The surprizing complexity of generalized reachability games

Nathanaël Fijalkow, Florian Horn

The surprizing complexity of generalized reachability games

Nathanaël Fijalkow[1,2] and Florian Horn[1]

[1] LIAFA
CNRS & Université Denis Diderot - Paris 7, France
florian.horn@liafa.jussieu.fr
[2] ENS Cachan
École Normale Supérieure de Cachan, France
nathanael.fijalkow@liafa.jussieu.fr

Abstract. Games on graphs with ω-regular objectives provide a natural model for reactive systems. In this paper, we consider generalized reachability games: given subsets $F_1, \dots, F_k \subseteq V$ of vertices, the objective is to reach one vertex of each F_i. We show that solving generalized reachability games is PSPACE-complete in general. In the special case where reachability sets are singletons, the problem becomes polynomial. The case where reachability sets have size 2 is still open. We consider one-player restrictions and prove they are NP-complete and polynomial, depending on whether the player tries to ensure or to spoil the generalized reachability objectives. We also investigate memory requirements of both players.

1 Introduction

Graphs games. Graphs games are used to model reactive systems. The system is modelled by a finite graph: vertices represent states and edges represent transitions. Interactions with the environment are captured by games played on graphs. If in a given state, the controller can choose the evolution of the system, then the corresponding vertex is controlled by the first player, Eve. If the system evolves in an uncertain way, we consider the worst-case scenario. A second player, Adam, controls those states. A pebble is initially placed on the vertex representing the initial state of the system. Then Eve and Adam move this pebble along the edges, constructing an infinite sequence of vertices. Eve tries to ensure that the sequence built satisfies some predetermined objective. So, in order to synthesize a controller, we are interested in whether Eve can ensure this objective and what resources she needs (see [GTW02] for more details).

Reachability and generalized reachability. One of the simplest objective is reachability, which requires, given a subset of vertices F, that a vertex from F is reached. Another is the Büchi objective, which requires that a vertex from

F is visited infinitely often. Generalized Büchi objectives have been defined in [DJW97], and are conjunctions of Büchi objectives. Reachability games, as well as both Büchi and generalized Büchi games can be solved in polynomial time [DJW97]. However, conjunctions of reachability objectives have not yet been studied. Accordingly, we define generalized reachability objectives as conjunctions of reachability objective. They require that given k subsets of vertices $F_1, \ldots, F_k \subseteq V$, one vertex of each F_i is visited.

Contributions. We first prove that the problem of solving generalized reachability games is PSPACE-complete. Using the same ideas, we also show that the one-player restriction, where all vertices belong to Eve, is NP-complete (the other one-player restriction, where all vertices belong to Adam can be solved in polynomial time). However, the PSPACE-hardness holds only when the reachability sets are of size at least 3. If for all i, F_i is a singleton, then we show that the problem of solving these games is polynomial. The case where reachability sets are of size 2 is still open.

We study the memory requirements of both players, focusing on two parameters: the number k of reachability sets and their size. In the general case, we first prove that both players need at most $2^k - 1$ memory states. We strengthen this result to get an upper bound of $\binom{k}{\lfloor k/2 \rfloor}$ for Adam, and show that these bounds are tight. In the case where each reachability set has size 2, we show that Eve requires memory $2^{\lfloor k/2 \rfloor}$. Giving a lower bound on memory needed for Adam in this case remains an open problem. Finally, we give exact memory requirements for the case where each reachability set has size 1: Eve needs k memory states and Adam only 2.

Outline. Section 2 will give the definitions we need in the paper. In section 3, we study the complexity of solving generalized reachability games. We also consider one-player games and restriction over the size of the reachability sets. In section 4, we investigate the memory requirements of both players, for the general case and for the restricted case where reachability sets have size 1.

2 Definitions

An *arena* $\mathcal{A} = (V, (V_\circ, V_\Box), E)$ consists of a finite graph (V, E) and a partition $(V_\circ, V_\Box)$ of the vertex set V: a vertex belongs to Eve if it lies in $V_\circ$ and to Adam if it lies in $V_\Box$. When drawing arenas, we will use circles for vertices owned by Eve and squares for those owned by Adam. We denote by n the number of vertices and m the number of edges. A *play* π in an arena $\mathcal{A}$ is an infinite sequence $\pi_0, \pi_1, \pi_2 \ldots$ of vertices such that for all $i \geq 0$ we have $(\pi_i, \pi_{i+1}) \in E$. We denote by Π the set of all plays. We define *objectives* for a player by

giving a set of winning plays $\Phi \subseteq \Pi$. We study zero-sum games, where the objectives of the two players are opposite. Hence, if Eve has the objective Φ, then Adam has the objective $\Pi \setminus \Phi$. A *game* is a couple $\mathcal{G} = (\mathcal{A}, \Phi)$ where $\mathcal{A}$ is an arena and Φ an objective.

A *strategy* for a player is a function that prescribes, given a finite history of the play, the next move. Formally, a *strategy* for Eve is a function $\sigma : V^* \cdot V_\circ \to V$ such that for all $w \in V^*$ and $v \in V_\circ$ we have $(v, \sigma(w \cdot v)) \in E$. Strategies for Adam are defined similarly, and usually denoted by τ. Once a game $\mathcal{G} = (\mathcal{A}, \Phi)$, a starting vertex v_0 and strategies σ for Eve and τ for Adam are fixed, there is a unique play $\pi(v_0, \sigma, \tau)$, which is said winning for Eve if it belongs to Φ. Given a game $\mathcal{G} = (\mathcal{A}, \Phi)$ and a starting vertex v_0, a strategy σ for Eve is winning if for all τ strategy for Adam, $\pi(v_0, \sigma, \tau)$ is winning. Eve wins the game $\mathcal{G} = (\mathcal{A}, \Phi)$ from v_0 if she has a winning strategy from v_0. Given an arena $\mathcal{A}$ and an objective Φ, we denote by $\mathcal{W}_E(\mathcal{A}, \Phi)$ the winning positions of Eve, that is the set of vertices from where Eve wins. The problem of solving a game is to decide whether Eve wins in a given game $\mathcal{G}$ from a given vertex v_0.

A *memory structure* $\mathcal{M} = (M, m_0, \mu)$ for an arena $\mathcal{A}$ consists of a set M of memory states, an initial memory state $m_0 \in M$, and an update function $\mu : M \times V \to M$. The update function can be extended to a function $\mu^* : V^+ \to M$ by defining $\mu^*(V) = \{m_0\}$ and $\mu^*(w \cdot v) = \mu(\mu^*(w), v)$. Given a memory structure $\mathcal{M}$ and a next-move function $\nu : V_\circ \times M \to V$, we can define a strategy σ for Eve by $\sigma(w \cdot v) = \nu(v, \mu^*(w \cdot v))$. A strategy with memory structure $\mathcal{M}$ has finite memory if M is a finite set. It is *memoryless*, or *positional* if M is a singleton: it only depends on the current vertex. Hence a memoryless strategy can be described as a function $\sigma : V_\circ \to V$.

An arena $\mathcal{A} = (V, (V_\circ, V_\square), E)$ and a memory structure $\mathcal{M}$ for $\mathcal{A}$ induce the expanded arena $\mathcal{A} \times \mathcal{M} = (V \times M, (V_\circ \times M, V_\square \times M), E \times \mu)$ where $E \times \mu$ is defined by: $((v, m), (v', m')) \in E'$ if $(v, v') \in E$ and $\mu(m, v) = m'$. Given ϕ an objective on $\mathcal{A}$, we consider the corresponding objective $\phi_\mathcal{M}$ on $\mathcal{A} \times \mathcal{M}$ that do not consider the memory. There is a natural bijection between plays in $\mathcal{A}$ and in $\mathcal{A} \times \mathcal{M}$. This bijection respects objectives: if ϕ is an objective on $\mathcal{A}$, then a play in $\mathcal{G} = (\mathcal{A}, \Phi)$ is winning if and only it is winning in $\mathcal{G} \times \mathcal{M} = (\mathcal{A} \times \mathcal{M}, \Phi_\mathcal{M})$. From a memoryless strategy in $\mathcal{A} \times \mathcal{M}$, we build a strategy in $\mathcal{A}$ using $\mathcal{M}$ as memory structure, which behaves as the original strategy. The key observation is that if Eve has a winning memoryless strategy in $\mathcal{G} \times \mathcal{M}$ from (v_0, m_0), then she has a winning strategy in $\mathcal{A}$ from v_0 using $\mathcal{M}$ as memory structure.

Reachability objectives require that, given a subset F of vertices, a vertex of F is visited: $\mathrm{Reach}(F) = \{\pi_0, \pi_1, \pi_2 \ldots \mid \exists p \in \mathbb{N}, \pi_p \in F\}$. Games in the form $\mathcal{G} = (\mathcal{A}, \mathrm{Reach}(F))$ are called reachability games. To determine whether Eve wins a reachability game, we compute the reachability set attractor. We

define the sequence $(\mathrm{Attr}_i(F))_{i\geq 0}$:

$$\begin{array}{lll} \mathrm{Attr}_0(F) & = F & \\ \mathrm{Attr}_{i+1}(F) = \mathrm{Attr}_i(F) \cup & \{u \in V_\circ \mid \exists (u,v) \in E, v \in \mathrm{Attr}_i(F)\} \\ & \cup \quad \{u \in V_\Box \mid \forall (u,v) \in E, v \in \mathrm{Attr}_i(F)\} \end{array}$$

Then $\mathrm{Attr}(F)$ is the limit of the non-decreasing sequence $(\mathrm{Attr}_i(F))_{i\geq 0}$. We can prove that $\mathcal{W}_E(\mathcal{A}, \mathrm{Reach}(F))$ is exactly $\mathrm{Attr}(F)$.

Generalized reachability objectives require that given k subsets of vertices $F_1, F_2, \ldots, F_k$, a vertex of F_i is visited for each i:

$$\mathrm{GenReach}(F_1, F_2, \ldots, F_k) = \bigcap_{1\leq i\leq k} \mathrm{Reach}(F_i)$$

Games in the form $\mathcal{G} = (\mathcal{A}, \mathrm{GenReach}(F_1, F_2, \ldots, F_k))$ are called generalized reachability games. The special cases where in $\mathcal{A}$, $V_\Box$ (resp. $V_\circ$) is empty are called one-player (resp. opponent-player) generalized reachability games.

Example 1. We consider the arena drawn in Figure 1. We define a generalized reachability game by giving reachability sets $F_1 = \{2, 3\}$ (thick vertices) and $F_2 = \{4\}$ (dashed vertex). 1 is the initial vertex. Eve tries to visit 2 and 4, or 3 and 4.

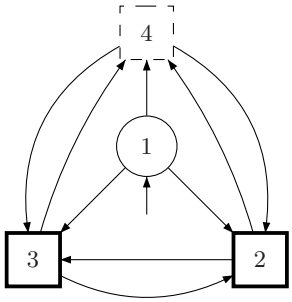

Fig. 1. An example of a generalized reachability game

3 Solving generalized reachability games

In this section we investigate the complexity of solving generalized reachability games. We first present a reduction from QBF (evaluation of a quantified boolean formulae in conjunctive normal form) to generalized reachability games. This reduction gives the PSPACE-hardness of solving these games. We then show it is actually PSPACE-complete.

We then study one-player restrictions. A special case of the previous reduction gives a reduction from SAT to one-player generalized reachability games, so solving these games is NP-hard. As above, we describe an algorithm in NP, and get the NP-completeness of solving one-player generalized reachability games. The other one-player restriction, opponent-player generalized reachability games, can be solved in polynomial time.

We consider the subclass of generalized reachability games when the size of reachability sets are restricted. We note that our PSPACE-hardness result holds only when the reachability sets are of size at least 3. Indeed, we show that the problem is polynomial if reachability sets are singletons. The case where reachability sets have size 2 is still open.

3.1 PSPACE-completeness of solving generalized reachability games

We first define a reduction from QBF to generalized reachability games. We consider a quantified boolean formula

$$Q_1 x_1 \, Q_2 x_2 \, \ldots Q_n x_n \, \phi \, ,$$

where ϕ is a propositional formula in conjunctive normal form, *i.e*

$$\phi = \bigwedge_{i \in \{1,\ldots,k\}} \ell_{i,1} \vee \ell_{i,2} \vee \ldots \vee \ell_{i,j_i}$$

and $\ell_{i,j}$ belongs to $\{x_1, \neg x_1, \ldots, x_n, \neg x_n\}$. We design a generalized reachability game. Intuitively, the two players will sequentially choose to assign values to variables, following the quantification order and starting from the outermost variable. Eve chooses existential variables and Adam chooses universal variables. Formally, the game is as follows:

- for each variable x_i, there is one vertex for each literal (x_i and $\neg x_i$);
- for each variable x_i, there is a choice vertex c_i which can lead to either x_i or $\neg x_i$. The choice vertex belongs to Eve if x_i is existentially quantified, and to Adam if x_i is universally quantified;
- for each variable x_i with $i < n$, there are two edges from x_i and $\neg x_i$ to the next choice vertex c_{i+1};
- there is a sink s, and two edges from x_n and $\neg x_n$ to s;
- for each clause $\ell_{i,1} \vee \ell_{i,2} \vee \ldots \vee \ell_{i,j_i}$, there is a reachability set $\{\ell_{i,1}, \ell_{i,2}, \ldots, \ell_{i,j_i}\}$.

The initial vertex is c_1. There is a natural bijection between assignments of the variables and plays in this game. The assignment satisfies the formula ϕ if

and only if all clauses are satisfied, which is equivalent to visit all reachability sets. Hence Eve has a winning strategy in the designed game if and only if the formula is valid.

Example 2. We consider the following QBF formula

$$\forall x \, \exists y \, \forall z \, (x \vee \neg y) \wedge (\neg y \vee z) \, .$$

Figure 2 shows the game built by the reduction. We have $F_1 = \{x, \neg y\}$ (thick vertices) and $F_2 = \{\neg y, z\}$ (dashed vertices).

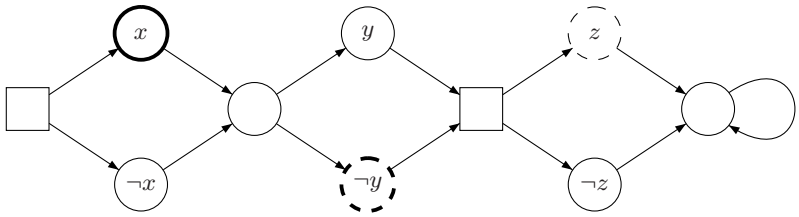

Fig. 2. An example of the reduction from QBF to generalized reachability games.

Theorem 1 (PSPACE-completeness of generalized reachability games). *Solving generalized reachability games is* PSPACE*-complete.*

Proof. We argue that if Eve has a winning strategy, then she has a winning strategy that visits each reachability set within nk steps. Indeed, if she can enforce to visit a reachability set, then she can enforce it within n steps.

We now describe an alternating Turing machine that solves generalized reachability games and works in polynomial time. Since $\mathrm{APTIME} = \mathrm{PSPACE}$, the result follows. The machine simulates the game on nk steps. Whenever a vertex belongs to Eve, the corresponding state is disjunctive, otherwise it is conjunctive. A branch of length nk is accepted if it is winning. This machine accepts if and only if Eve wins.

PSPACE-hardness follows from the previous reduction.

3.2 Solving one-player restrictions

Theorem 2 (One-player restrictions). *Solving one-player generalized reachability games is* NP*-complete. Solving opponent-player generalized reachability games is polynomial.*

Proof. We first deal with one-player generalized reachability games. In our previous reduction, consider the case where all variables in the original formula are quantified existentially. Then the problem is SAT (satisfiability of a boolean formula in conjunctive normal form), which is NP-complete. Resulting games are one-player games, as all vertices belong to Eve. Hence solving one-player generalized reachability games is NP-hard.

We now describe a non-deterministic algorithm to solve these games in polynomial time. As noted before, if Eve wins, then she has a winning strategy that visits each reachability set within nk steps. The algorithm guesses a path of length nk and checks whether it is winning. Thus solving one-player generalized reachability games is NP-complete.

We now consider opponent-player generalized reachability games. The key observation is that Adam wins if and only if he can avoid one reachability set, *i.e* if the initial vertex does not belong to each reachability set attractor. This is easily checked in polynomial time.

3.3 Restriction over the size of reachability sets

Theorem 3 (Generalized reachability games where reachability sets are singletons). *Solving generalized reachability games where reachability sets are singletons is polynomial.*

Proof. We denote by v_i the only vertex in F_i, for all i. Intuitively, under the assumption that reachability sets are singletons, we will see that if Eve wins, then she has a winning strategy that prescribes "reach $v_{f(1)}$, then $v_{f(2)}$, and so on", where f is a permutation over $\{1, \ldots, k\}$.

We denote by $(\dagger)$ the following property: there exists a permutation f over $\{1, \ldots, k\}$ such that for all $1 \leq i \leq k-1$, we have $v_{f(i)} \in \mathrm{Attr}(v_{f(i+1)})$. It implies that for all $i \leq j$, we have $v_{f(i)} \in \mathrm{Attr}(v_{f(j)})$ (the relation $v \preceq v'$ if $v \in \mathrm{Attr}(v')$ is a pre-order). We consider two cases:

- If $(\dagger)$ is satisfied, then we show that $\mathcal{W}_E$, set of winning positions for Eve, is $\cap_i \mathrm{Attr}(v_i)$. Let $v \in \cap_i \mathrm{Attr}(v_i)$ and f a permutation that satisfies $(\dagger)$, we design a winning strategy from v that visits $v_{f(1)}$, then $v_{f(2)}$, and so on. Note that this strategy only needs k memory states. Conversely, if $v \notin \cap_i \mathrm{Attr}(v_i)$, then Eve cannot win, as Adam can prevent her from reaching one reachability set.
- If $(\dagger)$ is not satisfied, then there exists v_i and v_j such that $v_i \notin \mathrm{Attr}(v_j)$ and $v_j \notin \mathrm{Attr}(v_i)$. We describe a winning strategy for Adam: "if v_i or v_j has been reached, then avoid the other". Following this strategy, Adam ensures that during a play, v_i and v_j cannot be both reached. Note that this strategy only needs 2 memory states.

4 Memory requirements

4.1 Memory bounds for the general case

Lemma 1 (Memory exponential upper bound for both players). *If Eve (resp. Adam) wins a generalized reachability game, then she (resp. he) wins using a strategy with memory* $2^k - 1$ *(resp.* $\binom{k}{\lfloor k/2 \rfloor}$*).*

Proof. Roughly speaking, the only information needed during a play is the subset of visited reachability sets. To prove this, we expand the arena by labelling vertices with this information. Let $\mathcal{G} = (G, \mathrm{GenReach}(F_1, \ldots, F_k))$ be a generalized reachability game. We consider the memory structure $\mathcal{M} = (2^{\{1,\ldots,k\}}, \emptyset, \mu)$, where $\mu(S, v) = S \cup \{i \mid v \in F_i\}$. Let $F = \{(v, S) \mid S = \{1, \ldots, k\}\}$. A play for the generalized reachability game $\mathcal{G}$ from v_0 is winning if and only if it is winning for the reachability game $\mathcal{G} \times \mathcal{M} = (G \times \mathcal{M}, \mathrm{Reach}(F))$ from (v_0, m_0). Since in the reachability game $\mathcal{G} \times \mathcal{M}$, the winning player has a memoryless winning strategy, he has in $\mathcal{G}$ a winning strategy with $\mathcal{M}$ as memory structure.

Now, $\mathcal{M}$ has 2^k memory states. In order to get the correct bounds for each players, we rely on the following observations:

- Eve does not need a specific memory state to remember that all the sets have been reached, as in this case, she has already won. Thus, she can always win with $2^k - 1$ memory states.
- In the game $\mathcal{G} \times \mathcal{M}$, Adam does not need to distinguish two vertices of the form (v, S) and (v, S') where S is a subset of S': if he can win from (v, S'), then he can win from (v, S) *with the same strategy*; if he cannot win from (v, S'), he might as well always play as if he was in (v, S). This reduces the number of different memory states for a single vertex to $\binom{k}{\lfloor k/2 \rfloor}$ —the maximal number of incomparable subsets of $\{1, \ldots, k\}$. It is thus possible to derive from it a winning strategy for Adam with $\binom{k}{\lfloor k/2 \rfloor}$ memory states. Note, however, that there is no longer a natural interpretation of memory states as subsets of $\{1, \ldots, k\}$, since the merging process can lead to different results for each vertex.

Theorem 4 (Memory lower bounds for both players). *There exists an arena where Eve needs* $2^k - 1$ *memory states to win, and one where Adam needs* $\binom{k}{\lfloor k/2 \rfloor}$ *memory states to win.*

Proof. An arena where Eve needs $2^k - 1$ memory states to win is presented in [CHH10] for the more general case of Request-response games. It can be easily adapted to generalized reachability games to give the same lower bound.

We now describe a generalized reachability game won by Adam, where he needs $\binom{k}{\lfloor k/2 \rfloor}$ bits of memory to win. Let $k = 2p+1$. First Eve chooses and visits p reachability sets, then Adam chooses and visits p reachability sets. Finally Eve chooses and visits p reachability sets. In order to win, Adam must visit exactly the same reachability sets Eve visited. Otherwise at least $p + 1$ sets have been visited when Eve plays for the second time, and she can choose and visit the remaining sets that have not yet been visited.

4.2 Memory lower bound for restricted cases

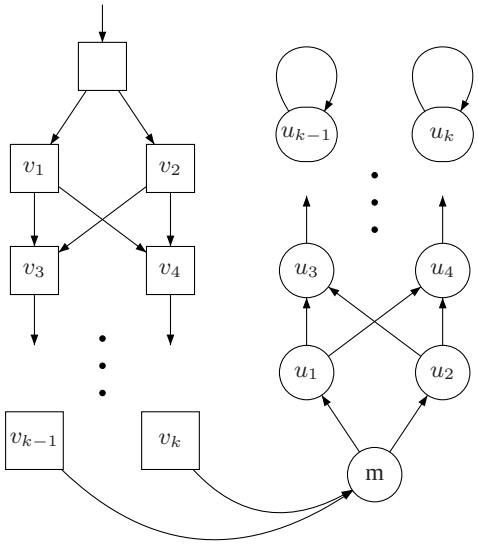

Fig. 3. Memory lower bound for Eve when reachability have size 2

Exponential memory lower bound for Eve when reachability sets have size 2. Figure 3 shows a generalized reachability game where reachability sets have size 2 won by Eve, where she needs $2^{\lfloor k/2 \rfloor}$ bits of memory to win. Let $F_i = \{u_i, v_i\}$ for all i. Adam will choose to visit either v_1 or v_2, then v_3 or v_4, and so on. When the vertex m is reached, Eve needs to remember the $\lfloor k/2 \rfloor$ choices made by Adam: if Adam visited v_1, then he did not visit v_2, so Eve has to visit u_2. Remembering those choices requires $2^{\lfloor k/2 \rfloor}$ states of memory.

Memory lower bounds. Memory lower bounds are summarized in the following array, where k is the number of reachability sets:

Size of reachability sets	1	2	any
Eve	k	$2^{\lfloor k/2 \rfloor}$	$2^k - 1$
Adam	2	?	$\binom{k}{\lfloor k/2 \rfloor}$

Memory upper bounds.

Size of reachability sets	1	2	any
Eve	k	$2^k - 1$	$2^k - 1$
Adam	2	$2^k - 1$	$\binom{k}{\lfloor k/2 \rfloor}$

References

CHH10. Krishnendu Chatterjee, Thomas A. Henzinger, and Florian Horn. The Complexity of Request-response Games. Technical report, Laboratoire d'Informatique Algorithmique: Fondements et Applications, CNRS UMR 7089, 2010.

DJW97. Stefan Dziembowski, Marcin Jurdziński, and Igor Walukiewicz. How Much Memory is Needed to Win Infinite Games? In *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science, LICS'97*, pages 99–110. IEEE Computer Society, 1997.

GTW02. Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *LNCS*. Springer-Verlag, 2002.

Hor08. Florian Horn. *Random Games*. PhD thesis, Université Paris 7 and RWTH Aachen, 2008.

HTW08. Florian Horn, Wolfgang Thomas, and Nico Wallmeier. Optimal Strategy Synthesis in Request-Response Games. In *Proceedings of the 6th International Symposium on Automated Technology for Verification and Analysis, ATVA'08*, volume 5311 of *LNCS*, pages 361–373. Springer-Verlag, 2008.

Tho02. Wolfgang Thomas. Infinite games and verification (extended abstract of a tutorial). In *International Conference on Computer Aided Verification, CAV'*, pages 58–64, 2002.