかない配置になってしまう。A を E にぶつけた場合は F を上に動かすことで取り返しのつかない配置を回避することができる。F を上に動かして元に戻したい場合は、右、下、左に動かすことで元通りになる。その後の A は下、左、下、右のように D、J、C、B にぶつかっていけば A は元の位置に動かすことができる。A を I にぶつけた場合は H を上に動かすことで取り返しのつかない配置を回避することができる。H を上に動かして元に戻したい場合は、左、下、右に動かすことで元通りになる。その後の A は下、右のように C、B にぶつかっていけば A は元の位置に動かすことができる。

AND ガジェットおよび OR ガジェットを用いて、任意の NCL 問題を摩擦の無い倉庫番に還元することができる。よって、摩擦の無い倉庫番は PSPACE 完全であることが示される。

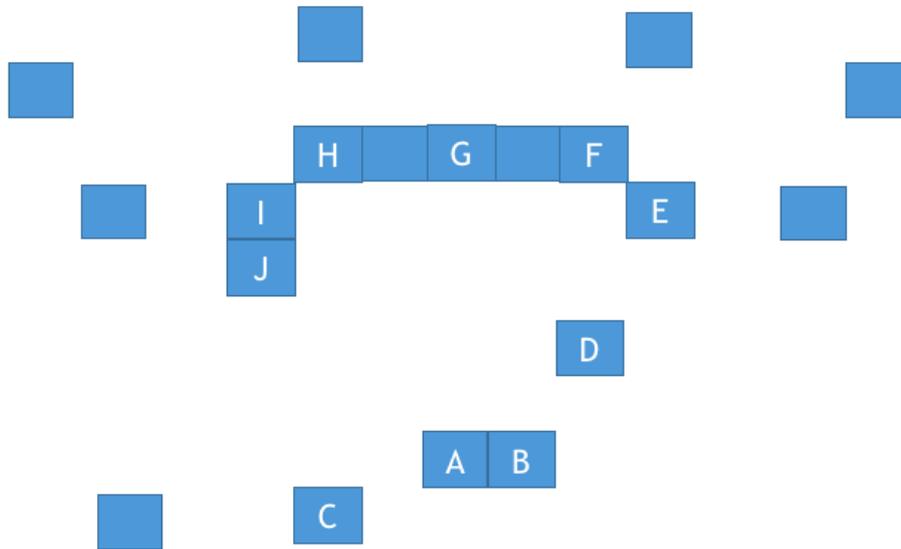


図 6 摩擦の無い OR ガジェット

5 結論・今後の課題

本研究で、二人倉庫番と摩擦の無い倉庫番の AND ガジェットと OR ガジェットを作成し PSPACE 完全であることを証明した。二人倉庫番では一人では不可能であった荷物と壁の間に二人の内一人が入りこみ取り返しの配置を回避するといったことが可能になった。摩擦の無い倉庫番では、動かした荷物を元通りに戻せる配置を作成することに成功した。こうすることで、初めからやり直す必要がなくスムーズに動かすことができる。

今後の課題は、変形倉庫番に対するヒューリスティックな手法を求めること、また、類似のゲームの計算量の検証を行うである。変形倉庫番に対するヒューリスティックな手法としては、代表的な荷物や壁の配置をパターン化し、パターンごとに解法を作成すること、手詰まりになる荷物の移動を避ける方法を求めること等が考えられる。倉庫番に類似したゲームとしては、荷物を引っ張れたり、荷物を壁に押し付けると荷物が横にずれたりする。荷物に重さを設定して、制限重量以下の荷物を一度に押せる、等の条件を加えたものが考えられる。

謝辞

卒業ゼミから卒業研究の一年半に渡って私が理解できなかった点を理解するまで付き添っていただき感謝します。初めは本研究のテーマが理論であることで参考書を読んでもあまり理解が出来ませんでした。ゼミの時間外に研究室に出向くと親身になって相談にのっていただきここまで頑張ってくれたと思います。倉庫番を作成するにあたって、どのような方法でNCLに還元するのか悩んでいたところで例を出して方向性を指し示し理論立てて教えて下さいました。私に対して研究を頑張っているのかと優しく接してくれた教員の方々にも感謝しています。そのおかげで沈みかけた気持ちが和らいだと感じています。これまでの研究を通して卒業してから社会人として真っ当に一生懸命頑張っていきたいと思います。今までありがとうございました。指導を受けた教員や、本研究を完成するにあたって支援を受けた研究室の諸氏に対しお礼の言葉を、独立したページに記述する。詳しくは卒業研究担当教員の指導に従うこと。

参考文献

- [1] Walter J.Savitch, 'Relationships between Nondeterministic and Deterministic Tape Complexities, Journal of Computer and System Sciences 4:2, pp. 177-192, (1970)
https://ac.els-cdn.com/S002200007080006X/1-s2.0-S002200007080006X-main.pdf?_tid=5e7bd8ca-c8ce-4a31-bb87-2d2acf8022c6&acdnat=1548385964_0ce63a28b5841ff90adcecd2642213f6
- [2] ロバート・A・ハーン, エリック・D・ドメイン : ゲームとパズルの計算量, 近代科学社 (2011).
- [3] 村瀬芳生, 松原仁, 平賀謙 : 「倉庫番」の問題の自動作成, 情報処理学会論文誌, No.3, Vol.39, pp. 567-574 (1998). <http://id.nii.ac.jp/1001/00013161/>
- [4] J.C.Culberson, Sokoban is PSPACE-complete, Technical Reports 97-02, Department of Computing Science, University of Alberta, (1997)
<http://cl-informatik.uibk.ac.at/teaching/ss07/alth/material/culberson97sokoban.pdf>
- [5] J.C.Culberson : Sokoban is PSPACE-complete, Proc. International Conference of Fun with Algorithms, pp.65-76 (1998)
- [6] 小田原大, 金子知適, 川合慧 : 倉庫番における部分マップの組み合わせに基づく手詰まり判定法, 情報処理学会 研究報告, Vol.2004GI-12, pp.33-40 (2004) <http://id.nii.ac.jp/1001/00058546/>
- [7] E.D.Demaine, I.Grosz, J.Lynch, Push-Pull Block Puzzles are Hard, Proc. Algorithms and Complexity: 10th International Conference, CIAC 2017 (2017)
http://erikdemaine.org/papers/PushPull_CIAC2017/paper.pdf
- [8] J.Taylor, I.Parberry, PROCEDURAL GENERATION OF SOKOBAN LEVELS, Proc. 6th Annual North American Conference on AI and Simulation in Games pp. 5-12, (2011)
https://larc.unt.edu/ian/pubs/GAMEON-NA_METH_03.pdf
- [9] C. J. Colbourn, The complexity of completing partial Latin squares.Discrete Appl. Math., Vol. 8, pp.2530 (1984).
- [10] T. Yato. Complexity and completeness of finding another solution and its application to puzzles. Master thesis, the University of Tokyo, 2003.
<http://www-imai.is.s.u-tokyo.ac.jp/~yato/data2/MasterThesis.pdf>
- [11] 瀬田 剛広, カックロの計算量、情報処理学会アルゴリズム研究会 84-8 (2002.5), http://www-imai.is.s.u-tokyo.ac.jp/~seta/paper/SIGAL84-8/cross_sum.pdf
- [12] 牟田 秀俊, ぷよぷよは NP 完全, 電子信学会技術研究報告, COMP, コンピューテーション Vol.105, No.72, pp.39-44, 電子情報通信学会 (2005)
- [13] Erik D. Demaine, Susan Hohenberger, David Liben-Nowell, Tetris is Hard, Even to Approximate, Computer Science Vol.2002, No.20 pp,1-56, Cornell University Library, (2002),
<https://arxiv.org/abs/cs/0210020>
- [14] G.W.Flake, E.B.Baum, Rush Hour is PSPACE-complete, or 'Why You Should Generously Tip Parking Lot Attendants', Theoretical Computer Science Vol. 270, No. 1-2, pp.895-911, (2002)
- [15] David Lichtenstein. 'Planar Formulae and Their Uses.' SIAM J. Comput. 11:2 (1982), 329-343.
- [16] 岩塚卓弥, 寺内多智弘, 結緑祥治 : 無限小定数と限量子除去法によるハイブリッドシステムの検証に向け

て, 情報処理学会論文誌, No.3, Vol.6, pp. 20-32 (2013),

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=8&ved=2ahUKEwiB8oidy5rgAhUW_GEKHSNyDP4QFjAHegQIBBAC&url=https

- [17] Richard Kaye, Minesweeper is NP-Complete, *Mathematical Intelligencer*, 22, 9-15, (2000).
- [18] J.C.Culberson : Sokoban is PSPACE-complete, *Proc. International Conference of Fun with Algorithms*, pp.65-76 (1998)