Theoretical Computer Science 409 (2008) 364–381

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Complexity of question/answer games

Sarmad Abbasi [a,*], Numan Sheikh [b]

[a] 117-BB, DHA Phase IV, Lahore, Pakistan

[b] Department of Computer Science, Lahore University of Management Sciences, Opposite Sector "U", DHA, Lahore, Pakistan

ARTICLE INFO

Article history:
Received 25 February 2007
Received in revised form 28 January 2008
Accepted 10 August 2008
Communicated by A. Fraenkel

Keywords:
Computational complexity
Combinatorial games
Polynomial-time hierarchy
Perfect information games

ABSTRACT

Question/Answer games (Q/A games for short) are a generalization of the Rényi–Ulam game and they are a model for information extraction in parallel. A Q/A game, $G = (D, s, (q_1, \ldots, q_k))$, is played on a directed acyclic graph, $D = (V, E)$, with a distinguished start vertex s. In the ith round, Paul selects a set, $Q_i \subseteq V$, of at most q_i non-terminal vertices. Carole responds by choosing an outgoing edge from each vertex in Q_i. At the end of k rounds, Paul wins if Carole's answers define a unique path from the root to one of the terminal vertices in D.

In this paper we analyze the complexity of Q/A games and explore the notion of fixed strategies. We show that the problem of determining if Paul wins the game played on a rooted tree via a fixed strategy is in **NP**. The same problem is $\mathbf{\Sigma_2 P}$-complete for arbitrary digraphs. For general strategies, the problem is **NP**-complete if we restrict a two-round game to a digraph of depth three. An interesting aspect of this game is that it captures the even levels of the polynomial-time hierarchy when restricted to a fixed number of rounds; that is, determining if Paul wins a k-round game is $\mathbf{\Sigma_{2k-2} P}$-complete. The general version of the game is known to be **PSPACE**-complete [S. Abbasi, N. Sheikh, Some hardness results for Q/A games, Integers 7 (2007) G08]. In this paper we show that it remains **PSPACE**-complete even if each round consists of only two questions.

1. Introduction and definitions

A Q/A game is a *perfect information game* that is played between two persons Paul and Carole.[1] The game is motivated by the famous game, *Twenty Questions*, and is a generalization of the Rényi–Ulam game. Formally, a Q/A game [3] is $G = (D, s, (q_1, \ldots, q_k))$, where:

(1) $D = (V, E)$ is a directed acyclic graph.
(2) The vertex $s \in V$ is a distinguished vertex called the *root*. s is the starting vertex of the path that Paul is trying to find.
(3) There are k rounds in the game and in the ith round Paul is allowed to ask at most q_i questions.

We will refer to $\mathbf{q} = (q_1, \ldots, q_k)$ as the *question vector*. If the maximum number of questions in each round is the same; that is, $q = q_1 = q_2 = \cdots = q_k$, we denote the game by $G = (D, s, q, k)$. Furthermore, when the root of D is clear from the context we may ignore specifying it. For a vertex $v \in V$ let

$$N^+(v) = \{w : (v, w) \in E\} \quad \text{and} \quad d^+(v) = |N^+(v)|.$$

* Corresponding author.
E-mail addresses: sarmad_abbasi@yahoo.com (S. Abbasi), numan@lums.edu.pk (N. Sheikh).

[1] Joel Spencer [17] has suggested that the players in these types of search games be called Paul and Carole: Paul represents the great questioner Paul Erdös; whereas, Carole, being an anagram of "oracle", represents the notoriously obtuse oracle of Apollo at Delphi.

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.08.034

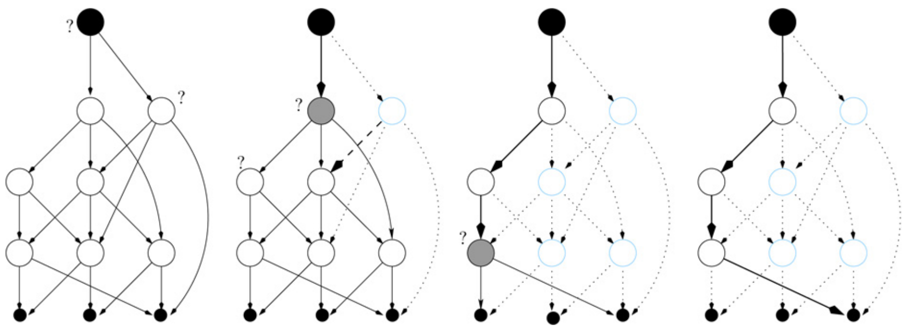

Fig. 1. A Q/A game: The grey vertex is the pseudo-root. The little black vertices are the terminal vertices. "?" represent Paul's questions. Bold arrows represent the choices made by Carole (Note that Carole wastes a question of Paul in round one.). The unreachable part of the graph is dotted out.

$I = \{v : d^+(v) > 0\}$ denotes the set of *internal vertices* of D and $T = V \setminus I$ denotes the set of *terminal vertices* of D. Throughout this section we assume that $G = (D, s, (q_1, \ldots, q_k))$ is the game under consideration and I and T are the internal and terminal vertices of D, respectively. In the ith round, Paul chooses a set $Q_i \subseteq I$, such that $|Q_i| \leq q_i$. If $v \in Q_i$, we say that Paul *inquires* or *asks* about the vertex v. Carole declares the value $f_i(v) \in N^+(v)$ for all $v \in Q_i$. When Carole declares $f_i(v)$, we say that she *responds* by $f_i(v)$ or Carole *points* or *leads* v to $f_i(v)$. When the context is clear, we simply say that Carole leads Paul to $f_i(v)$.

For simplicity, assume that Paul never repeats a question (or equivalently we can require that Carole, once having chosen an outgoing edge from a vertex v, consistently chooses the same edge when re-inquired about v). Let U_i denote the set of questions posed by Paul till the end of the ith round; that is, $U_i = \bigcup_{j=0}^{i} Q_i$. Let us define $f_{\leq i} : U_i \to V$ as

$$f_{\leq i}(v) = f_j(v) \quad \text{for } j \in Q_j.$$

After the end of the ith round, the pair $(U_i, f_{\leq i})$ completely determines the *state* or *position* of the game. For a path, $P = v_0, \ldots, v_t$, in D, we say that P is *consistent* with the position $(U_i, f_{\leq i})$ if $v_j \in U_i \Rightarrow v_{j+1} = f_{\leq i}(v_j)$ for all $0 \leq j < t$. When the position is clear from the context we say that P is consistent with Carole's answers. At the end of k rounds, Paul wins G if there is a unique path that is consistent with $(U_k, f_{\leq k})$, the final state the game.

An equivalent formulation of Q/A games is given by algorithms that *probe* information in parallel. An *input* for G is a function $f : I \to V$ where $f(v) \in N^+(v)$. Note that each input naturally defines a path P_f from the root to one of the terminal vertices of D. The following theorem is easy to prove.

Theorem 1 ([3]). *Paul wins $G = (D, s, \mathbf{q})$ if and only if there exists a decision tree algorithm that probes q_i values of f in the ith step and at the end of k steps outputs P_f.* □

We will always assume that all the vertices of D are reachable at the beginning of the game. Furthermore, we will assume that all internal vertices of D have out-degree at least two.

Consider G in position $(U_i, f_{\leq i})$ after i rounds. We call a vertex v *reachable*, if there exists a path from the root to v that is consistent with Carole's answers. An internal vertex v is called *open*, if it is reachable and Paul has not inquired about v. A vertex v is called the *pseudo-root*, if v is open and all predecessors of v are not open. We say that D is a *leveled graph* of *depth* d if the vertex set, V, can be partitioned into $d + 1$ sets $V_0, \ldots, V_d$, such that all the edges go from level V_i to V_j, where $i < j$. Whenever we speak about a digraph of depth d we tacitly assume that the graph is leveled. Note that $V_0 = \{s\}$, and all the vertices in V_d are terminal vertices.

It is interesting to contrast Q/A games with the k-round version of the Rényi–Ulam game [13,19,12]. In the Rényi–Ulam game, $U(n; (q_1, \ldots, q_k))$, Carole thinks of an x from the set $S = \{1, \ldots, n\}$. Paul tries to find this "x" by asking questions of the form: "Is $x \in A$?" where A can be any subset of $\{1, \ldots, n\}$. The game proceeds in k rounds and in the ith round Paul is allowed to ask q_i questions. After k rounds Paul wins if he can determine x. The outcome of this game depends only on the total number of questions asked, that is, $T = \sum_{i=1}^{k} q_i$. One can easily show that Paul wins $U(n; (q_1, \ldots, q_k))$ if and only if $2^T \geq n$. The Rényi–Ulam game becomes much more interesting if Carole is allowed to lie at times. Very interesting results are known for the Rényi–Ulam game with fixed number of lies [10,17]. Pelc [11] gives an excellent survey of the Rényi–Ulam game with lies. The Rényi–Ulam game is often compared [17] with the classical *Twenty Questions*. However, in Twenty Questions an interesting aspect is that it is important to know the answers of the first (say) five rounds in order to pose the sixth question and in the Rényi–Ulam original game without lies this is not the case. Q/A games are an attempt to model these more interesting games.

Many well-known combinatorial games are **PSPACE**-complete [6,8] and it would be interesting to characterize the complexity of restricted versions of these games in **PH** [18]. However, the compendium on complete problems in polynomial-time hierarchy [14,15] lists only one game [5,20], in the games and puzzles category, which is $\mathbf{\Sigma_2 P}$-complete. In fact, the compendium lists three problems, all in the logic category, that capture **PH**. These problems seem to be close variants of QBF.

Table 1
Comparison of results presented in [3] and this paper

Restriction on $(q_1, \ldots, q_k)$	Other restriction(s)	Previous characterization [3]	Improved characterization
$q_i = 1\ \forall i < k$ q_k arbitrary	None	**coNP**-hard	–
$k = 2$	None	**coNP**-hard	$\mathbf{\Sigma_2 P}$-complete
$k = 2$	Depth 3 graphs	–	**NP**-complete
Fixed k	None	–	$\mathbf{\Sigma_{2k-2} P}$-complete
None	Fixed strategy games	–	$\mathbf{\Sigma_2 P}$-complete
None	Fixed strategy games on trees	–	in **NP**
None	None	**PSPACE**-complete	–
$q_i = 2\ \forall i$	None	–	**PSPACE**-complete

An interesting aspect of Q/A games is that they allow us to capture the even levels of **PH**: k-round Q/A games are $\mathbf{\Sigma_{2k-2} P}$-complete. Moreover, determining if Paul wins a two-round game on a depth three graph is **NP**-complete. Other odd levels of **PH** can be captured with Q/A games with somewhat unnatural restrictions (see Section 7).

We use standard complexity theoretic notions [9]. Let us define the following languages:

$$\begin{aligned} \mathcal{P} &= \{\langle G\rangle : G \text{ is a Paul-win game}\} \\ \mathcal{P}_k &= \{\langle G\rangle : G \text{ is a } k\text{-round Paul-win game}\} \\ \mathcal{P}_k^d &= \{\langle G\rangle : G \text{ is a } k\text{-round Paul-win game on a digraph of depth } d\}. \end{aligned}$$

$\mathcal{C}$, $\mathcal{C}_k$ and $\mathcal{C}_k^d$ are analogously defined to be encodings of Carole-win games.

The complexity of Q/A games was initially studied in [3]. It was shown that $\mathcal{P}_2$ is **coNP**-hard. However this result is not tight, as we show in this paper that $\mathcal{P}_2$ is in fact $\mathbf{\Sigma_2 P}$-complete.

Note that if every round in a Q/A game consists of exactly one question then Paul wins an r-round game if and only if the length of the longest path in D is at most r. It was shown in [3] that determining if Paul wins a Q/A game in which all rounds, except the last round, consists of one question each is **coNP**-hard. The exact complexity of this mildly interesting problem is still open.

It may be noted that all the proofs in [3] reduce "hard" problems to Carole-win games. We continued our study of Q/A games in a hope to capture the polynomial-time hierarchy. The languages $\mathcal{P}_k$ and $\mathcal{C}_k$ are easily seen to belong to $\mathbf{\Sigma_{2k-2} P}$ and $\mathbf{\Pi_{2k-2} P}$ respectively. The main result of this paper is the proof of $\mathbf{\Sigma_{2k-2} P}$-completeness of $\mathcal{P}_k$. This result shows a reduction from QSAT_{2k-2} to $\mathcal{P}_k$ and therefore uses entirely different gadgetry from the one used in [3]. The ideas used in the reductions also allow us to show that $\mathcal{P}_2^3$ is **NP**-hard. We show that $\mathcal{P}_2^3$ is actually **NP**-complete thereby capturing an odd level of the polynomial-time hierarchy.

A much more complicated proof showing that $\mathcal{C}$ is **PSPACE**-complete was given in [3]. In this paper, we give a simpler proof showing that $\mathcal{P}$ is **PSPACE**-complete. In fact the new proof shows that determining the winner of Q/A games in which each round consists of at most two questions remains **PSPACE**-complete. This new proof uses ideas that are similar to the ones used in the proof of $\mathbf{\Sigma_{2k-2} P}$-completeness of $\mathcal{P}_k$.

Table 1 summarizes the comparison of results obtained in [3] and this paper.

All graphs considered in this paper are directed. In the figures, all edges are directed downwards and sometimes the arrowheads are not shown for simplicity. For certain classes of undirected graphs, such as rooted trees, there is a natural way to orient the edges to obtain a digraph. When referring to such an undirected graph, we will tacitly assume a natural orientation that will be clear from the context.

To prove our completeness results we will work with SAT, QSAT and QSAT_k. These problems are known to be **NP**-complete, **PSPACE**-complete and $\mathbf{\Sigma_k P}$-complete respectively. We will assume that all instances, ψ, of SAT are given in 3-CNF, and all instances of QSAT are of the form

$$\phi = \exists x_1 \forall x_2 \ldots \exists x_{n-1} \forall x_n \psi$$

where ψ is in 3-CNF.

For QSAT_k we will assume that the formula is given by

$$\phi = \exists \mathbf{x_1} \forall \mathbf{x_2} \cdots Q_k \mathbf{x_k} \psi$$

where the last quantifier, Q_k, is existential if k is odd and universal if k is even. Furthermore, ψ is in 3-CNF if k is odd and 3-DNF if k is even.

For precise definitions and proofs of completeness see [7,9].

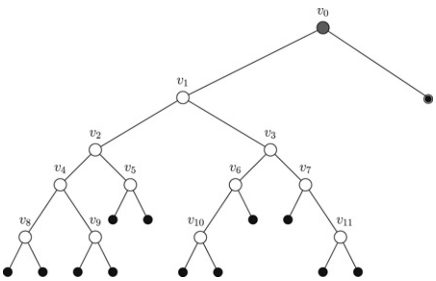

Fig. 2. A tree for which the question vector (1, 2, 2) does not have a fixed strategy.

1.1. Perfect play by Paul and Carole

We will be analyzing Q/A games when Paul and Carole play perfectly; therefore, we will assume that if Paul inquires about a vertex v and there is an edge from v to a terminal vertex, u, and to some internal vertex, w, then Carole never points v to u. In addition to this, we will allow Carole to make the following additional moves:

(1) Carole, at any point in the game, can delete a set of vertices S from the game, as long as the pseudo-root is not in S. In this case, we say that Carole restricts the game to $V \setminus S$.
(2) Carole may answer any number of questions without Paul inquiring about them. In this case, we will say that Carole answers *generously*.

It is readily seen that Paul wins a game, G, if and only if he wins G with these additional moves allowed to Carole. Indeed, these moves are only book keeping tools and do not give any additional power to Carole. However, they do simplify exposition of many proofs. We will also use the following fact throughout the paper:

Fact 2. *Paul wins $G = (D, s, (q_1, \ldots, q_k))$ if in the state $(U_{k-1}, f_{\leq k-1})$, after $k - 1$ rounds the number of open nodes is at most q_k.*

The rest of this paper is organized as follows: In Section 2 we discuss the notion of fixed strategies. Section 3 discuss Q/A games on trees. Section 4 proves the **NP**-completeness of two-round Q/A games on graphs of depth three. Section 5 proves the $\mathbf{\Sigma_{2k-2}P}$-completeness of k-round Q/A games. Section 6 proves the **PSPACE**-completeness of general Q/A games. Section 7 gives some concluding remarks along with a few interesting open problems.

2. Fixed strategies

We start with the simplest strategies that Paul can have for a Q/A game. Given a rooted digraph, D, and an integer, k, a *fixed strategy* for Paul is a labeling, $l : I \to \{1, \ldots, k\}$. Paul plays l if he inquires about all the open vertices labeled i in the ith round. Let $L_{<i}$ denote the vertices with labels $\{1, \ldots, i-1\}$, and $\mathbf{q} = (q_1, \ldots, q_k)$ be a question vector. We say that Paul wins $G = (D, s, \mathbf{q})$ via the fixed strategy, l, if for all possible answers to $L_{<i}$, the number of open vertices labeled i in the ith round is at most q_i. Given a game G (along with a question vector), it is not easy to determine if Paul wins G via the fixed strategy l. However, if we are given an input $f : I \mapsto V$, it is easy to determine in polynomial time how many questions will be asked in the ith round if Paul plays the fixed strategy l. We can simply simulate the strategy for $i - 1$ rounds and count the number of open vertices labeled i. This observation leads to the following theorem.

Theorem 3. *Determining if Paul wins a game using a fixed strategy is $\mathbf{\Sigma_2P}$-complete.*

Proof. A $\mathbf{\Sigma_2P}$-machine can existentially guess the labeling l and then universally check all the inputs to see if l requires at most q_i questions for rounds in round i, for each i. We observe that if G is a two-round game then all the strategies of Paul are fixed. In Section 5 (Theorem 12), we show that two-round Q/A games are $\mathbf{\Sigma_2P}$-hard thereby establishing the theorem. □

Consider the tree, T_1, shown in Fig. 2. There is a non-fixed strategy such that Paul wins $(T_1, (1, 2, 2))$. Paul inquires about v_1 in the first round. If Carole points Paul to v_2 then in the second round Paul inquires about v_2 and v_4. After getting these answers there is at most one open node in the subtree rooted at v_2. In the last round he can win by inquiring about the root, v_0, and this open node and win the game.

On the other hand if Carole points v_1 to v_3 then Paul asks about the root in the second round along with the vertex v_3. He wins the game as the number of open nodes after round two is at most two. This strategy is non-fixed as root is being asked in the second round in one case and the third round in the other.

To verify that Paul cannot win this game using a fixed strategy, we show that any labeling l,

$$l : \{v_0, \ldots, v_{11}\} \to \{1, 2, 3\},$$

is not a winning strategy for Paul.

For the sake of contradiction suppose l is a winning strategy for Paul. It is easy to see that if l is a winning strategy for Paul then it must satisfy the following facts.

Fact 4. *$l(v) = 1$ for at most one vertex $v \in \{v_0, \ldots, v_{11}\}$.*

Fact 5. *Any path of length five in T_1 must have one vertex labeled 1, two vertices labeled 2 and two vertices labeled 3.*

Applying these two facts to the paths v_0, v_1, v_2, v_4, v_8 and $v_0, v_1, v_3, v_7, v_{11}$ implies that either $l(v_1) = 1$, or $l(v_0) = 1$.

Case 1a: $l(v_1) = 1, l(v_0) = 2$.

If Carole points v_1 to v_2 then we note that in the subtree rooted at v_2 at most one vertex is labeled 2 and thus all other vertices must be labeled 3. If $l(v_2) = 2$ then if Carole points v_2 to v_4 then Paul has to ask three questions in the last round which is a contradiction. On the other hand, if $l(v_4) = 2$ then Paul has to ask v_2, v_5 and one of the vertices from v_8 and v_9 in the last round. Again we note that he has to ask three questions in the last round which is a contradiction. If $l(v_8) = 2$ then Paul has to ask about v_2, v_4, v_5, v_9 in the last round. This exhausts all the cases, as we have observed that at least one of the vertices in v_2, v_4, v_8 has to be labeled 2.

Case 1b: $l(v_1) = 1, l(v_0) = 3$.

Consider the case when Carole points v_1 to v_3. Note that if $l(v_3) = 3$ then by Fact 5 v_6, v_{10}, v_7 and v_{11} all are labeled 2 which is a contradiction. On the other hand, if $l(v_3) = 2$ then at most one of the vertices in the $\{v_6, v_7, v_{10}, v_{11}\}$ can be labeled 2. With loss of generality assume that v_6 is labeled 2 (other cases are similar). In this case both v_7 and v_{11} are labeled 3. If Carole points v_3 to v_7 then Paul has to ask three questions in the last round, which is a contradiction.

Case 2: $l(v_0) = 1$.

In this case consider the paths v_1, v_2, v_4, v_8 and v_1, v_3, v_7, v_{11}. By Fact 5, exactly two vertices on both these paths must be labeled 2. This implies that at least three vertices in the tree must be labeled 2, which again leads to a contradiction.

3. Q/A games on the trees

Perhaps, the simplest class of graphs on which a Q/A game has been analyzed is complete binary trees [1,2]. We briefly discuss the results here. Let $(T_n, (q_1, \ldots, q_k))$ be the Q/A game played on complete binary trees with n levels. In [1,2] it was shown that:

Theorem 6 *([1,2]). Paul wins $(T_n, (q_1, \ldots, q_k))$ if and only if*

$$\sum_{i=1}^{k} \lfloor \log_2(q_i + 1) \rfloor \geq n. \quad \square$$

The upper bound follows from a simple strategy in which Paul moves $t_i = \lfloor \log_2(q_i + 1) \rfloor$ levels by asking all the questions in the top t_i levels.

Note that the floor sign plays an important role here. Since $\lfloor \log_2(q + 1) \rfloor = t$ for all k satisfying $2^t - 1 \leq k \leq 2(2^t - 1)$, the result states that Paul does not gain "anything" by asking $2(2^t - 1)$ questions, twice as many, as opposed to $2^t - 1$ questions. In both cases, he simply moves t levels down in that round. The proof in [1,2] can be modified to obtain a result for all T_n^d; that is n-level d-ary trees.

Theorem 7. *Paul wins $(T_n^d, (q_1, \ldots, q_k))$ if and only if*

$$\sum_{i=1}^{k} \lfloor \log_d ((d - 1)q_i + 1) \rfloor \geq n. \quad \square$$

It would be an interesting problem to study Q/A games on arbitrary trees. Our understanding of fixed strategies on trees is also far from complete. However, it is easy to see that the problem is in **NP**.

Theorem 8. *The problem of determining if Paul wins with a fixed strategy on a tree is in* **NP**.

Proof. Given a tree T and a question vector $(q_1, \ldots, q_k)$, consider a fixed strategy l; that is, a labeling from the internal vertices of T to $\{1, \ldots, k\}$. For each, $i = 1, \ldots, k$, and each vertex, v, we compute $\mu(i, v)$ as follows:

$$\mu(i, v) = \begin{cases} 0, & \text{if } v \text{ is a terminal vertex;} \\ \sum_{w \in N^+(v)} \mu(i, w) + 1, & \text{if } l(v) = i; \\ \sum_{w \in N^+(v)} \mu(i, w), & \text{if } l(v) > i; \\ \max_{w \in N^+(v)} \mu(i, w), & \text{if } l(v) < i. \end{cases}$$

It is easy to see that $\mu(i, v)$ is the maximum number of questions asked by Paul in the ith round on the subtree rooted at v. Let s be the root of T. We can non-deterministically guess a strategy (labeling) and check if $\mu(i, s) \leq q_i$ for all i. $\square$

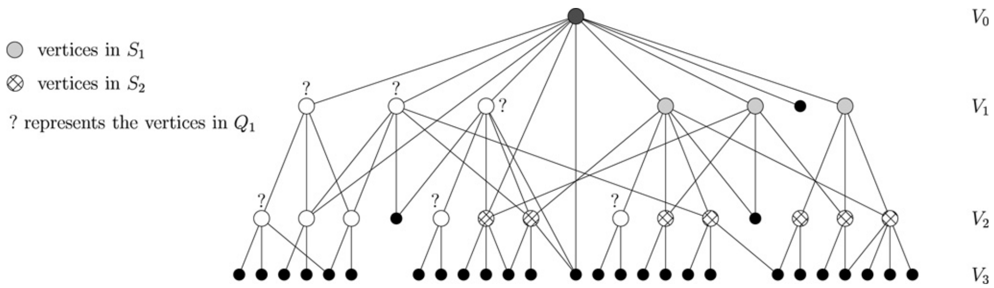

Fig. 3. The matching argument.

4. $\mathcal{P}_2^3$ is NP-complete

Determining if Paul wins a one-round game is trivial. If the depth of the graph is two and Paul has more than one round then the game is trivially a Paul-win game. On depth three graphs if Paul is given at least three rounds then he wins. Thus the first non-trivial question about Q/A games is to characterize the complexity of two-round games on depth three graphs. We have the following theorem:

Theorem 9. *$\mathcal{P}_2^3$ is* **NP***-complete.*

Proof. The proof is established by Lemmas 10 and 11. □

Lemma 10. $\mathcal{P}_2^3 \in$ **NP**.

Proof. Let $G = (D, s, (q_1, q_2))$ be a two-round Q/A game, where D is a digraph of depth three. Let I be the internal vertices of D. We argue that given a set, Q_1, of q_1 questions, it is possible to determine if Paul can win the game in the second round if he inquires about all the vertices in Q_1 in the first round. This is sufficient since a non-deterministic Turing machine can "guess" Q_1.

Let V_i be the vertices in the ith level of the graph. Note that $V_0 = \{s\}$. Suppose $s \in Q_1$. If Carole points s to a vertex in V_2, she loses as $q_2 > 0$. If she points s to a vertex in $V_1 \cap Q_1$, she also loses as Paul moves down two levels in one round. Hence, she must point s to a vertex $v \in V_1 \setminus Q_1$. If she leads him to a vertex $v \in V_1 \setminus Q_1$ the number of open vertices will be

$$g(v) = 1 + |(N^+(v) \setminus Q_1) \cap I|.$$

We can easily check in polynomial time for all vertices, $v \in V_1 \setminus Q_1$, if $g(v) \leq q_2$.

On the other hand, if $s \notin Q_1$, the problem can be solved by computing a maximum matching as follows: Carole is trying to find answers such that the number of open vertices in round two is maximized. All the vertices in

$$S_1 = (V_1 \setminus Q_1) \cap I$$

are open in round two, regardless of Carole's answers. All the vertices in

$$S_2 = \{x \in (V_2 \setminus Q_1) \cap I : (u, x) \in D \text{ for some } u \in S_1 \cup \{s\}\}$$

are also open regardless of Carole's answers (see Fig. 3).

Thus the only vertices whose "openness" depends on the answers in round one, are the ones in $B = (V_2 \cap I) \setminus (Q_1 \cup S_2)$. A vertex in B will be open if some vertex in $A = Q_1 \cap V_1$ is answered in its direction. Since we want to maximize the number of open vertices, we can compute a maximum matching between A and B in polynomial time (see [4]). Let m be the size of this matching. Paul wins in the second round if and only if $|S_1| + |S_2| + m + 1 \leq q_2$. □

Lemma 11. *$\mathcal{P}_2^3$ is* **NP***-hard.*

Proof. We show that SAT is LOGSPACE reducible to $\mathcal{P}_2^3$. Towards this end, we show that given an instance, ψ, of SAT, we can compute a Q/A game G_ψ in LOGSPACE such that ψ is satisfiable if and only if Paul wins G_ψ. Furthermore, G_ψ is a two-round Q/A game played on a digraph of depth three.

Let ψ be a formula on n variables $\{x_1, \ldots, x_n\}$ with m clauses; that is, $\psi = C_0 \wedge C_1 \wedge \cdots \wedge C_{m-1}$. The game G_ψ is played on a directed graph, D_ψ, and consists of only two rounds. In the first round Paul is allowed to ask $n + 1$ questions and in the second round he can ask 3 questions. Hence $G_\psi = (D_\psi, (n + 1, 3))$.

We assume that $m \geq 4$. The graph, D_ψ, corresponding to the game, G_ψ, consists of $1 + 4n + m$ internal vertices, with s as the root. On level one there are $n + m$ vertices; the first n are labeled by the variable $x_1, \ldots, x_n$ and the last m vertices correspond to clauses and are labeled $c_0, \ldots, c_{m-1}$. Level two consists of three vertices for each variable x_i, labeled d_i, T_i and F_i. The root is connected to all the vertices on level one. Each clause vertex c_j is connected to T_i (resp. F_i) if $x_i \in C_j$ (resp. $\bar{x}_i \in C_j$). Finally, each variable vertex x_i is connected to d_i, T_i and F_i. The last level consists of two terminal vertices for each vertex on level two. Fig. 4 shows the entire graph for

$$\psi_0 = (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\overline{x_1} \vee \bar{x}_2 \vee \bar{x}_4) \wedge (x_2 \vee x_3 \vee x_4).$$

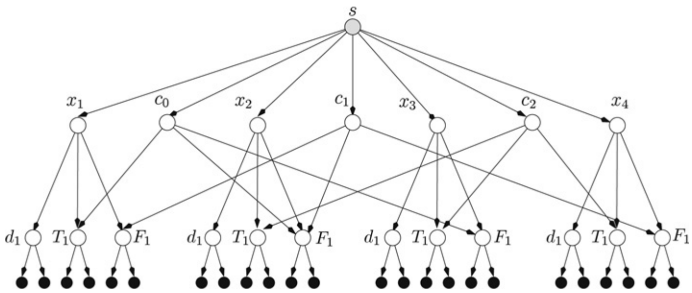

Fig. 4. The graph D_{ψ_0}.

We claim that Paul wins this game if and only if ψ is satisfiable. Suppose that ψ is satisfiable. Let $(t_1, \ldots, t_n)$ be a satisfying assignment for ψ. In the first round, Paul inquires about the root s. For each variable x_i Paul inquires about T_i if $t_i = 1$, and F_i if $t_i = 0$ (which Carole answers arbitrarily). The only choice Carole has is to respond to the root. If she leads the root to a clause vertex c_j then out of the three reachable vertices on level three, at least one is already answered. Hence, Paul can win the game by asking the remaining two vertices and c_j. On the other hand, if she leads him to a variable vertex x_i then out of the three reachable vertices from x_i, either T_i or F_i is answered. Once again Paul wins by inquiring about x_i, d_i and the un-inquired vertex from F_i and T_i.

Lastly, we show that if ψ is not satisfiable then Carole wins. Consider the set $S_i = \{x_i, F_i, T_i, d_i\}$. First, we claim that Paul must ask at least one question in each S_i in the first round, otherwise Carole wins. Indeed, if he does not inquire about any vertex in S_i then all the vertices in S_i are reachable (even if the root is inquired, Carole can lead the root to x_i), so he loses in round two. Next, we claim that he must inquire about the root in the first round. Since there are $n + m$ vertices on the second level, if he does not inquire about the root in the first round, at least $m - 1 \geq 3$ vertices are open on level one, and Paul loses the game as the root is also open. Hence, Paul must ask about the root and at least one vertex on each of the S_i's. This implies that Paul inquires about exactly one vertex on each S_i. The idea is that if Paul does not inquire about T_i or F_i, he is essentially forfeiting his right to set the variable x_i and this cannot help him. Consider the assignment $(t_1, \ldots, t_n)$, where $t_i = 1$ if Paul inquires about T_i and 0 otherwise.

Since ψ is not satisfiable, there is at least one clause C_j that is not satisfied by the assignment. Carole leads the root to c_j. Suppose $x_i \in C_j$. Since the assignment does not satisfy C_j, Paul did not inquire about T_i and T_i is open. Similarly, if $\bar{x_i} \in C_j$ and as C_j is not satisfied this means Paul inquired about T_i. Since he only inquires about one vertex in $S_i = \{T_i, F_i, x_i, d_i\}$, he did not inquire about F_i and F_i is open. Thus, all the reachable vertices from c_j are open. This means that there are at least four open vertices and Paul loses. □

5. Q/A games and the polynomial-time hierarchy

In this section we prove our main theorem.

Theorem 12. *$\mathcal{P}_k^{k+3}$ is $\mathbf{\Sigma_{2k-2}P}$-complete.*

Proof. It is to see easy that $\mathcal{P}_k$, hence $\mathcal{P}_k^{k+3}$ is in $\mathbf{\Sigma_{2k-2}P}$.

The rest of this section gives the proof that $\mathcal{P}_k^{k+3}$ is $\mathbf{\Sigma_{2k-2}P}$-hard. Towards this end, we show that given an instance of QSAT_{2k-2}; that is, ϕ with $2k - 2$ alternating quantifiers starting with an existential quantifier we can compute in LOGSPACE a k-round game, G_ϕ, such that Paul wins G_ϕ if and only if ϕ is true. Furthermore, the graph, D_ϕ on which G_ϕ is played, is a $k + 3$-level graph.

Let

$$\phi = \exists \mathbf{x_1} \forall \mathbf{y_1} \cdots \exists \mathbf{x_{k-1}} \forall \mathbf{y_{k-1}} \psi.$$

Here we assume that ψ is a Boolean formula in 3-DNF; that is,

$$\psi = D_1 \vee D_2 \vee \cdots \vee D_m$$

and each clause, D_i, is a conjunction of exactly three variables. We also assume that after each quantifier the number of variables appearing in ϕ is exactly the same; that is, $\mathbf{x_i} = \{x_{i,1}, \ldots, x_{i,n}\}$ and $\mathbf{y_i} = \{y_{i,1}, \ldots, y_{i,n}\}$ for $i = 1, \ldots, k - 1$. This can be accomplished by adding dummy variables if needed.

The main ingredient of the proof is a gadget that allows both Carole and Paul, in the ith round, to select the assignment of their respective variable sets; namely, $\mathbf{x_i}$ and $\mathbf{y_i}$.

For each existentially quantified variable $x \in \mathbf{x_i}$ there is a set $V(x)$ including two special vertices T and F, representing the two possible assignments of the variable. Paul sets the value of x by asking about T or F. For each universally quantified variable $y \in \mathbf{y_i}$ there is a set $W(y)$ which includes a special vertex C that leads to two vertices labeled T and F. When Paul inquires about C he invites Carole to set the value of y.

In the ith round Paul has $2n + 1$ questions. He can use these questions as follows.

(1) Paul asks one question on the pseudo-root.

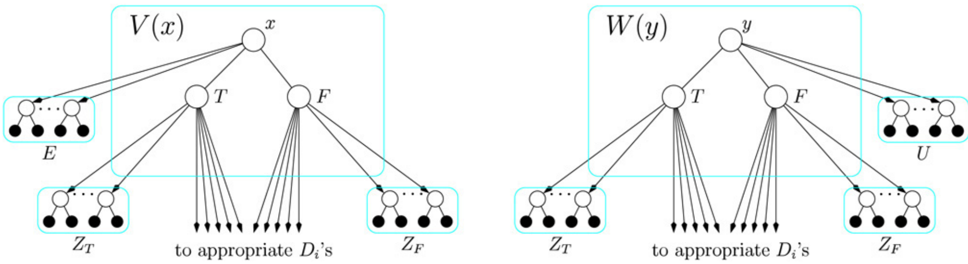

Fig. 5. Selectors $V(x)$ and $W(y)$.

(2) Paul asks n questions for assigning a value to each of the n existentially quantified variables.
(3) Paul asks n questions such that he allows Carole to assign the value to each of the n universally quantified variables.

The construction of D_ϕ has to be so robust that both Paul and Carole essentially have no other option to play the game. The game is set up so that the number of open vertices in the last round is very large if the assignment of the variables made by Paul and Carole makes ψ false. The elaborate gadgetry assures that Paul and Carole play in this predictable fashion.

The game G_ϕ has k rounds. The first $k-1$ rounds consist of $2n+1$ questions and the last round consists of L questions, where the number L will be described later. The game can formally be described as follows:

$$G_\phi = (D_\phi, (\underbrace{2n+1, \ldots, 2n+1}_{k-1 \text{ times}}, L)).$$

Let us define:

$$X = \bigcup_{i=1}^{k-1} \mathbf{x}_i \quad \text{and} \quad Y = \bigcup_{i=1}^{k-1} \mathbf{y}_i;$$

that is X and Y are the set of existentially and universally quantified variables in ϕ, respectively. Throughout this section we will be referring to the formula ϕ. We assume that $n \geq 4$ for the rest of this section. Let us choose two parameters t and z where $t \geq 4$ and $z = mt + 2kn$. In fact, any large enough values for these parameters yield a correct result. We take $L = 7z + 2n(k-1) + (m-1)t + 4$.

We now describe the graph D_ϕ on which the game is played.

5.1. *The gadgets*

In order to describe the construction of D_ϕ, we will introduce some basic gadgets.

5.2. *The clause nodes*

For each clause, D_i, we make t internal nodes labeled $d_i^1, \ldots, d_i^t$ in the graph. Each one of these nodes is connected to two terminal vertices. For simplicity we denote the set $\{d_i^1, \ldots, d_i^t\}$ by D_i. Let D denote all the clause nodes; that is,

$$D = \bigcup_{i=1}^{m} D_i.$$

We have sets Z_T, Z_F, E, U and R consisting of $2z, 2z, 4z, 8z$ and $3z$ internal nodes, respectively. Each one of these nodes is also connected to two terminal vertices. Note that our graph will contain exactly one copy of each of these sets.

5.2.1. *Value selectors*

For each variable $x \in X$ we create a *value selector* (or just *selector*), $V(x)$, consisting of three vertices as shown in Fig. 5. If Paul asks about a vertex labeled T (resp. F), he sets $x = 1$ (resp. $x = 0$). The vertices T and F are connected to the clause vertices as follows:

If x appears in a clause D_i then T is connected to all the clause nodes in D_i. Similarly, if $\bar{x}$ appears in D_i then F is connected to all the vertices of D_i. T (resp. F) is also connected to all the vertices in Z_T (resp. Z_F). The vertex x is connected to all internal the vertices in E.

Suppose a variable, x, occurs in D_i and Paul wants to set this variable to 1. He inquires about T in $V(x)$ and no matter how Carole responds, she can only make at most one node in D_i reachable from x. In the game, Paul will try to satisfy a clause D_i and if he succeeds then he will be able to make all but at most three nodes in D_i unreachable, thereby reducing the number of questions he needs in the last round.

Fact 13. *If at most* $2n+1$ *vertices in* $V(x)$ *are answered and* T, F *and* x *are not answered then Paul cannot win* $V(x)$ *by asking* L *questions in one round.*

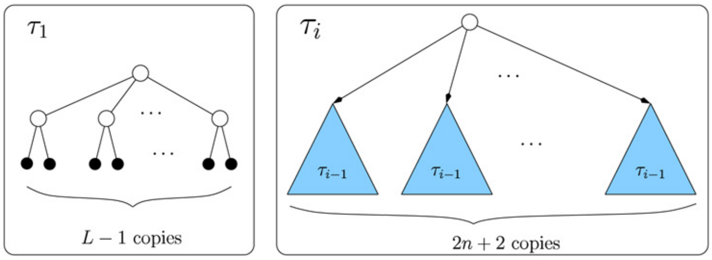

Fig. 6. Construction of the tree τ_i.

Proof. The total number of nodes in Z_T, Z_F and E is $8z$. As all the vertices in $Z_T \cup Z_F \cup E$ are reachable from x, the number of open nodes is at least $8z - 2n - 1 > L$, he loses. □

Fact 14. *If T or F is answered in $V(x)$ then Paul wins $V(x)$ in one round by asking L questions.*

Proof. If T (resp. F) is answered then the number of reachable nodes to Z_T (resp. Z_F) is at most one. Hence the total number of reachable nodes from x is $2z + 4z + mt + 3 < 7z < L$, and Paul wins. □

The value selector, $W(y)$, for each $y \in Y$ is shown in Fig. 5. When Paul inquires about y, he is allowing Carole to set the value of y in a natural way; that is, if she points y to T (resp. F) then she sets $y = 1$ (resp. $y = 0$).

If y (resp. $\bar{y}$) appears in a clause D_i then F (resp. T) is connected to all the vertices in D_i. T (resp. F) is connected to all the nodes in Z_T (resp. Z_F). Lastly, y is connected to all the internal nodes in U.

Note that here the connections from T and F to the clause nodes are opposite to that of the existentially quantified variables. Suppose that y appears in a clause D_i and Carole choose $y = 1$ by pointing y in $W(y)$ to T. In this case, all the nodes in D_i become unreachable from y.

Fact 15. *If at most $2n + 1$ questions are answered in $W(y)$ and y is not answered in $W(y)$ then Paul cannot win $W(y)$ by asking L questions in one round.*

Proof. The total number of nodes in U is $8z$. As y is not answered, all the nodes in U are reachable. Hence the number of open nodes is at least $8z - 2n - 1 > L$ and Paul loses. □

Fact 16. *If y is answered in $W(y)$ then Paul wins $W(y)$ in one round by asking L questions.*

Proof. If y is pointing to a node in U then Paul needs only a single question to win. If y leads to T (resp. F), the total number of open nodes reachable from T (resp. F) is at most $2z + mt + 1 < L$ and he wins. □

5.2.2. Trees

Let τ_1 denote a tree containing one internal vertex connected to $L - 1$ internal vertices. Each of these internal vertices is connected to two terminal vertices. We let τ_i be a tree constructed inductively by adding a root and pointing it to $2n + 2$ copies of τ_{i-1}, (see Fig. 6). It is easy to check that the number of vertices in τ_i is

$$(2n+2)^{i-1}(3L-2) + \frac{(2n+2)^{i-1} - 1}{2n+1} = O(n^i m).$$

Let us collect some simple facts about the games played on a τ_t.

Fact 17. *Paul wins on τ_t by asking one question in each of the $t - 1$ rounds and L questions in the last round.* □

This game is robust in the sense that Paul cannot win this game in $t - 1$ rounds, even if he has a reasonable number of questions in each round. More precisely:

Fact 18. *For $t \geq 2$, Paul cannot win on τ_t by asking $2n + 1$ questions in the first $t - 2$ rounds and L questions in the last round.*

Proof. For $t = 2$, there are more than L open vertices in τ_2, hence Carole wins. If $t \geq 3$ and Paul asks $2n + 1$ questions in the first round, Carole can always point the root (regardless of his asking about the root) to a copy of τ_{t-1} such that he has not asked any questions in that copy. She can apply this strategy inductively to win the game on this τ_{t-1}. □

5.2.3. Protectors

Let $x \in X$ and $V(x)$ be a value selector for x. We define a *protector of order one*, $P_1(x)$, for x to be $V(x)$ itself. A protector of order i, $P_i(x)$, consists of a root that is connected to $P_{i-1}(x)$. It is also connected to $2n$ copies of τ_i and $4n + 3$ copies of τ_{i-1}. Similarly, we define $W(y)$ to be a protector, $Q_1(y)$, of order one. A protector of order i, $Q_i(y)$, consists of a root that is connected to $Q_{i-1}(y)$. It is again connected to $2n$ copies of τ_i and $4n + 3$ copies of τ_{i-1} (see Fig. 7).

Note that unlike the sets Z_T, Z_F, E, U and R which have one global copy in the graph, each protector has its own copy of the trees τ_i.

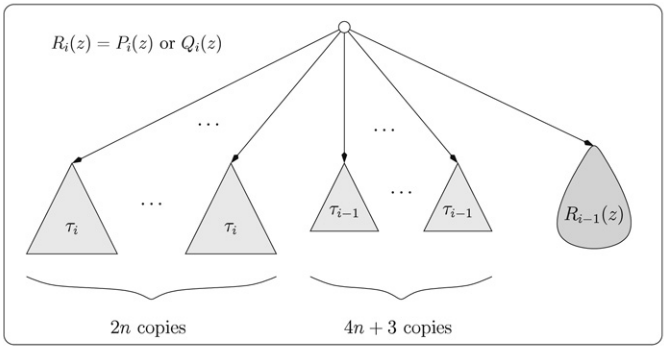

Fig. 7. $P_i(x)$ and $Q_i(y)$ have the same recursive construction.

We say that $P_1(x)$ (resp. $Q_1(y)$) is *almost clean* if there are at most $2n+1$ answered nodes in $P_1(x)$ (resp. $Q_1(y)$). Furthermore, these nodes do not include x, T or F (resp. y). We say $P_i(x)$ is *almost clean* if there are at most $2n+1$ nodes answered in $P_i(x)$ (resp. $Q_i(y)$). All these nodes are in $P_{i-1}(x)$ (resp. $Q_{i-1}(y)$) and $P_{i-1}(x)$ (resp. $Q_{i-1}(y)$) is also almost clean.

Equivalently, we can say that $P_i(x)$ is almost clean if there are at most $2n+1$ nodes reachable from x that are already answered. Furthermore, the nodes belong to Z_T, Z_F, E or the clauses that are reachable from x.

Fact 19. *If $P_t(x)$ is almost clean, Paul cannot win on $P_t(x)$ by asking $2n+1$ questions in the first $t-1$ rounds and L questions in the last round.*

Proof. We have already proved this claim for the base case in Fact 13. For $t > 1$, if he does not inquire about any vertex in one of the τ_t's, Carole leads him to that copy of τ_t which she wins by Fact 18. If he does not inquire about $P_{t-1}(x)$, she points the root to $P_{t-1}(x)$ and wins by induction. Lastly, if he does not inquire about the root then by the pigeonhole principle there are at least $2n+1$ copies of τ_{t-1}'s in which he has not inquired about any vertex. These copies along with the root make a τ_t. Carole can restrict the game to this τ_t and win by Fact 18. □

$P_t(x)$ contains $V(x)$, which is a sensitive part of $P_t(x)$. If Paul has an advantage on $V(x)$, of a single question, he can win the game on $P_t(x)$. More precisely:

Fact 20. *If at least one of T or F is answered in $V(x)$ then Paul wins on $P_t(x)$ by asking $2n+1$ questions in the first $t-1$ rounds, and L questions in the last round.*

Proof. For $t = 1$ we have already proved this claim in Fact 14. Now, we prove the claim for general $P_t(x)$ by induction. Paul asks the root and one question each on the roots of τ_t's. If Carole leads him to a τ_t, he moves two levels down and the game is played on a subtree of τ_t which is a τ_{t-1} and he wins by Fact 17. For the same reason she cannot send him to any τ_{t-1}'s. Lastly, if she sends him to $P_{t-1}(x)$, he wins by induction. □

Fact 21. *If $Q_t(x)$ is almost clean, Paul cannot win on $Q_t(x)$ by asking $2n+1$ questions in the first $t-1$ rounds and L questions in the last round.*

Proof. Similar to Fact 19. □

Fact 22. *If y is answered on $W(y)$ then Paul wins on $Q_t(y)$ by asking $2n+1$ questions in the first $t-1$ rounds and L questions in the last round.*

Proof. Similar to Fact 20. □

Lastly, we note that $P_i(x)$ and $Q_i(y)$ have $O(n^i)$ vertices and depth $i+2$.

5.3. The graph D_ϕ

D_ϕ contains k *special vertices*, $r_1, \ldots, r_k$, where r_1 is the root of D_ϕ and each r_i, for $i = 1, \ldots, k-1$, is connected to the following vertices:

- The roots of $4n+3$ copies of τ_{k-i}.
- The root of the protector $P_{k-i}(x_{i,j})$ for each variable $x_{i,j} \in \mathbf{x}_i$ and the root of the protector $Q_{k-i}(y_{i,j})$ for each variable $y_{i,j} \in \mathbf{y}_i$.
- The special vertex labeled r_{i+1}.

The special vertex r_k is connected to all vertices labeled y in $W(y)$ for all $y \in Y$, and all the vertices labeled T and F in $V(x)$. Note that r_k is not connected to any vertices labeled x in $V(x)$ for any $x \in X$. The vertex r_k is also connected to all the vertices in R (see Fig. 8).

We note that the size of the graph D_ϕ is $mn^{O(k)}$ and its depth is $k+3$. The description of this graph can be computed in LOGSPACE.

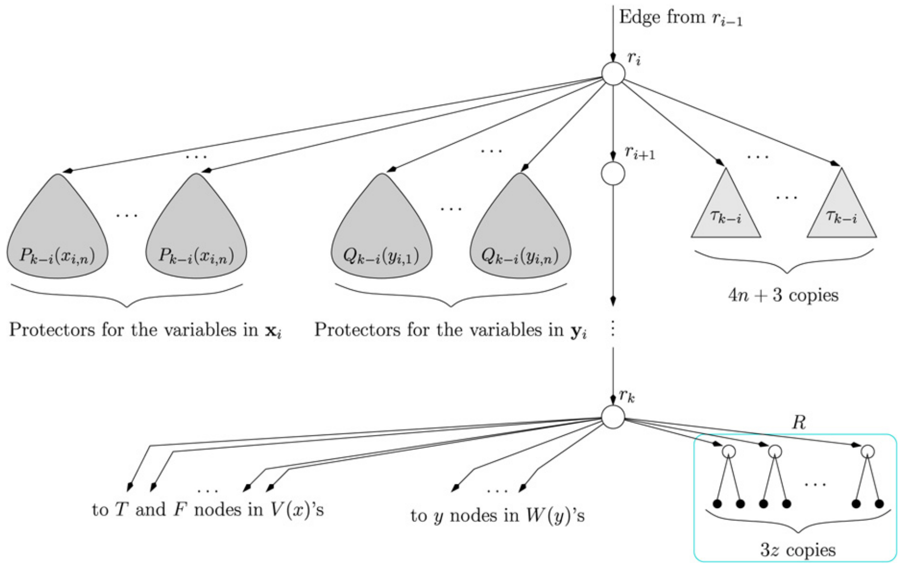

Fig. 8. Part of the graph D_ϕ .

5.4. Paul's strategy

Let us assume that ϕ is true. We now describe Paul's strategy for the game G_ϕ. For $i = 1, \ldots, k$, Paul will maintain the following invariants at the start of the ith round.

(1) The vertex r_i is the pseudo-root.
(2) For all $t < i$, there are assignments $\alpha_t : \mathbf{x}_t \to \{0, 1\}$ and $\beta_t : \mathbf{y}_t \to \{0, 1\}$ such that:
  (a) if $\alpha_t(x_{t,j}) = 1$ (resp. 0) then T (resp. F) is answered in $V(x_{t,j})$.
  (b) Each $y_{t,j}$ in $W(y_{t,j})$ is answered.
  (c) If $\beta_t(y_{t,j}) = 1$ then y is answered to T in $W(y_{t,j})$.

These invariants clearly hold for $i = 1$. To make sure that the invariants hold for $i+1$, in the ith round Paul inquires about r_i. He takes an assignment $\boldsymbol{\alpha}_i : \mathbf{x_i} \to \{0, 1\}$ and for each $x_{i,j}$ if $\boldsymbol{\alpha}_\mathbf{i}(x_{i,j}) = 1$ (resp. 0), he inquires about T (resp. F) in $V(x_{i,j})$. He inquires about y in each of $W(y_{i,j})$, thus inviting Carole to pick her assignment β_i for each $y_{i,j}$. After getting her answers, he defines:

$$\beta_i(y_{i,j}) = \begin{cases} 1, & \text{if she answers } y \text{ to } T \text{ in } W(y_{i,j}); \\ 0, & \text{otherwise.} \end{cases}$$

Note that he is asking a total of $2n + 1$ questions. The only way that the invariants can fail to hold is if Carole does not point r_i to r_{i+1}. We show that in this case Paul wins. Indeed, if she points r_i to a copy of τ_{k-i}, he wins in the next $k-i$ rounds by Fact 17. If she points r_i to the root of some protector, $P_{k-i}(x_{i,j})$, as he has already asked a question in $V(x_{i,j})$, he wins by Fact 20. If she leads him to some $Q_{k-i}(y_{i,j})$, then, as he has already asked y in $W(y_{i,j})$, he wins by Fact 22. Thus, the only choice she has is to lead him to r_{i+1}.

At the end of $k - 1$ rounds r_k is the pseudo-root. Furthermore, the play has determined an assignment $(\alpha_1, \beta_1, \ldots, \alpha_{k-1}, \beta_{k-1})$ of all the variables. We make the following critical observation:

Lemma 23. *If a clause D_i is satisfied by $(\alpha_1, \beta_1, \ldots, \alpha_{k-1}, \beta_{k-1})$ then at most three nodes are reachable in D_i.*

Proof. Suppose D_i is satisfied and it contains a universally quantified variable y. Then Carole chose $y = 1$, hence pointed y in $W(y)$ to T, thus making all the nodes in D_i unreachable from y. The same is the case, if a universal variable y appears negated in D_i.

If D_i contains an existentially quantified variable x then Paul asked about T from which all the nodes in D_i were reachable. Since Carole responds to T, she can only point it to one of the nodes in D_i.

Since D_i can contain at most three existentially quantified variables, at most three of the nodes in D_i can remain open. □

Lemma 24. *Let us assume that Paul plays as described above and Carole leads to r_k in round $k - 1$. Suppose the assignment $(\alpha_1, \beta_1, \ldots, \alpha_{k-1}, \beta_{k-1})$ satisfies ψ, then the number of open nodes in the last round is at most*

$$7z + 2n(k-1) + (m-1)t + 4 = L.$$

Proof. We count the total number reachable nodes from r_k that are open. There are $2z + 2z + 3z = 7z$ nodes in Z_T, Z_F and R. Hence the number of open nodes in $Z_T \cup Z_F \cup R$ cannot exceed $7z$.

For each $x \in X$ we count the number of open nodes in $V(x) \setminus (Z_T \cup Z_F \cup D)$. Since Paul has inquired about either T or F in $V(x)$, at most one node out of T and F is open. Note that there are no edges from r_k to x in $V(x)$, therefore, x and all the nodes in E are unreachable from r_k. Thus there are $(k-1)n$ open nodes, reachable from r_k in

$$\left(\bigcup_{x\in X} V(x)\right) \setminus (Z_T \cup Z_F \cup D).$$

For each $y \in Y$, we count the number of open nodes in each $W(y) \setminus (Z_T \cup Z_F \cup D)$. As y is answered in $W(y)$, we have two cases: If y is answered to a node in U then exactly one node is open in U. If y is answered to T (resp. F) then T (resp. F) is the only open node in $W(y) \setminus (Z_T \cup Z_F \cup D)$. Hence, the total number of open nodes, reachable from r_k in

$$\left(\bigcup_{y\in Y} W(y)\right) \setminus (Z_T \cup Z_F \cup D)$$

is again at most $(k-1)n$.

Lastly, we count the number of open nodes in D. As $(\alpha_1, \beta_1, \ldots, \alpha_{k-1}, \beta_{k-1})$ satisfies ψ, at least one of the clauses, D_i, is satisfied. By Lemma 23, the total number of open clause nodes in D_i is at most three. Hence, the number of open nodes in D is at most $(m-1)t + 3$. The node r_k is also open, therefore, $7z + 2n(k-1) + (m-1)t + 4$ upper bounds the number of open nodes reachable from r_k. □

We can now conclude the following:

Lemma 25. *If ϕ is true, Paul wins G_ϕ.* □

5.5. Carole's strategy

Let us now assume that ϕ is false. In this case, Paul has no incentive to play as described in Section 5.4. Here we show that he can only deviate from the strategy very slightly and these deviations do not help him in winning the game.

We call a subgraph of D_ϕ , *clean*, if no vertices in that subgraph have been answered. A subgraph is *dirty*, if it is not clean. We will show that before the ith round, Carole can maintain the following invariants:

(1) For all $i \le k$, r_i is the pseudo-root.
(2) $P_{k-t}(x_{t,j})$ and $Q_{k-t}(x_{t,j})$ are clean for all $t \ge i$.
(3) All the trees connected to r_t, with $t \ge i$, are clean.
(4) For all $t < i$, there are assignments $\alpha_t : \mathbf{x}_t \to \{0, 1\}$ and $\beta_t : \mathbf{y}_t \to \{0, 1\}$ such that:
  (a) if $\alpha_t(x_{t,j}) = 0$ (resp. 1) then T (resp. F) is not answered in $V(x_{t,j})$.
  (b) For $t < i$, the vertex y, in $W(y_{t,j})$, points to T if $\beta_t(y_{t,j}) = 1$, and to F otherwise. Furthermore, both T and F are not answered in $W(y_{t,j})$.
  (c) All of Z_T, Z_F, E, U, R and D are clean.

We show that if Paul tries to destroy these invariants, Carole wins the game. Note that, the invariants are trivially true before the first round. Given that the invariants are true before the ith round, we have the following claims:

Fact 26. *If Paul does not inquire about r_i, he loses.*

Proof. Note that r_i is connected to $4n + 3$ copies of τ_{k-i} and he can only make $2n + 1$ of them dirty, by asking a question. This leaves at least $2n + 2$ copies of τ_{k-i} clean. These trees, along with r_i, make a copy of τ_{k-i+1}. Carole now restricts the game to τ_{k-i+1} and wins by Fact 18. □

Fact 27. *If Paul does not inquire about one of the nodes in*

$$P_{k-i}(x_{i,j}) \setminus (Z_T \cup Z_F \cup E \cup D)$$

he loses.

Proof. Note that in this case at the end of the ith round $P_{k-i}(x_{i,j})$ is almost clean and he loses by Fact 19. □

Fact 28. *If Paul does not inquire about at least one node*

$$Q_{k-i}(y_{i,j}) \setminus (Z_T \cup Z_F \cup U \cup D)$$

he loses.

Proof. Note that in this case at the end of the ith round $Q_{k-i}(y_{i,j})$ is almost clean and he loses by Fact 21. □

These facts account for all the $2n+1$ questions that Paul has in round i. Therefore, he cannot make any other part of the graph dirty. Furthermore, Paul must ask exactly one question on each of the $P_{k-i}(x_{i,j})$ and $Q_{k-i}(y_{i,j})$ for all j.

Carole can maintain the invariants by defining:

$$\alpha_i(x_{i,j}) = \begin{cases} 1, & \text{If Paul inquires about } T \text{ in } W(x_{i,j}); \\ 0, & \text{otherwise.} \end{cases}$$

It is possible that Paul has not inquired about any vertex in $V(x_{i,j})$ and has inquired about some other vertex in $P_{k-i}(x_{i,j})$. In this case, he has forfeited his right to assign a value to $x_{i,j}$. The way Carole has defined $\alpha_i(x_{i,j})$ she has acted as if Paul has assigned $x_{i,j}=0$.

Fact 15 shows if Paul has asked about T or F in $Q_{k-i}(y_{i,j})$ then he loses.

She now chooses an assignment β_i, and points y, in $W(y_{i,j})$, to T if $\beta(y_{i,j}) = 1$, and F otherwise. Note, that Carole can answer $y_{i,j}$ generously even if Paul has not inquired about it. Lastly, she points r_i to r_{i+1} and maintains the invariants.

We need the following fact at the end of $k-1$ rounds.

Fact 29. *If $(\alpha_1, \beta_1, \ldots, \alpha_{k-1}, \beta_{k-1})$ does not satisfy a clause D_i then all the nodes in D_i are open.* □

The following lemma finishes our proof of Theorem 12.

Lemma 30. *If all the clauses are not satisfied then the number of open nodes in the last round is*

$$7z + (2n(k-1)) + mt > L.$$

Proof. Similar to the analysis of Lemma 24. The only difference is that Paul may not have inquired about any vertices in some $V(x_{i,j})$. However, in this case the number of reachable vertices cannot decrease. □

6. General Q/A games

We cannot use the construction given in the previous section for D_ϕ to show that Q/A games are **PSPACE**-complete. The reason is that the number of vertices in D_ϕ is at least $mn^{O(k)}$, where k is the number of alternations, and for arbitrary QSAT this is too large. A much more complicated proof of **PSPACE**-completeness, based on completely different gadgetry was given in [3]. Interestingly the proof shows that determining if Carole wins a Q/A game is **PSPACE**-complete. However, that proof does not show that Q/A games are **PSPACE**-complete even when the number of questions in each round is restricted to two. We provide a somewhat different construction of the graph, D_ϕ , to prove the following theorem:

Theorem 31. *$\mathcal{P}$ is* **PSPACE***-complete. In fact, determining if Paul wins a Q/A game in which each round has* 2 *questions is also* **PSPACE***-complete.*

It can easily seen that $\mathcal{P} \in$ **PSPACE** [3]. The rest of this section is devoted to proving that $\mathcal{P}$ is **PSPACE**-hard. We show that given an instance of QSAT, that is, a formula

$$\phi = \exists x_1 \forall x_2 \ldots \exists x_{n-1} \forall x_n \psi$$

where ψ is in 3-CNF. More precisely,

$$\psi = (C_1 \wedge C_2 \wedge \cdots \wedge C_m),$$

where C_i is a disjunction of exactly three variables. Note that n is even; that is, the last quantifier is universal.

We create a Q/A game

$$G_\phi = (D_\phi, r_1, (\underbrace{2, 2, \ldots, 2}_{n \text{ times}}, 6)),$$

such that Paul wins G_ϕ if and only if ϕ is satisfiable. Moreover, this reduction would be computable in LOGSPACE. At the end we will comment on how to modify the game so that instead of having six questions in the last round we have additional five rounds of two questions each.

Let X (resp. Y) denote the set of existentially (resp. universally) quantified variables in ϕ.

6.1. The gadgets

In this section we describe the gadgets used in the graph D_ϕ .

6.1.1. Value selectors, strands and double strands

For each variable $x \in X$, we create a *value selector* (or just *selectors*), $V(x)$, on four vertices as shown in Fig. 9. The vertices of this selector have the following interpretation. If Paul asks about a vertex labeled T_i (resp. F_i), he sets $x=0$ (resp. $x=1$).

We use a different kind of value selector, $W(y)$, for each $y \in Y$ as shown in Fig. 9. When Paul inquires about C, he allows Carole to set the value of y in a natural way; that is, if she points C to T then she sets $y=1$, and if she points it to F then she sets $y=0$.

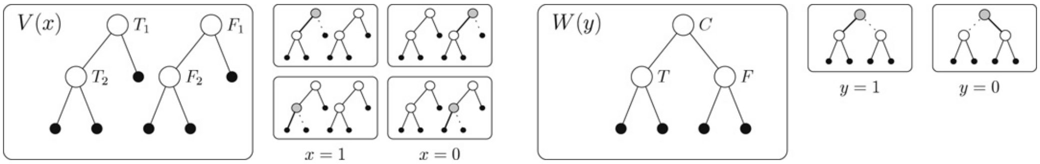

Fig. 9. Selectors $V(x)$ and $W(y)$ and how to set the values of the variables.

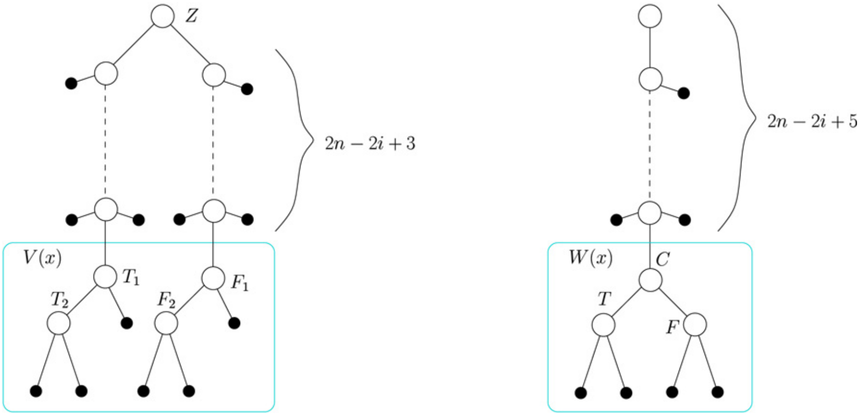

Fig. 10. Protection strands: $P_i(x)$ and $Q_i(x)$.

A *strand*, S_t, of length t consists of t internal vertices connected as a directed path. Each internal vertex is also connected to a terminal vertex and the last internal vertex is connected to two terminal vertices. A *double strand*, D_t, of length t is constructed by taking a vertex Z and connecting it to two copies of S_t. The following facts are easy to verify for strands and double strands.

Fact 32. *Paul wins on S_t if and only if the total number of questions that Paul can ask (regardless of the number of rounds) is at least t.* □

Fact 33. *If Paul asks two questions in each of the first t rounds and six questions in the last round he loses on D_{2t+5}.*

Proof. We use induction on t with the base case being trivial. If in the first round, Paul does not ask one question on both strands, Carole will point Z to the strand on which he has not asked any question. He has a total of $2(t-1)+6=2t+4$ questions, and the game is played on a strand of length $2t+5$; therefore, he loses by Fact 32. If he asks a question on both strands then Carole can answer these questions so that the game is now played on D_{2t+4}, which he loses by induction. □

Fact 34. *Let $t \geq 1$. Suppose one of the questions is already answered on D_{2t+5}. If Paul asks two questions each in the first t rounds and six questions in the last round he wins on D_{2t+5}.*

Proof. If the answered question is Z then the game is played on a strand of length S_{2t+5} which Paul wins by Fact 32. If the answered question is on one of the strands then Paul can inquire about the root and one question on the other strand. As both strands have an answered question, the game is now played on a strand of length $2t+4$, regardless of Carole's answer to Z. Paul has enough questions now to win this game. □

6.1.2. Protectors

For $x \in X$, a protector of order i, $P_i(x)$, consists of a double strand, $D_{2n-2i+3}$, whose last two internal vertices are connected to T_1 and F_1 in $V(x)$, respectively. We observe that $P_i(x)$ is just $D_{2n-2i+5}$. For $x \in Y$, a protector of order i, $Q_i(x)$, consists of a strand, $S_{2n-2i+5}$, whose last internal vertex is connected to the vertex C in $W(x)$ (see Fig. 10).

Fact 35. *Suppose that at most one question is answered on $Q_i(x)$, and this question is either T or F in $W(x)$. If Paul asks two questions each in the first $n-i$ rounds, and six questions in the last round he loses on $Q_i(x)$.*

Proof. If either T or F in $W(x)$ is answered then $Q_i(x)$ is a strand of length $2n-2i+7$. □

Fact 36. *Suppose C is answered on $Q_i(x)$. If Paul asks two questions each in the first $n-i$ rounds and six questions in the last round he wins on $Q_i(x)$.*

Proof. If C is answered then $Q_i(x)$ is the same as a strand of length $2n-2i+6$. □

The proofs of the following facts are the same as Fact 33 and Fact 34, respectively.

Fact 37. *If Paul asks two questions each in the first $n-i$ rounds and six questions in the last round he loses on $P_i(x)$.* □

Fact 38. *Suppose one of the questions is already answered on $P_i(x)$. If Paul asks two questions each in $n-i$ rounds and six questions in the last round he wins on $P_i(x)$.* □

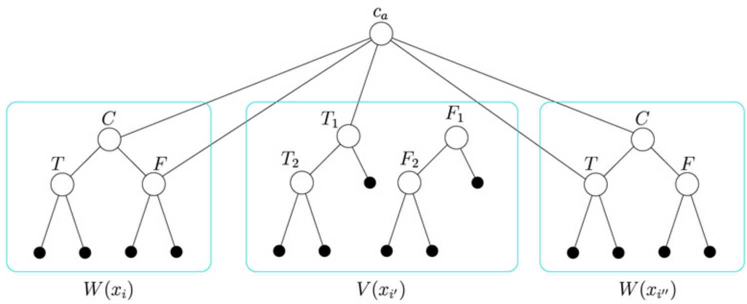

Fig. 11. Clause vertex for $C_a = (\bar{x}_i \vee x_{i'} \vee x_{i''})$; x_i and $x_{i''}$ are existentially quantified and $x_{i'}$ is universally quantified.

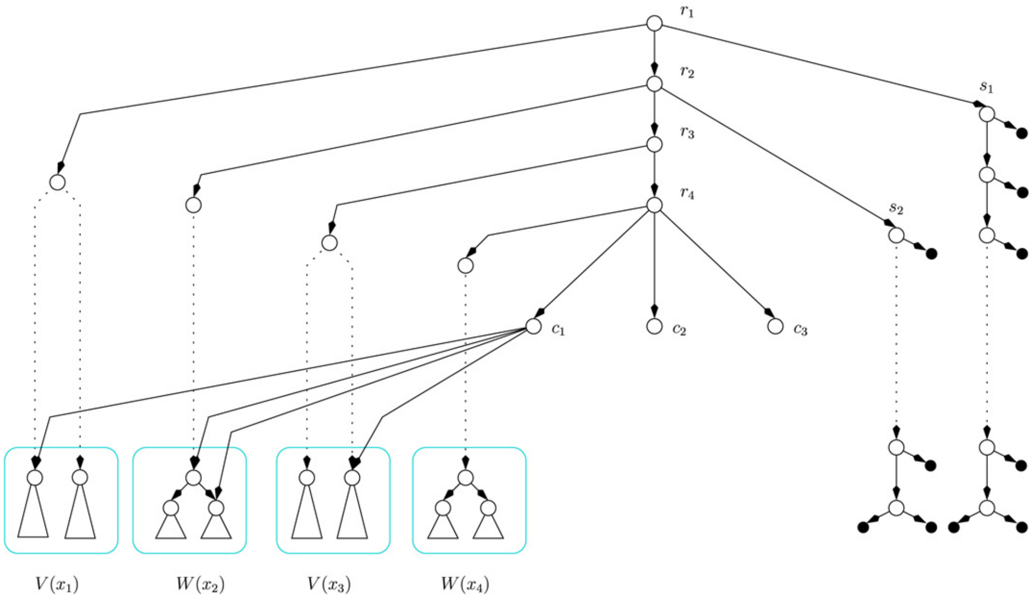

Fig. 12. The graph D_{ϕ_0} (only partial graph is shown for clarity).

6.2. The graph D_ϕ

D_ϕ will have n *special vertices* labeled $r_1, \ldots, r_n$ with r_1 being the root of D_ϕ. For each variable $x_i \in X$, we will have a protector, $P_i(x_i)$; and for each $x_i \in Y$, we will have a protector Q_i.

We also have n *protection strands*, S^i, of length $2n-2i+4$ for $i = 1, \ldots, n$, rooted at vertex s_i, respectively. For $i = 1, \ldots, n$, each vertex r_i is connected to r_{i+1} (if $i < n$) and the root of $P_i(x_i)$ (resp. $Q_i(x_i)$) if $x_i \in X$ (resp. $x_i \in Y$). The vertex r_i is also connected to s_i on the protection strand S^i, for all $i \le n$. There will be one vertex labeled c_j for each of the clauses C_j. The vertex r_n is also connected to the m clause vertices. A clause vertex c_j is connected to the value selectors as shown in Fig. 11. If an existentially quantified variable, $x_i \in X$, appears in a clause C_t (resp. $\bar{x}_i$ appears in C_t) then there is an edge from c_t to the vertex labeled T_1 (resp. F_1) in $V(x_i)$. If a universally quantified variable, $x_i \in Y$, appears in C_t (resp. $\bar{x}_i$ appears in C_t) then there is an edge from c_t to the vertex labeled C and T (resp. F) in $W(x_i)$.

Fig. 12 shows the partial graph D_{ϕ_0} for

$$\phi_0 = \exists x_1 \forall x_2 \exists x_3 \forall x_4 (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_2 \vee x_3 \vee x_4).$$

6.3. Paul's strategy

Let us assume that ϕ is true. For $i = 1, \ldots, n+1$, Paul will maintain the following invariants at the start of the ith round.

(1) If $i \le n$ then r_i is the pseudo-root. If $i = n+1$ then some clause vertex, c_a, is the pseudo-root.
(2) There is an assignment $\alpha_{i-1} : \{x_1, \ldots, x_{i-1}\} \to \{0, 1\}$, such that:
  (a) for $x_j \in X$, if $\alpha_{i-1}(x_j) = 1$ (resp. 0) then T_1 (resp. F_1) is answered in $V(x_j)$.
  (b) For $x_j \in Y$, if $\alpha_{i-1}(x_j) = 1$ (resp. 0) then C is answered to T (resp. F) in $W(x_j)$.

This invariants clearly hold for $i = 1$. To make sure that the invariants hold for $i + 1$, Paul has to extend α_{i-1} to the next variable, x_i. He uses one question to inquire about r_i. If $x_i \in X$, and he wishes to set $\alpha_i(x_i) = 1$ (resp. 0), he inquires about T_1

(resp. F_1) in $V(x_i)$. If $x_i \in Y$ he inquires about C in $W(x_i)$, thus inviting Carole to pick her assignment for x_i. He defines:

$$\alpha_i(x_i) = \begin{cases} 0, & \text{if she answers } C \text{ to } F \text{ in } W(x_i); \\ 1, & \text{if she answers } C \text{ to } T \text{ in } W(x_i). \end{cases}$$

The only way the invariants can fail to hold is if Carole does not point r_i to r_{i+1}. We show that in this case Paul wins. Indeed, if she points r_i to $P_i(x_i)$ (resp. $Q_i(x_i)$), he wins in the next $n - i + 1$ rounds by Fact 38 (resp. 36). If she points r_i to s_i then he wins by Fact 32. Similarly, in round n she can only point him to a clause vertex c_a.

Lemma 39. *If Paul plays as described above and Carole leads to a clause vertex, c_a, then the number of open vertices in the last round is*

$$7 - \lambda,$$

where λ is the number of literals in C_a that are made true by the assignment, $(\alpha_1, \ldots, \alpha_n)$.

Proof. Let $x_i \in X$ that is, x_i be an existentially quantified variable. If x_i appears as a literal in C_a then c_a has edges to T_1 in $V(x_i)$. If $\alpha_i(x_i) = 1$ then Paul has inquired about T_1 and only one vertex is open in $V(x_i)$. If $\alpha_i(x_i) = 0$ then both T_1 and T_2 are open. The analysis is similar if $\bar{x}_i$ appears in C_a.

On the other hand for $x_i \in Y$, if x_i appears C_a then c_a has edges to C and T in $W(x_i)$. If $\alpha_i(x_i) = 1$ then only T is open. On the other hand if $\alpha_i(x_i) = 0$ then both T and F are open. Again the analysis is similar if $\bar{x}_i \in C_a$. Therefore, the number of open vertices reachable from c_a is exactly $\lambda + 2(3 - \lambda) = 6 - \lambda$. The total number of open vertices including c_a is $7 - \lambda$. □

Hence, we conclude that if ϕ is true Paul wins.

6.4. Carole's strategy

Assume that ϕ is false. In this case, we show that before the ith round Carole can maintain the following invariants.

(1) If $i \le n$ then r_i is the pseudo-root. If $i = n + 1$ then some clause vertex, c_a, is the pseudo-root.
(2) The protectors for the variables x_j are clean for all $j \ge i$.
(3) The strands S^j for all $j \ge i$ are clean.
(4) None of the clause vertices is answered.
(5) There is an assignment $\alpha_{i-1} : \{x_1, \ldots, x_{i-1}\} \to \{0, 1\}$, such that for $j < i$:
  (a) If $x_j \in X$ then $\alpha_{i-1}(x_j) = 0$ (resp. 1) then both T_1 and T_2 (resp. F_1 and F_2) are not answered in $V(x_j)$.
  (b) If $x_j \in Y$ and $\alpha_{i-1}(x_j) = 0$ (resp. 1) then C in $W(x_j)$ points to T (resp. F). Furthermore, both T and F are not answered in $W(x_j)$.

In the ith round, if Paul does not inquire about r_i, or some vertex in S^i, then he loses as the vertex r_i along with S^i makes a path of length $2n - 2i + 5$ (Fact 32). If $x_i \in X$ and Paul does not inquire about some vertex in $P_i(x_i)$, he loses (Fact 37). Similarly, if $x_i \in Y$ and he does not inquire about any vertex in $Q_i(x_i)$, or inquires about T or F in $Q_i(x_i)$, he loses (Fact 35).

This accounts for the two questions Paul has in round i and shows that he cannot make any other part of the graph dirty. If $x_i \in X$ then Carole can maintain the invariants by defining:

$$\alpha_i(x_i) = \begin{cases} 0, & \text{If Paul inquires about } F_1 \text{ or } F_2 \text{ in } W(x_{i,j}); \\ 1, & \text{otherwise.} \end{cases}$$

If $x_i \in Y$, she can choose an assignment $\alpha_i(x_i)$ and points C to T (resp. F), if $\alpha_i(x_i) = 1$ (resp. $\alpha_i(x_i) = 0$), in $W(x_i)$. In the penultimate round she points r_n to a clause vertex c_a. The following lemma shows that if ϕ is false then Carole wins G_ϕ.

Lemma 40. *The number of open vertices in the last round is at least $7 - \lambda$, where λ is the number of literals in C_a made true by the assignment $(\alpha_1, \ldots, \alpha_n)$.*

Proof. Similar to the analysis of Lemma 39. The only difference is that Paul may not have inquired about any vertices in some $V(x_{i,j})$. However, in this case the number of reachable vertices cannot decrease. □

6.5. Reducing the maximum number of questions

We briefly discuss how to reduce the maximum number of questions in any round of the game to two. We replace the last round with five rounds of two questions each. Each strand of length t is replaced by a strand of $t + 4$. Similarly, each double strand of length t is replaced by a double strand of length $t+4$. Each clause vertex, c_j, is replaced with a *clause-gadget* which is a complete binary tree of depth four. There are 16 vertices, $c_j^1, \ldots, c_j^{16}$, on the fourth level of this tree. Each c_j^l is connected to the value selectors in the same way that c_j would have been connected. The analysis of this game for the first n rounds is exactly the same as the one given above. Only the last five rounds need to be re-analyzed, which is done below.

If ϕ is satisfiable then the set of open vertices, O, reachable from each c_j^l has at most five vertices. Paul starts from the root of the clause-gadget and in each round he uses one of the questions to inquire about the pseudo-root of the tree, and the other one on a vertex in O, and wins.

In case that ϕ is not satisfiable, the number of open vertices in O, reachable from each c_j^l, is at least six. By Theorem 6, if he does not inquire about a vertex in the tree in any one of these five rounds, he loses. Hence, he has only one question to spare in each round. As $|O| \ge 6$, he loses.

This completes the proof of Theorem 31. □

Table 2
Summary of complexity theoretic results known about fixed round Q/A games

k (rounds)	depth $k+1$	depth $k+2$	depth $k+3$
2	**NP**	$\mathbf{\Sigma_1}$-hard $\mathbf{\Pi_1}$-hard	$\mathbf{\Sigma_2}$-complete
$k \geq 3$	$\mathbf{\Sigma_{2k-5}}$-hard $\mathbf{\Pi_{2k-5}}$-hard	$\mathbf{\Sigma_{2k-3}}$-hard $\mathbf{\Pi_{2k-3}}$-hard	$\mathbf{\Sigma_{2k-2}}$-complete
$k+\frac{1}{2}$	–	$\mathbf{\Sigma_{2k-3}}$-hard $\mathbf{\Pi_{2k-3}}$-hard	$\mathbf{\Pi}_{2k-1}$-complete

7. Conclusion

Q/A games show a rich variety from a complexity theoretic point of view. One can define a game on $k+\frac{1}{2}$ rounds. In such a game, a set of questions, Q_1, to be asked in the first round, is included as a part of the game. Carole responds to Q_1 in the first *half* round and the next k rounds continue between Paul and Carole as usual. We can show that $\mathcal{P}_{k+\frac{1}{2}}$ is $\mathbf{\Pi}_{2k-1}\mathbf{P}$-complete and capture the odd levels of **PH** also. Some readers may find this way of capturing the odd levels of **PH** unnatural. This prompts the following question:

Problem 7.1. Can the odd levels of **PH** be captured by some natural class of Q/A games?

We have shown that $\mathcal{P}_k^{k+3}$ is $\mathbf{\Sigma_{2k-2}P}$-complete. It is easy to see that $\mathcal{P}_k^k$ is trivial as Paul can move down one level in each round and win any game played on digraphs of depth k. The exact complexity of $\mathcal{P}_k^{k+1}$ remains elusive, except in the case of $k = 2$ (Theorem 9). One can show that $\mathcal{P}_k^{k+2}$ is $\mathbf{\Sigma_{2k-3}}$-hard and $\mathbf{\Pi_{2k-3}}$-hard. This leads to the following problem.

Problem 7.2. Characterize the complexity of $\mathcal{P}_k^{k+1}$ and $\mathcal{P}_k^{k+2}$ for $k > 2$.

A summary of known results for fixed round Q/A games is given in Table 2. To obtain these results we use similar gadgetry as given in the proof of Theorem 12. The detailed proofs are given in [16].

The PSPACE-completeness has the following interesting consequence.

Theorem 41. *There exists a LOGSPACE computable function* $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ *such that* f *maps encodings of Q/A games to Q/A games. Furthermore, Paul wins G if and only if Carole wins* $f(G)$. □

This theorem says that one can turn "answers" into "questions" and vice versa. We are not aware of any "simple" or "intuitive" transformation that will convert Paul-win games to Carole-win games. It will be interesting to look for such a transformation.

There are many intriguing questions about Q/A games when the digraph is restricted in some way. For trees, the senior author finds the following question fascinating:

Problem 7.3. Given an arbitrary tree, T, and a question vector, $\mathbf{q} = (q_1, \ldots, q_k)$, is it possible to determine in polynomial time, if Paul wins $(T, \mathbf{q})$?

The above question is also open for fixed strategies. We recall from Section 2 that there are trees and question vectors where Paul can win via a non-fixed strategy and lose if he follows any fixed strategy.

Analysis of Q/A games for many simple classes of graphs is also open. Consider an n-level graph with vertex set $\{(i, j) : 0 \leq i \leq n, \text{ and } 0 \leq j \leq i\}$. Each vertex, (i, j), is connected to $(i+1, j)$ and $(i+1, j+1)$. We call this graph, M_n, a *mesh of order n*. The following question mentioned in [1,2] is still open for all $q > 3$.

Problem 7.4. How many rounds are required by Paul to win on M_n if he inquires about q vertices in each round?

We believe that studying Q/A games on other specific graphs can be very interesting.

Lastly, Theorem 12 motivates the following question about the complexity of combinatorial games in general.

Problem 7.5. Capture **PH**, or infinitely many levels of **PH**, by restricted versions of other well-known combinatorial games.

For further reading

Fig. 1.

Acknowledgements

We would like to thank Ahsan Kamal and Yasser Hashmi for proof reading a draft of this paper. We are also thankful to an anonymous referee (of CCC) who pointed out that while reducing $\mathbf{QSAT}_{2k-2}$, we were assuming that the formula was in 3-CNF thereby only proving that k-round games are $\mathbf{\Sigma_{2k-3}}$-hard. The second author would like to thank his advisor Arif Zaman for his encouragement and guidance. We would also like to thank anonymous referees of this journal whose comments helped in improving the exposition.

References

[1] S. Abbasi, Do answers help in posing questions? Technical Report 98-35, DIMACS, 1998.
[2] S. Abbasi, Impact of parallelism on the efficiency of binary tree search, Ars Combin. (in press).
[3] S. Abbasi, N. Sheikh, Some hardness results for Q/A games, Integers 7 (2007) G08.
[4] T.H. Cormen, C. Stein, R.L. Rivest, C.E. Leiserson, Introduction to Algorithms, McGraw-Hill Higher Education, 2001.
[5] B. Durand, A.C. Fabret, On the complexity of deadlock detection in families of planer nets, Theoret. Comput. Sci. 215 (1–2) (1999) 225–237.
[6] S. Even, R.E. Tarjan, A combinatorial problem which is complete in polynomial space, J. ACM 23 (4) (1976) 710–719.
[7] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, 1979.
[8] C. Löding, P. Rohde, Solving the sabotage game is pspace-hard, in: MFCS, 2003, pp. 531–540.
[9] C.M. Papadimitriou, Computational Complexity, Addison-Wesley, Reading, Massachusetts, 1994.
[10] A. Pelc, Ulam's problem on searching with a lie, J. Combin. Theory. Ser. A. 44 (1987) 129–140.
[11] A. Pelc, Searching games with errors—Fifty years of coping with liars, Theoret. Comput. Sci. 270 (1–2) (2002) 71–109.
[12] A. Rényi, On a problem of information theory, MTA Mat. Kut. Int. Kozl. 6B (1961).
[13] A. Rényi, Napl'o az információelméletről, Gondolat, Budapest, 1976. (English Translation: A Diary on Information Theory, Wiley, New York, 1984.).
[14] M. Schaefer, C. Umans, Completeness in the polynomial-time hierarchy: A compendium, SIGACT News 33 (3) (2002) 32–49.
[15] M. Schaefer, C. Umans, Completeness in the polynomial-time hierarchy: A compendium (updated version). August 2006. available at http://ovid.cs.depaul.edu/documents/phcom.ps.
[16] N. Sheikh, Question/Answer Games, Ph.D. Thesis, Lahore University of Management Sciences (in preparation).
[17] J. Spencer, Ulam's searching game with fixed number of lies, Theoret. Comput. Sci. 95 (1992).
[18] L.J. Stockmeyer, The polynomial-time hierarchy, Theoret. Comput. Sci. 3 (1) (1976) 1–22.
[19] S. Ulam, Adventures of a Mathematician, Scribner, New York, 1977.
[20] van Emde Boas, The convenience of tilings, in: Complexity, Logic, and Recursion Theory, Marcel Dekker, Inc., 1997.