BRICS

Basic Research in Computer Science

Maker-Maker and Maker-Breaker Games are PSPACE-Complete

Jesper Makholm Byskov

BRICS Report Series RS-04-14

ISSN 0909-0878 August 2004

See back inner page for a list of recent BRICS Report Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/04/14/`

Maker-Maker and Maker-Breaker games are PSPACE-complete

Jesper Makholm Byskov
BRICS*
Department of Computer Science
University of Aarhus, Denmark
jespermn@brics.dk

August, 2004

Abstract

We show that the problems of deciding the outcome of Maker-Maker and Maker-Breaker games played on arbitrary hypergraphs are PSPACE-complete. Maker-Breaker games have earlier been shown PSPACE-complete by Schaefer (1978); we give a simpler proof and show a reduction from Maker-Maker games to Maker-Breaker games.

1 Introduction

Maker-Maker and Maker-Breaker games are finite two-player perfect-information games played on a hypergraph $G = (V, E)$. The players take turns in playing an unplayed vertex. In Maker-Maker games, the first player, Maker1, wins if he plays all vertices in one edge, and the other player, Maker2, also wins if he plays all vertices in one edge. One example of a Maker-Maker game is Tic-Tac-Toe. A player has a winning strategy if he can win the game no matter how the other player plays. Given a hypergraph G it will always be the case that either one of the players has a winning strategy, or both have a drawing strategy meaning that

*Basic Research in Computer Science (www.brics.dk),
funded by the Danish National Research Foundation.

they can both force the game to end in a draw, which is what happens if all vertices have been played and none of the players have won. By an argument known as the strategy-stealing argument, Maker2 cannot have a winning strategy: suppose he has, then since having played an extra vertex will never do you any harm, Maker1 can just play any vertex v first and then play according to Maker2's winning strategy as though v had never been played. Maker-Breaker games are played similarly and the first player, Maker, also wins by playing all vertices of an edge. The second player, Breaker, however, wins by preventing Maker from winning. Thus, there are no draws and one of the players will have a winning strategy. Maker-Breaker games have the property that a game where both players have played some vertices can easily be transformed to a game with no played vertices by simply removing all played vertices and removing all edges containing at least one of the vertices played by Breaker. For simplicity, we always reduce Maker-Breaker games in this way. We can also easily transform between a Maker-Breaker game in which Maker starts and one where Breaker starts and vice versa by either adding a single vertex and an edge containing only this vertex (which Breaker will then have to play) or a single vertex which is added to all edges (which Maker will then have to play).

Definition 1. MAKER-MAKER is the problem of given a hypergraph G to decide if Maker1 has a winning strategy or Maker2 has a drawing strategy in the Maker-Maker game on G.

Definition 2. MAKER-BREAKER is the problem of given a hypergraph G to decide if Maker or Breaker has a winning strategy in the Maker-Breaker game on G.

2 Results

Theorem 3. MAKER-BREAKER *reduces to* MAKER-MAKER.

Proof. Let $G = (V, E)$ be an instance of MAKER-BREAKER and let d_1 and d_2 be two vertices not in V. Then we let $V' = V \cup \{d_1, d_2\}$, $E' = \{d_1\} \times E \cup \{(d_1, d_2)\}$ and $G' = (V', E')$. Now, G' is an instance of MAKER-MAKER in which Maker1 has a winning strategy if Maker has one on G and both players have a drawing strategy on G' if Breaker has a winning strategy on G: in any game, Maker1 has to play d_1, since it is contained in all edges. Now, Maker2 cannot win and he has to play d_2, or Maker1 would win by playing it. Then the rest of the game can be played according to the strategies for Maker and Breaker on G, since Maker2 cannot win. If Maker has a winning strategy Maker1 can also win, and if Breaker has a winning strategy Maker2 can make the game draw. $\square$

Schaefer [1] shows several games PSPACE-complete, among which are G_{pos}(POS DNF) which is equivalent to MAKER-BREAKER where Maker starts and G_{pos}(POS CNF) which is equivalent to MAKER-BREAKER where Breaker starts. We provide a simpler proof below.

Theorem 4. MAKER-BREAKER *is* PSPACE-*complete.*

Proof. MAKER-BREAKER is clearly in PSPACE. To show completeness, we will reduce QBF to it. Let $\forall x_1 \exists y_1 \dots \forall x_n \exists y_n \; C_1 \wedge \cdots \wedge C_m$ be an instance of QBF. We construct an instance of MAKER-BREAKER in which Breaker plays the role as satisfier meaning that he has to play one variable from each clause. We will slightly abuse notation and use x_i and y_j as both variables in the formula and vertices in the game and C_j for both clauses in the formula and vertices in the game. We let $V = \{u_1, u_1', \dots, u_n, u_n'\} \cup \{e_1, \dots, e_n\} \cup \{x_1, \bar{x}_1, \dots, x_n, \bar{x}_n\} \cup \{y_1, \bar{y}_1, y_1', \bar{y}_1', \dots, y_n, \bar{y}_n, y_n', \bar{y}_n'\}$ and $E = \bigcup_{i=1}^n (X_i \cup U_i \cup Y_i) \cup \bigcup_{j=1}^m C_j$, where

$$\begin{aligned}
X_i =& \{(u_1, e_1, \dots, u_{i-1}, e_{i-1}, x_i, \bar{x}_i)\}, \\
U_i =& \{(x_i, u_1, e_1, \dots, u_{i-1}, e_{i-1}, u_i, u_i'), (\bar{x}_i, u_1, e_1, \dots, u_{i-1}, e_{i-1}, u_i, u_i')\}, \\
Y_i =& \{(x_i, u_1, e_1, \dots, u_i, e_i, y_i, \bar{y}_i), (\bar{x}_i, u_1, e_1, \dots, u_i, e_i, y_i, \bar{y}_i)\} \cup \\
& \{(x_i, u_1, e_1, \dots, u_i, e_i, y_i, \bar{y}_i'), (\bar{x}_i, u_1, e_1, \dots, u_i, e_i, y_i, \bar{y}_i')\} \cup \\
& \{(x_i, u_1, e_1, \dots, u_i, e_i, y_i', \bar{y}_i), (\bar{x}_i, u_1, e_1, \dots, u_i, e_i, y_i', \bar{y}_i)\}, \\
C_j =& \{(u_1, e_1, \dots, u_n, e_n, l_{j1}, \dots, l_{j|C_j|}\}
\end{aligned}$$

and l_{jk} is the vertex corresponding to the kth literal in the clause C_j.

The idea is that the game is played in n rounds, each round consisting of the moves depicted in Table 1. If the game is played according to Table 1, Breaker will play a vertex from each of the X_is, U_is and Y_is so Maker cannot win with any of those, and he has to win by playing all vertices from one of the C_js. If the formula is satisfiable, no matter if Maker chooses x_i or $\bar{x}_i$, Breaker can choose y_i or $\bar{y}_i$ so that the formula is satisfied meaning that he will have played a vertex from each edge C_j. If the formula is unsatisfiable there must be at least one clause that Breaker

Move	Player	Choice
1	Maker	x_i or $\bar{x}_i$
2	Breaker	the other
3	Maker	u_i
4	Breaker	u'_i
5	Maker	e_i
6	Breaker	y_i or $\bar{y}_i$
7	Maker	the other
8	Breaker	y'_i or $\bar{y}'_i$, same as in round six

Table 1: Normal play in round i.

has not played any vertex from, but then Maker has played them all, so he wins.

We now argue that none of the players gain anything from not playing according to Table 1. Assume that the game is played according to Table 1 for the first $i-1$ rounds. Then Breaker has played at least one vertex in X_j, U_j and Y_j for $j < i$ and Maker has played $u_1, \ldots, u_{i-1}$ and $e_1, \ldots, e_{i-1}$. Let us first assume that Maker plays the first move according to Table 1, i.e., he plays either x_i or $\bar{x}_i$. Then Breaker has to play the other or Maker will win with the edge in X_i. Now, u_i is contained in all remaining edges, so Maker has to play it. Then as before, Breaker will have to play u'_i or Maker will win with one of the edges in U_i. Now, e_i is contained in all remaining edges, so Maker has to play it. In move six, Maker has played $u_1, \ldots, u_i, e_1, \ldots, e_i$ and x_i or $\bar{x}_i$, so he has all but two of the vertices in three of the edges in Y_i. Now, Breaker has to play either y_i or $\bar{y}_i$, or Maker can play one of them so that he only needs to play one vertex in two edges. In move seven, Maker is not forced to pick the other vertex, but since this forces Breaker to play the primed version of the variable he picked in move six, and the primed variables occur in no other edges, Maker loses nothing by picking the other vertex.

We only need to argue that Maker might as well play either x_i or $\bar{x}_i$ in his first move. Suppose that he does not play either. The only other vertex he can play is u_i, since it is in all the remaining edges. Then Breaker plays x_i. Now, e_i is in all remaining edges, except the second edge in U_i, so Maker has to play e_i, $\bar{x}_i$ or u'_i. If he plays $\bar{x}_i$, the players have the same vertices as they would have from a normal play in the first three moves, except that Breaker got to choose which of x_i and $\bar{x}_i$ he wanted, so this is not an advantage for Maker. If he instead plays u'_i, Breaker plays $\bar{x}_i$ and then Maker will have to play e_i and the play continues according to normal play from move five. This means that Maker neither got x_i nor $\bar{x}_i$, and u'_i is in no more clauses than the ones in U_i so this is not advantageous either. Finally if Maker plays e_i, Breaker just continues play from move six as if Maker had played $\bar{x}_i$. Maker can at any point play $\bar{x}_i$ in which case Breaker plays u'_i, but otherwise, play continues normally. Thus, Maker gains no advantage from not playing according to Table 1 in his first move. □

Corollary 5. MAKER-MAKER *is* PSPACE-*complete.*

Proof. MAKER-MAKER is clearly in PSPACE, and completeness follows from Theorems 3 and 4. □

References

[1] T. J. Schaefer. On the complexity of some two-person perfect-information games. *J. Comput. Syst. Sci.*, 16(2):185–225, Apr. 1978. `http://dx.doi.org/10.1016/0022-0000(78)90045-4`.

Recent BRICS Report Series Publications

RS-04-14 **Jesper Makholm Byskov.** ***Maker-Maker and Maker-Breaker Games are PSPACE-Complete*****. August 2004. 5 pp.**

RS-04-13 **Jens Groth and Gorm Salomonsen.** ***Strong Privacy Protection in Electronic Voting*****. July 2004. 12 pp. Preliminary abstract presented at Tjoa and Wagner, editors,** ***13th International Workshop on Database and Expert Systems Applications*****, DEXA '02 Proceedings, 2002, page 436.**

RS-04-12 **Olivier Danvy and Ulrik P. Schultz.** ***Lambda-Lifting in Quadratic Time*****. June 2004. 34 pp. To appear in** ***Journal of Functional and Logic Programming*****. This report supersedes the earlier BRICS report RS-03-36 which was an extended version of a paper appearing in Hu and Rodríguez-Artalejo, editors,** ***Sixth International Symposium on Functional and Logic Programming*****, FLOPS '02 Proceedings, LNCS 2441, 2002, pages 134–151.**

RS-04-11 **Vladimiro Sassone and Paweł Sobociński.** ***Congruences for Contextual Graph-Rewriting*****. June 2004. 29 pp.**

RS-04-10 **Daniele Varacca, Hagen Völzer, and Glynn Winskel.** ***Probabilistic Event Structures and Domains*****. June 2004. 41 pp. Extended version of an article to appear in Gardner and Yoshida, editors,** ***Concurrency Theory: 15th International Conference*****, CONCUR '04 Proceedings, LNCS, 2004.**

RS-04-9 **Ivan B. Damgård, Serge Fehr, and Louis Salvail.** ***Zero-Knowledge Proofs and String Commitments Withstanding Quantum Attacks*****. May 2004. 22 pp.**

RS-04-8 **Petr Jančar and Jiří Srba.** ***Highly Undecidable Questions for Process Algebras*****. April 2004. 25 pp. To appear in Lévy, Mayr and Mitchell, editors,** ***3rd IFIP International Conference on Theoretical Computer Science*****, TCS '04 Proceedings, 2004.**

RS-04-7 **Mojmír Křetínský, Vojtěch Řehák, and Jan Strejček.** ***On the Expressive Power of Extended Process Rewrite Systems*****. April 2004. 18 pp.**

RS-04-6 **Gudmund Skovbjerg Frandsen and Igor E. Shparlinski.** ***On Reducing a System of Equations to a Single Equation*****. March 2004. 11 pp. To appear in Schicho and Singer, editors,** ***ACM SIGSAM International Symposium on Symbolic and Algebraic Computation*****, ISSAC '04 Proceedings, 2004.**