QBF ソルバを用いた一般化三並べの拡張の勝敗判定

ディプタラマ[1]　石黒 裕也[1]　成澤 和志[1]　篠原 歩[1]　ジョーダン チャールズ[2]

概要：一般化三並べは Frank Harary によって提案された 2 人完全情報ゲームであり，碁盤面上に先手後手が交互に石を 1 つずつ置き，あらかじめ定められた動物を先に作ったプレイヤが勝ちというゲームである．2 人完全情報ゲームの勝敗判定問題は与えられたゲームに対して，先手必勝，後手必勝または引き分けとなるかを判定する問題である．オセロや五目並べなど，多くの 2 人完全情報ゲームの勝敗判定問題は PSPACE 完全であることが知られている．それに対して，代表的な PSPACE 完全問題として Quantified Boolean Formula (QBF) に対する充足可能性問題 (TQBF) が存在し，TQBF を高速に解くプログラム，QBF ソルバが開発されてきた．本研究では一般化三並べの拡張である $\mathsf{GTTT}(p,q)$ および $\mathsf{TorusGTTT}(m,n)$ の勝敗判定問題を TQBF に帰着し，QBF ソルバを用いてゲーム勝敗判定問題を解く．ゲームのパラメータによっては既存のゲームの探索手法より，QBF ソルバの方がより速く勝敗判定問題を解けることが見られた．

Solving Generalized Tic-Tac-Toe by Using QBF Solver

Diptarama[1]　Yuya Ishiguro[1]　Kazuyuki Narisawa[1]　Ayumi Shinohara[1]　Charles Jordan[2]

Abstract: Generalized tic-tac-toe is an achievement game for polyominoes introduced by Frank Harary. Two players alternately put one stone over a board, and the player who first achieves a given polyomino wins the game. The game solving problem for 2 players game with perfect information is to determine whether the first player wins the game, the second player wins the game, or the game draws if both players play optimally. Many game solving problems for two players with perfect information games, such as Gomoku and Othello are PSPACE complete. On the other hand, boolean satisfiability problem for quantified boolean formula (QBF) is a representative of PSPACE complete problems, and efficient QBF solvers have been developed. In this paper we solve the game solving problems for two extensions of generalized tic-tac-toe, $\mathsf{GTTT}(p,q)$ and $\mathsf{TorusGTTT}(m,n)$ by reducing them to TQBF, and then use QBF solvers. From the results, in some cases, QBF solvers can solve the games faster than game tree search algorithm.

1. はじめに

ゲームの**勝敗判定問題**は，与えられたゲームに対して，すべてのプレイヤが最善手を打ち続けた場合，どのプレイヤが勝つか，または引き分けになるかを判定する問題である．特に，2 人完全情報ゲームに対する勝敗判定問題は多く研究されており，様々なゲームの勝敗が解明されてきた [14]．

2 人完全情報ゲームで古くから研究されているゲームの一つとして**一般化三並べ**がある．一般化三並べは，Frank Harary によって提案された 2 人ゲームであり，碁盤面上に先手後手が交互に石を 1 つずつ置き，あらかじめ決められた動物と呼ばれる辺で連結した石の配置を先に作ったプレイヤが勝ちという完全情報ゲームである [6], [7], [16]．一般化三並べでは，盤面の拡張 [3], [13] や手数の拡張 [13], [15] など，様々なルールを拡張したゲームにおける勝敗判定に関する研究が行われてきた．

一般化三並べをはじめとする五目並べやオセロなど，多くの 2 人完全情報ゲームの勝敗判定問題は PSPACE 完全問題であり [8], [12]，勝敗判定問題の解（先手必勝，後手必勝，または引き分け）を求めることが，困難である．そのため，各ゲームにおける勝敗判定問題を解くために，α-β

[1] 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University
[2] 北海道大学大学院情報科学研究科
Graduate School of Information Science and Technology, Hokkaido University

法や探索証明数探索 [1] などが開発されてきた．

一方，PSPACE 完全問題の代表的な問題の一つに True Quantified Boolean Formula (TQBF) がある．TQBF は最も基本的な PSPACE 完全問題であり，TQBF を高速に解く様々なアルゴリズムが開発されている [4], [5], [9], [11]．これらのアルゴリズムを用いて，TQBF を解くプログラムを **QBF ソルバ**という．

本研究では，一般化三並べの拡張の勝敗判定問題を TQBF に帰着し，QBF ソルバを用いて勝敗判定を行う．ゲームとして，初手は q 個，その後 p 個ずつ石を置く一般化三並べ拡張，GTTT(p,q) [15] および，ドーナツ型盤面を用いた一般化三並べ拡張，TorusGTTT(m,n) [17] を用いる．また，この手法と既存の各探索手法を用いた場合での勝敗判定にかかる計算時間の速度比較を行う．実験を通して，ゲームのパラメータによって QBF ソルバが探索手法に比べて高速であることを示す．

2. 一般化三並べの拡張

一般化三並べは，Frank Harary によって提案された 2 人完全情報ゲームであり，以下のように定義される．

定義 1 (一般化三並べ) 無限大の大きさの碁盤目状の盤面に，先手後手が交互に石を 1 つずつ置き，あらかじめ決められた目標動物を先に作ったプレイヤが勝ちとなるゲームを一般化三並べという．

ここで，動物とは辺で連結した石の配置の形であり，90 度ごとの回転や反転した配置は同じ動物であると見なす．また，動物を構成する石の数を細胞数と呼ぶ．

一般化三並べに対して様々な拡張が行われてきたが，本研究では手数に対する拡張である GTTT(p,q)，および盤面に対する拡張である TorusGTTT(m,n) の 2 つの拡張一般化三並べに対して解析を行う．

手数に対する一般化三並べの拡張である GTTT(p,q) は以下のように定義される．

定義 2 (GTTT(p,q) [15]) $p \geq 1$，$q \geq 1$ とする．無限大の大きさの碁盤目状の盤面に，先手が最初に q 個の石を置き，その後は各プレイヤが交互に石を p 個ずつ置き，あらかじめ決められた目標動物を先に作ったプレイヤが勝ちとなるゲームを GTTT(p,q) という．

一般化三並べでは後手が勝てないことが知られているが，GTTT(p,q) では p および q の値によって，後手必勝となることもある．

また，盤面に対する一般化三並べの拡張である TorusGTTT(m,n) は以下のように定義される．

定義 3 (TorusGTTT(m,n) [17]) 中央領域が $m \times n$ $(m \geq n)$ の大きさのトーラス盤面に先手後手が交互に石を置き，あらかじめ決められた目標動物を先に作ったプレイヤが勝ちとなるゲームを TorusGTTT(m,n) という．

ここで，トーラス盤面とは，盤面の右端と左端，上端と下端もつながっている盤面のことである．TorusGTTT(m,n) は一般化三並べと同様に後手が勝てないという性質をもつ．

3. QBF ソルバ

Quantified Boolean Formula (QBF) とは，量化記号 $(\forall, \exists)$ がつけられた論理式のことであり，以下のように定義されている．

$$Q_1 x_1 Q_2 x_2 ... Q_n x_n . \phi(x_1, x_2, ..., x_n)$$

$$Q_1, Q_2, ..., Q_n \in \{\forall, \exists\}$$

ここで，$\phi(x_1, x_2, ..., x_n)$ は $x_1, x_2, ..., x_n$ を変数としたブール式である．また，与えられた QBF が充足可能かどうかを判定する問題を True Quantified Boolean Formula (TQBF) という．TQBF は PSPACE 完全の代表的な問題であり，高速に解くことが非常に困難な問題である．

TQBF に対して様々なアルゴリズムが開発されており，実用的に高速に解くプログラムを QBF ソルバという．本論文では拡張した一般化三並べを解くために，DepQBF [11] と RAReQS [9] の 2 つの QBF ソルバを用いる．

QBF ソルバは与えられた QBF に対して充足可能かどうかを出力する．ゲーム勝敗判定問題から TQBF に帰着するときに，先手の視点からの QBF と後手の視点から QBF は異なる．先手の視点からの QBF は充足可能であれば，そのゲームは先手必勝であり，充足不可能であれば，そのゲームは後手必勝または引き分けである．後手も同様に充足可能な場合は後手必勝であり，そうでない場合は後手必勝ではない．そのため，先手必勝または後手必勝を証明するために，どちらかが充足可能であれば十分であるが，引き分けを証明するために両方が充足不可能であることを確認しなければならない．

一般化三並べおよび TorusGTTT(m,n) においては，後手必勝でないという性質を使うことで，先手の視点からの QBF のみを解くだけで十分である．なぜなら，QBF が充足可能の場合は先手必勝であり，充足不可能の場合は引き分けであることがわかるからである．

4. Toss

拡張一般化三並べを QBF ソルバで解くために，TQBF に帰着することが必要である．そこで，帰着ツールとして Toss を用いる [10]．

Toss とは，離散構造とその構造の更新法則を入力して，グラフの描画や，グラフの還元，性質を観測できるツール

```
PLAYERS 1, 2
REL DiagA (x, y) = ex u (R(x, u) and C(u, y))
REL DiagB (x, y) = ex u (R(x, u) and C(y, u))
REL Row3 (x, y, z) = R(x, y) and R(y, z)
REL Col3 (x, y, z) = C(x, y) and C(y, z)
REL DiagA3 (x, y, z) = DiagA(x, y) and DiagA(y, z)
REL DiagB3 (x, y, z) = DiagB(x, y) and DiagB(y, z)
REL Conn3 (x, y, z) = Row3(x, y, z) or Col3(x, y, z) or DiagA3(x, y, z) or DiagB3(x, y, z)
REL WinQ() = ex x, y, z (Q(x) and Q(y) and Q(z) and Conn3(x, y, z))
REL WinP() = ex x, y, z (P(x) and P(y) and P(z) and Conn3(x, y, z))
RULE Black:
  [a1| | - ]
    ->
  [a1|P{a1}| - ]
    emb Q,P pre not WinQ()
RULE White:
  [a1| | - ]
    ->
  [a1|Q{a1}| - ]
    emb Q,P pre not WinP()
LOC 0 {
  PLAYER 1 { PAYOFF :(WinP()) - :(WinQ())
             MOVES [Black -> 1] }
  PLAYER 2 { PAYOFF :(WinQ()) - :(WinP()) }
}
LOC 1 {
  PLAYER 1 { PAYOFF :(WinP()) - :(WinQ()) }
  PLAYER 2 { PAYOFF :(WinQ()) - :(WinP())
             MOVES [White -> 0] }
}
START [ | P:1 {}; Q:1 {} | ] "

        *  *  *

        *  *  *

        *  *  *
"
```

図 1 三並べを Toss に入力する場合の toss ファイルの例

である．このツールの 1 つの機能として，五目並べやオセロなどのボードゲームを離散構造および更新法則として入力することで，QBF へ自動的に変換することができる．

4.1 Toss の入力

ゲームを QBF に変換するために，Toss の入力の形式に従って定義しなければならない．ここで，Toss の入力形式で書かれたファイルを **toss ファイル**と呼ぶ．Toss の入力は関係，構造，更新法則，手番と初期状態で構成されている．図 1 に通常の三並べを toss ファイルにした場合の例を示す．以降，この例をもとに説明する．

関係

関係とは，ゲームの勝利条件や，可能な手などのゲームの性質をブール式で定義するものであり，以下のように表記する．

$$\text{REL } R(x_1, x_2, ...) = f$$

ただし，ブール式 f は以下の文法を満たす．

$$f := t > 0 \mid t = 0 \mid R(x_1, x_2, ...) \mid \text{not } f \mid f \text{ and } f \mid f \text{ or } f \mid \text{ex } x\ f \mid \text{all } x\ f$$

t は整数の式であり，以下の文法を満たす．

$$t := :(f) \mid t + t \mid t * t \mid t \hat{}\ t$$

ここで，R を関係，f をブール式，t を整数の式，x を変数とする．$:(f)$ はブール値から整数値への評価を表す．図 1 の 2 行目から 10 行目は三並べにおける関係の例を示す．

図 1 の 2〜8 行目は石の連結に関する関係の定義である．$R(x,y)$ は y が x の右隣りにあるとき，$C(x,y)$ は y が x の真下にあるときに真となる関係として定義している．また，y が x の右斜め下にあるときに真となる関係を $DiagA(x,y) = \exists u.R(x,u) \wedge C(u,y)$，$y$ が x の右斜め上にあるときに真となる関係を $DiagB(x,y) = \exists u.R(x,u) \wedge C(y,u)$ と定義している．

4〜8 行目は 3 連結に関する定義である．$Row3(x,y,z)$ は石が横に 3 つ連結した状態に関する定義であり，2 連結に関する $R(x,y)$ と $R(y,z)$ の両方が成り立つときに真となる．同様に石が縦に 3 つ連結した状態は $Col3(x,y) = C(x,y) \wedge C(y,z)$ と定義する．石が斜めに 3 つ連結した場合の定義は，図 1 の 6 および 7 行目に示す．これらを用いて，8 行目に縦，横，斜めのいずれかで 3 連結するときに真となる関係を定義している．

最後に，図 1 の 9〜10 行目は勝利条件に関する関係の定義である．三並べができており，かつその三並べがすべて同じ石で構成されている場合に真になる関係として定義している．

構造

構造は，ゲームの中の状態を表し，変数，関係と関数の 3 項組で構成され，以下のように表記する．

$$S := [\ x_1, x_2, ... \mid R_1, R_2, ... \mid F_1; F_2; ...\ ]$$

ここで，x_i を変数，R_i を関係，F_i を関数とする．

構造はこれらの 3 項組で構成されるが，三並べにおいては変数と関係の 2 つのみで十分であるため，関数は使用しない．変数はマスを表し，関係はそのマスに対する成り立つ関係を表す．例えば，関係 $P\{x\}$ を先手が石を位置 x に置くことすれば，

$$[\ x_1 \mid P\{x_1\} \mid\ ]$$

は，マス x_1 に先手の石が置かれることを表す構造である．

構造の例を図 1 の 12,14,17,19 行目に示す．12 行目および，17 行目は a_1 に石が置かれていない構造を表し，14 行目および，19 行目は a_1 にそれぞれ先手の石と後手の石が置かれている構造を表している．

更新法則

更新法則は，構造から構造への写像を表し，写像前後の構造と条件で構成されている．更新法則は以下のように表記する．

$$\text{RULE } U : S \to S' \text{ emb } R_1, R_2, ... \text{ pre } f_1 \text{ post } f_2$$

ここで，S は手を打つ前の構造，S' は手を打った後の構造である．f_1 は手を打つために必要な条件，f_2 は手を打った後に満たさなければならない条件である．つまり，更新法則はゲームにおいて，打つ手の種類を表している．

三並べにおいて手の種類として，

- 先手が石を 1 つ置く
- 後手が石を 1 つ置く

の 2 種類が存在する．三並べにおける先手の手の更新法則を図 1 の 11〜15 行目に示す．先手の手は，空いてるマスを表す構造を，先手の石が置かれたマスを表す構造に更新するものと定義する．このとき，先手が手を打つ条件としては，後手がまだ勝っていない場合とする．

同様に 16〜20 行目は後手の手を示し，空いているマスを表す構造から，後手の石が置かれるマスを表す構造への更新法則とする．石を置くための条件としては，先手が勝利条件を満たしていない場合とする．

手番

手番は，プレイヤと行われる手の種類，手を打たないときの結果（負けか引き分けかなど），次の手番への遷移で構成されており，以下のように表記する．

$$\begin{array}{l}\text{LOC } L \{ \\ \quad \text{PLAYER 1} \{ \quad \text{PAYOFF } t \\ \qquad\qquad\quad \text{MOVES } [U_{11} \to L_{11}];\ [U_{12} \to L_{12}];\ ...\} \\ \quad \text{PLAYER 2} \{ \quad \text{PAYOFF } t \\ \qquad\qquad\quad \text{MOVES } [U_{21} \to L_{21}];\ [U_{22} \to L_{22}];\ ...\} \\ \qquad\quad \vdots \\ \}\end{array}$$

ここで，L は手番の番号，U_i は更新法則である．PAYOFF はプレイヤが手を打てないときの結果を示し，t の値によって決まる．t が正のときは勝ち，0 のときは引き分け，負のときは負けと定義される．

手番の例を図 1 の 21〜30 行目に示す．三並べにおいては，先手と後手の 2 種類の手番が存在する．それぞれの手番で打つ手は更新法則で定義された手とする．また，三並べにおいては手番が交互にくるため，先手の手番が終わったら，後手の手番に遷移し，後手の手番が終わったら先手の手番に遷移するように定義する．

先手の手番において，後手が勝利条件を満たしたら，行える手がなくなったので，PAYOFF が計算される．このとき，後手は勝利条件を満たしているため $WinQ()$ の値が 1 となり，先手が勝利証券を満たしていないので $WinP()$ の値が 0 となる．そのため，先手の PAYOFF の値が -1，後手の PAYOFF の値が 1 であり，後手の勝ちとなる．

また，盤面のすべてのマスに石が置かれた場合は，両方のプレイヤが手を打つことができないため，PAYOFF が計

```
1 REL Animal0(x00, x01, x11, x21) = C(x00, x01) and R(x01, x11) and R(x11, x21)
2 REL Animal1(x12, x13, x04, x14) = C(x12, x13) and C(x13, x14) and R(x04, x14)
3 REL Animal2(x23, x33, x43, x44) = R(x23, x33) and R(x33, x43) and C(x43, x44)
4 REL Animal3(x30, x40, x31, x32) = R(x30, x40) and C(x30, x31) and C(x31, x32)
5 REL Animal4(x40, x21, x31, x41) = C(x40, x41) and R(x21, x31) and R(x31, x41)
6 REL Animal5(x00, x10, x11, x12) = R(x00, x10) and C(x10, x11) and C(x11, x12)
7 REL Animal6(x03, x13, x23, x04) = R(x03, x13) and C(x03, x04) and R(x13, x23)
8 REL Animal7(x32, x33, x34, x44) = C(x32, x33) and C(x33, x34) and R(x34, x44)
```

図 2 Toss における動物 Elly の定義

算される．このとき，先手後手はともに勝利条件を満たしていないため，$WinP()$ と $WinQ()$ の値が 0 となる．つまり，先手の PAYOFF の値も後手の PAYOFF の値も 0 であり，引き分けとなる．

初期状態

初期状態は，プレイヤの人数と初期盤面を定義したものである．プレイヤの人数が n 人のとき，PLAYERS $1, 2, .., n$ と表記する．また，初期盤面を以下のように表記する．

START S_0 ”

$$\begin{matrix} * & * & \dots & * \\ * & * & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \dots & * \end{matrix}$$

”

ここで S_0 は初期構造であり，” ”の中に初期盤面を定義する．初期盤面において，’$*$’ はマスの状態を表し，石が置かれていない場合は’$*$’ と表記し，そうでない場合は’P’ や’Q’ など，マスの状態を表す関係の記号で表記する．この表記方法を，縦および横に盤面の大きさと同じ数で繰り返す．

三並べにおけるプレイヤの人数を図 1 の 1 行目に示し，初期盤面を 31〜38 行目に示す．

4.2 Toss における拡張一般化三並べ

拡張一般化三並べを Toss の入力の形式に定義するために，いくつかの工夫を行う．まず，GTTT(p, q) にも TorusGTTT(m, n) にも使われる動物の定義である．動物の定義は，図 2 に示すように，横並びの関係 $R(x, y)$ および縦並びの関係 $C(x, y)$ の組み合わせですべて定義する．また，動物の 90 度回転および反転は，すべて個別に定義しなければならない．

GTTT(p, q) に対する定義

次に，GTTT(p, q) の特徴である手数の定義を行う．GTTT(p, q) の手数の定義の例を図 3 に示す．GTTT(p, q)

```
1  REL First() = all x not P(x)
2  RULE BlackFirst:
3    [a0| | - ]
4      ->
5    [a0|P{a0}| - ]
6      emb Q,P pre not WinQ() and First()
7  RULE Black:
8    [a0, a1| | - ]
9      ->
10   [a0, a1|P{a0}; P{a1}| - ]
11     emb Q,P pre not WinQ() and not First()
12 RULE White:
13   [a0, a1| | - ]
14     ->
15   [a0, a1|Q{a0}; Q{a1}| - ]
16     emb Q,P pre not WinP()
17 LOC 0 {
18   PLAYER 1 {
19     PAYOFF :(WinP()) - :(WinQ())
20     MOVES [BlackFirst -> 1]; [Black -> 1]
21   }
22   PLAYER 2 { PAYOFF :(WinQ()) - :(WinP()) }
23 }
24 LOC 1 {
25   PLAYER 1 { PAYOFF :(WinP()) - :(WinQ()) }
26   PLAYER 2 {
27     PAYOFF :(WinQ()) - :(WinP())
28     MOVES [White -> 0]
29   }
30 }
```

図 3 Toss における GTTT(2, 1) の手数の定義

の手数に関する定義は，複数の石を置く手と，初手かどうかを区別に関する 2 つを定義すればよい．複数の石を置く手の定義は，図 3 の 7〜16 行目に示すように，構造に手数の数と同じ数の変数および関係を書けば，1 つの手番でおける石の数を定義することができる．また，初手かどうかの区別は図 3 の 1〜11 行目および 17〜23 行目に示す．1 行目が初手かどうかを判定するための関係であり，すべてのマスに先手の石が置かれていないことを初手と定義する．先手の手の種類を 2〜6 行目と 7〜11 行目の 2 つに分けて，

```
1 REL Rt(x0, x3) = ex x1, x2 (R(x1, x0) and R(x2, x1) and R(x3, x2))
2 REL Ct(x0, x3) = ex x1, x2 (C(x1, x0) and C(x2, x1) and C(x3, x2))
3 REL RT(x, y) = R(x, y) or Rt(x, y)
4 REL CT(x, y) = C(x, y) or Rt(x, y)
```

図 4　Toss における大きさ 4 × 4 のトーラス盤面の定義

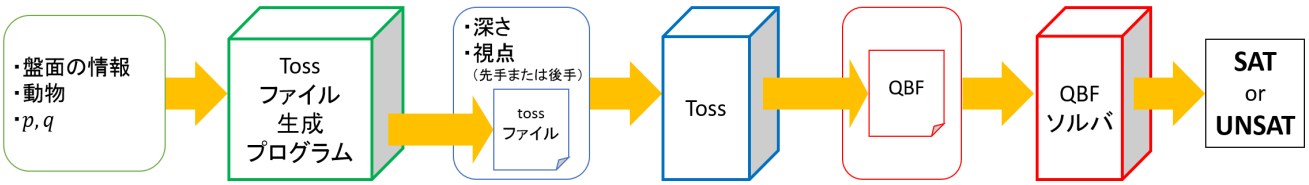

図 5　QBF ソルバを用いた拡張一般化三並べの勝敗判定の流れ

初手は初期盤面のときしか打つことができず，それ以外の手は初期盤面のときに打つことができないと定義する．

TorusGTTT(m, n) に対する定義

最後に，TorusGTTT(m, n) におけるトーラス盤面の定義を行う．図 4 に大きさ 4 × 4 のトーラス盤面の定義を示す．図 4 の 1 行目には盤面の右端を左マス，盤面の左端を右マスとしてつながっている関係を定義する．同様に 2 行目には盤面の上下のつながりを定義する．また，3 行目と 4 行目にはトーラス盤面におけるマスのつながりは普通の盤面のつながりに加えてトーラス盤面における盤面の端のつながりも考える関係を定義する．

ゲームを Toss を用いて変換するには，盤面の大きさや動物など，ゲームのパラメータによって toss ファイルを書かなければならい．様々なパラメータに対して拡張一般化三並べを QBF に変換するための toss ファイルの作成には非常に手間がかかる．そこで，たくさんのパラメータに対して一般化三並べを QBF に変換するために，toss ファイルを生成するプログラムを作成する．本プログラムに盤面の大きさ，動物，p, q 値など，単純な入力するだけで，Toss に対する入力ファイルを生成することができる．

QBF ソルバを用いた拡張一般化三並べの勝敗判定の全体の流れを図 5 に示す．最初に，toss ファイルを生成するプログラムに，盤面の大きさや動物など，拡張一般化三並べのパラメータを入力し，入力したパラメータに従った toss ファイルを生成する．次に，生成された toss ファイルとともにゲームの深さおよび視点の情報（先手または後手）を Toss に入力し，Toss によって QBF が書かれたファイルを生成する．最後に，QBF を QBF ソルバに入力することで，入力された視点から必勝となる（SAT）かどうか（UNSAT）を出力する．

5.　実験

QBF ソルバの性能を計るために，拡張一般化三並べの勝敗判定の時間測定において，既存の探索手法と QBF ソルバとの比較を行う．実験で用いた既存の探索手法は，枝刈り深さ優先探索（DFS）と証明数探索（PNS）である．QBF ソルバは，DepQBF と RAReQS を用いる．

QBF ソルバは，まず先手の視点からゲームを解き，先手必勝であれば，後手の視点からの判定は行わない．それに対して，先手必勝でない場合は，後手の視点からの判定を行い，後手必勝か引き分けかを判定する．また，前処理として拡張一般化三並べの QBF への変換に Toss，QBF の整形に bloqqer [2] を用いる．

本実験では，目標動物を 4 細胞動物 Tippy とする．拡張一般化三並べとして，3×3 と 4×4 の盤面におけるトーラス一般化三並べ，および GTTT$(1, 1)$, GTTT$(2, 1)$, GTTT$(2, 2)$ を用いる．また，ゲームの深さ（先手後手のターン数の合計）を 3 から 9 まで，またはゲームが終わるまでの手数とする．

表 1 に 3 × 3 の盤面における実験結果を示す．表 1 より，ほとんどのゲームにおいて証明数探索が最も速く勝敗判定を行うことができたが，GTTT$(2, 1)$ の深さ 3 においては，QBF ソルバの方が探索手法より速いことがわかる．また，QBF ソルバの中で DepQBF は RAReQS より速いことがわかる．

表 2 に 4 × 4 の盤面における実験結果を示す．表 2 より，先手必勝の証明には探索手法の方が優れていることがわかる．それに対して，引き分けの証明において，盤面の大きさが大きいほど QBF ソルバの DepQBF の方が探索手法より速くなることがわかる．

また，表 1 および表 2 からわかるように，トーラス GTTT は他のゲームより前処理に時間がかかる．これは，toss ファイルにおけるトーラス盤面の定義に多くの計算が必要であるからである．それに対して，トーラス盤面かどうかは，QBF ソルバの計算時間に対してあまり影響しないことがわかる．

6.　まとめ

本論文では拡張一般化三並べ，QBF ソルバを用いて，

表 1 大きさ 3×3 の盤面における実験結果 (秒)

ゲーム	深さ	勝敗判定	深さ優先探索	証明数探索	前処理	DepQBF	RAReQS
トーラス GTTT	3	引き分け	**0.005**	**0.005**	3.438	0.011	0.021
	4	引き分け	0.007	**0.006**	4.654	0.036	0.075
	5	引き分け	0.018	**0.011**	5.688	0.052	0.100
	6	引き分け	0.095	**0.031**	6.743	0.108	0.430
	7	先手必勝	**0.024**	0.039	7.745	0.386	5.754
	8	先手必勝	**0.034**	0.042	8.883	0.957	45.102
	9	先手必勝	**0.021**	0.037	9.952	0.614	19.565
GTTT$(1,1)$	3	引き分け	**0.005**	**0.005**	0.840	0.006	0.008
	4	引き分け	0.008	**0.005**	1.116	0.018	0.055
	5	引き分け	0.018	**0.009**	1.287	0.032	0.077
	6	引き分け	0.068	**0.030**	1.445	0.053	0.215
	7	引き分け	0.224	**0.106**	1.681	0.147	0.397
	8	引き分け	0.515	**0.235**	1.843	0.260	3.014
	9	先手必勝	0.295	**0.052**	2.093	0.304	4.944
GTTT$(2,1)$	3	引き分け	0.011	0.012	1.462	**0.006**	0.009
	4	引き分け	0.042	**0.017**	2.082	0.027	0.084
	5	引き分け	0.056	**0.015**	2.402	0.045	0.206
GTTT$(2,2)$	3	引き分け	0.020	**0.014**	2.020	0.016	0.030
	4	引き分け	0.039	**0.015**	2.093	0.045	0.346

表 2 大きさ 4×4 の盤面における実験結果 (秒)

ゲーム	深さ	勝敗判定	深さ優先探索	証明数探索	前処理	DepQBF	RAReQS
トーラス GTTT	3	引き分け	**0.007**	**0.007**	22.804	0.028	0.040
	4	引き分け	0.021	**0.008**	33.370	0.064	0.090
	5	引き分け	0.189	**0.055**	43.266	0.145	0.186
	6	引き分け	6.840	0.824	53.522	**0.395**	0.752
	7	先手必勝	**0.433**	0.939	64.044	4.326	6.484
	8	先手必勝	2.845	**1.011**	74.830	18.303	46.756
	9	先手必勝	2.176	**0.926**	85.132	48.156	32.931
GTTT$(1,1)$	3	引き分け	0.007	**0.006**	1.976	0.019	0.027
	4	引き分け	0.024	**0.008**	2.686	0.032	0.074
	5	引き分け	0.180	**0.054**	3.331	0.074	0.107
	6	引き分け	4.725	0.651	3.703	**0.131**	0.336
	7	引き分け	38.215	8.603	4.404	**3.838**	18.835
	8	引き分け	285.668	30.831	4.960	**14.330**	1,341.717
	9	先手必勝	314.885	**1.888**	5.854	75.877	65.500
GTTT$(2,1)$	3	引き分け	0.112	0.110	11.419	**0.044**	0.055
	4	後手必勝	1.277	1.429	15.623	0.993	**0.377**
	5	後手必勝	8.823	**2.182**	21.036	6.163	3.359
	6	後手必勝	68.609	**4.861**	26.084	93.777	27.714
	7	後手必勝	323.422	**6.614**	31.323	370.489	99.614
	8	後手必勝	846.876	**9.482**	36.223	1,342.394	4,853.000
GTTT$(2,2)$	3	先手必勝	0.119	0.130	15.703	0.134	**0.073**
	4	先手必勝	1.077	0.185	21.115	**0.088**	0.907
	5	先手必勝	9.543	**0.852**	31.280	13.522	3.063
	6	先手必勝	46.285	**1.156**	36.902	24.529	26.992
	7	先手必勝	172.559	**2.533**	46.794	216.890	58.782
	8	先手必勝	288.288	**2.732**	52.034	274.699	580.089

GTTT(p,q) および TorusGTTT(m,n) に対する勝敗判定を行った．QBF ソルバの性能を測定するために，既存の探索手法との計算時間の比較を行った．結果として，たいていのゲームにおいて探索手法のほうがより速く勝敗判定を行えるが，いくつかのゲームにおいては QBF ソルバの方がより速く引き分けの判定を行うことができた．

今後の課題として，それぞれの QBF ソルバの性質によって，ゲーム勝敗判定に対する有効性を解析する必要があると考えられる．また，ゲームから変換された QBF の性質を解析することで，ゲームの勝敗判定問題に有効な QBF ソルバの開発が行えると考えられる．

参考文献

[1] L.Victor Allis, Maarten van der Meulen, and H Jaap van den Herik. Proof-number search. *Artificial Intelligence*, Vol. 66, No. 1, pp. 91–124, 1994.

[2] Armin Biere, Florian Lonsing, and Martina Seidl. Blocked clause elimination for qbf. In *CADE'11*, pp. 101–115, 2011.

[3] Jens-P. Bode and Heiko Harborth. *Achievement games on Platonic solids*. Institute für Mathematik, Techn. Univ., 1997.

[4] Marco Cadoli, Andrea Giovanardi, and Marco Schaerf. An algorithm to evaluate quantified boolean formulae. *AAAI/IAAI*, Vol. 98, pp. 262–267, 1998.

[5] Enrico Giunchiglia, Massimo Narizzano, and Armando Tacchella. Qube: A system for deciding quantified boolean formulas satisfiability. In *Automated Reasoning*, pp. 364–369. Springer, 2001.

[6] Frank Harary. Achievement and avoidance games for graphs. *Ann. Discrete Math*, Vol. 13, pp. 111–120, 1982.

[7] Frank Harary. Achieving the skinny animal. *Eureka*, Vol. 42, pp. 8–14, 1982.

[8] Shigeki Iwata and Takumi Kasai. The othello game on an n× n board is pspace-complete. *Theoretical Computer Science*, Vol. 123, No. 2, pp. 329–340, 1994.

[9] Mikoláš Janota, William Klieber, Joao Marques-Silva, and Edmund Clarke. Solving qbf with counterexample guided refinement. In *SAT'12*, pp. 114–128, 2012.

[10] Łukasz Kaiser and Łukasz Stafiniak. Playing structure rewriting games. In *AGI'10*, 2010.

[11] Florian Lonsing and Armin Biere. Depqbf: A dependency-aware qbf solver. *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 7, pp. 71–76, 2010.

[12] Stefan Reisch. Gobang ist pspace-vollständig. *Acta Informatica*, Vol. 13, No. 1, pp. 59–66, 1980.

[13] Nándor Sieben. Hexagonal polyomino weak $(1, 2)$-achievement games. *Acta Cybernetica*, Vol. 16, No. 4, pp. 579–585, 2004.

[14] H Jaap Van den Herik, Jos WHM Uiterwijk, and Jack Van Rijswijck. Games solved: Now and in the future. *Artificial Intelligence*, Vol. 134, No. 1, pp. 277–311, 2002.

[15] ディプタラマ, 成澤和志, 篠原歩. 一般化三並べの拡張：一手 p 石. 情報処理学会論文誌, Vol. 55, No. 11, pp. 2344–2352, nov 2014.

[16] 伊藤大雄. パズル・ゲームで楽しむ数学: 娯楽数学の世界. 森北出版, 2010.

[17] 石黒裕也, ディプタラマ, 成澤和志, 篠原歩. トーラス盤面における一般化三並べの解析. In *GPW'15*, 2015. 投稿中.