# Approximating the Value of a Concurrent Reachability Game in the Polynomial Time Hierarchy*

Søren Kristoffer Stiil Frederiksen and Peter Bro Miltersen

Aarhus University

**Abstract.** We show that the value of a finite-state concurrent reachability game can be approximated to arbitrary precision in TFNP[NP], that is, in the polynomial time hierarchy. Previously, no better bound than PSPACE was known for this problem. The proof is based on formulating a variant of the state reduction algorithm for Markov chains using arbitrary precision floating point arithmetic and giving a rigorous error analysis of the algorithm.

## 1 Introduction

A *concurrent reachability game* (e.g., [3,1,8,7]) $G$ is a finitely presented two-player game of potentially infinite duration, played between Player 1, the *reachability* player, and Player 2, the *safety* player. The arena of the game consists of a finite set of positions $0, 1, 2, \ldots, N$. When play begins, a pebble rests at position 1, the "start position". At each stage of play, with the pebble resting at a particular "current" position $k$, Player 1 chooses an *action* $i \in \{1, 2, \ldots, m\}$ while Player 2 concurrently, and without knowledge of the choice of Player 1 similarly chooses an action $j \in \{1, 2, \ldots, m\}$. A fixed and commonly known *transition function* $\pi : \{1, 2, \ldots, N\} \times \{1, 2, \ldots, m\}^2 \to \{0, 1, 2, \ldots, N\}$ determines the next position of the pebble, namely $\pi(k, i, j)$. If the pebble ever reaches 0 (the"goal position"), play ends, and Player 1 wins the game. If the pebble *never* reaches goal, Player 2 wins.

A *stationary strategy* for a player is a family of probability distribution on his actions, one for each state of the game. Everett [5] showed that every concurrent reachability game has a *value* which is a real number $v \in [0, 1]$ with the following properties [5,12,9]:

– For every $\epsilon > 0$, the reachability player has a stationary strategy for playing the game that guarantees that the pebble *eventually* reaches position 0 with probability at least $v - \epsilon$, no matter what the safety player does; such a strategy is called $\epsilon$-optimal.

* The authors acknowledge support from The Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation and from the Center for research in the Foundations of Electronic Markets (CFEM), supported by the Danish Strategic Research Council.

L. Cai, S.-W. Cheng, and T.-W. Lam (Eds.): ISAAC2013, LNCS 8283, pp. 457–467, 2013.

– The safety player has a stationary strategy for playing the game that guarantees that the pebble will never reach goal with probability at least $1-v$, no matter what the reachability player does; such a strategy is called optimal.

The present paper concerns the computation of $v$, given an explicit representation of the game (by its transition function). More precisely, as $v$ can be an irrational number, we consider finding an approximation to $v$ within an additive error of $\epsilon > 0$, when the transition function and $\epsilon$ (in standard fixed point binary representation, e.g., 0.00001) are given as input. This problem has an interesting history: Chatterjee *et al.* [2] claimed that the problem is in $\mathsf{NP}\cap\mathsf{coNP}$. Their suggested nondeterministic algorithm supposedly establishing this result was based on guessing stationary strategies for the two players. Etessami and Yannakakis [4] pointed out that the correctness proof of the algorithm of Chatterjee *et al.* is not correct, and that the best known upper bound on the complexity of the problem remained to be PSPACE, a bound that follows from a reduction to the decision problem for first order theory of the real numbers. The crucial flaw in the argument of Chatterjee *et al.* was its failure to establish correctly that the length of the standard fixed point bit representation of the numbers associated with the stationary strategies to be guessed is polynomially bounded in the size of the input. Hansen, Koucky and Miltersen [8] subsequently established that some games actually require strategies whose standard fixed point bit representations have superpolynomial size. That is, not only was the correctness proof of Chatterjee *et al.* incorrect, but so was the algorithm itself.

The main result of the present paper is the first "complexity class upper bound" better than PSPACE on the computational complexity of the problem of approximating the value of a concurrent reachability game. More specifically, consider the search problem APPROX-CRG-VALUE which on input $\langle G, 1^k\rangle$ finds an approximation to the value of the finite state concurrent reachability game $G$ within additive error $2^{-k}$. Then, our main theorem is the following.

**Theorem 1.** *APPROX-CRG-VALUE can be solved in* $\mathsf{TFNP}[\mathsf{NP}]$

The class $\mathsf{TFNP}[\mathsf{NP}]$ ("total functions from NP with an oracle for NP") was defined by Megiddo and Papadimitriou [10]. A total search problem can be solved in this class if there is a nondeterministic Turing machine $M$ with an oracle for an NP language, so that $M$ runs in polynomial time and on all computation paths either outputs *fail* or a correct solution to the input (in this case, a value approximation), and on at least one computation path does the latter. For readers unfamiliar with (multi-valued) search problem classes, we point out that by a standard argument, the fact that APPROX-CRG-VALUE is in $\mathsf{TFNP}[\mathsf{NP}]$ implies that there is a *language* $L$ in $\Delta_3^p = \mathsf{P}[\mathsf{NP}[\mathsf{NP}]]$ encoding a (single-valued) function $f$, so that $f(G, 1^k)$ approximates the value of $G$ within additive error $2^{-k}$.

Interestingly, the main key to establishing our result is to work with *floating point* rather with *fixed point* representation of the real numbers involved in the computation. We are not aware of any previous case where this distinction has been important for establishing membership in a complexity class. Nevertheless, it is natural that this distinction turns out to be important in the context of concurrent reachability games as good strategies in those are known to involve real numbers of very different magnitude (such as $2^{-1}$ and $2^{-10000000}$), by the examples of Hansen, Koucky and Miltersen. As the main technical tool, we adapt the *state reduction algorithm* for analyzing Markov chains due to Sheskin [13] and Grassman *et al.* [6]. This algorithm was shown to have very good numerical stability by O'Cinneide [11] in contrast to the standard ways of analyzing Markov chains using matrix inversion. Our adapted finite precision algorithm computes absorption probabilites rather than steady state probabilities and the numerical stability argument of O'Cinneide is adapted so that a formal statement concerning polytime Turing machine computations on arbitrary precision floating point numbers, with numbers of widely different orders of magintude appearing in a single computation, is obtained (Theorem 4 below). We emphasize that the adaptation is standard in the context of numerical analysis – in particular, the error analysis is an instance of the backward error analysis paradigm due to Wilkinson [15] – but to the best of our knowledge, the bridge to formal models of computation and complexity classes was not previously built.

## 2 Preliminaries

### Relative Distance and Closeness

For non-negative real numbers $x, \tilde{x}$, we define the *relative distance between* $\tilde{x}$ *and* $x$ to be $\delta(x, \tilde{x}) = \frac{\max(x,\tilde{x})}{\min(x,\tilde{x})} - 1$ with the convention that $0/0 = 1$ and $c/0 = +\infty$ for $c > 0$. We shall say that a non-negative real $x$ is $(u, j)$*-close* to a non-negative real $\tilde{x}$ where $u$ and $j$ are non-negative integers if $\delta(x, \tilde{x}) \le (\frac{1}{1-2^{-u+1}})^j - 1$. We omit the proofs of the following straightforward lemmas.

**Lemma 1.** *If $x$ is $(u, i)$-close to $y$ and $y$ is $(u, j)$-close to $z$, then $x$ is $(u, i+j)$-close to $z$.*

**Lemma 2.** *Let $x, \tilde{x}, y, \tilde{y}$ be non-negative real numbers so that $x$ is $(u, i)$-close to $\tilde{x}$ and $y$ is $(u, j)$-close to $\tilde{y}$. Then, $x + y$ is $(u, \max(i, j))$-close to $\tilde{x} + \tilde{y}$, $xy$ is $(u, i+j)$-close to $\tilde{x}\tilde{y}$ and $x/y$ is $(u, i+j)$-close to $\tilde{x}/\tilde{y}$.*

### Floating Point Numbers

Let $\mathbb{D}(u)$ denote the set of non-negative dyadic rationals with a $u$-bit mantissa, i.e.

$$\mathbb{D}(u) = \{0\} \cup \{x2^{-i} | x \in \{2^{u-1}, 2^{u-1}+1, \ldots, 2^u - 1\}, i \in \mathbf{Z}\}$$

The *$u$-bit floating point representation* of an element $x2^{-i} \in \mathbb{D}(u)$ is $\langle 1^u, b(x), b(i) \rangle$, where $b$ denotes the map taking an integer to its binary representation. Note that the representation is unique (for fixed $u$). The *exponent* of $x2^{-i} \in \mathbb{D}(u)$ is the number $-i$; note that this is well-defined. The $u$-bit floating point representation of 0 is $\langle 1^u, 0 \rangle$. For convenience of expression, we shall blur the distinction between an element of $\mathbb{D}(u)$ and its floating point representation. Let $\oplus^u, \oslash^u, \otimes^u$ denote the finite precision analogoues of the arithmetic operations $+, /, *$. All these operations map $\mathbb{D}(u)^2$ to $\mathbb{D}(u)$ and are defined by truncating (rounding down) the result of the corresponding exact arithmetic operation to $u$ digits.

The following lemma is straightforward.

**Lemma 3.** *Let $x$ be a non-negative real number and let $u$ be a positive integer. There is a number $\tilde{x} \in \mathbb{D}(u)$, which is $(u, 1)$-close to $x$.*

**Lemma 4.** *Let $\tilde{x}, \tilde{y} \in \mathbb{D}(u)$ with $\tilde{x}$ being $(u, i)$-close to a non-negative real number $x$ and $\tilde{y}$ being $(u, j)$-close to a non-negative real number $y$. Then $\tilde{x} \oplus^u \tilde{y}$ is $(u, \max(i, j) + 1)$-close to $x + y$, $\tilde{x} \otimes^u \tilde{y}$ is $(u, i + j + 1)$-close to $xy$ and $\tilde{x} \oslash^u \tilde{y}$ is $(u, i + j + 1)$-close to $x/y$.*

*Proof.* The statement follows from combining Lemma 2 and Lemma 1 and noting that we have that $(a + b)(1 - 2^{-u+1}) \leq a \oplus^u b \leq a + b$, and similarly for the other operations.

For technical reasons, we want to be able to represent probability distributions in floating point representation, i.e., as finite strings, in such a way that the semantics of each string is some exact, well-defined, actual probability distribution that we can refer to. Simply representing each probability in floating point will not work for us, as we would not be able to ensure the numbers summing up to exactly one. Therefore we adopt the following definition: We let $\mathbb{P}(u)$ denote the set of finite probability density functions $(p_1, p_2, \ldots, p_k)$ for some finite $k$ with the property that there exists numbers $p'_1, \ldots, p'_k \in \mathbb{D}(u)$, so that $p_i = p'_i / \sum_j p'_j$ for $i = 1, .., k$ and so that $\sum_j p'_j$ is $(u, k)$-close to 1. We also refer to the vector $(p'_1, p'_2, \ldots, p'_k)$ as an *approximately normalized representation* of $(p_1, p_2, \ldots, p_k)$. As an example, $(\frac{1}{1+2^{-100}}, \frac{2^{-100}}{1+2^{-100}})$ is in $\mathbb{P}(64)$, and has an approximately normalized 64-bit floating point representation being the concatenation of the 64-bit floating point representations of the numbers 1 and $2^{-100}$ . On the other hand, $(1 - 2^{-100}, 2^{-100})$ is not in $\mathbb{P}(u)$ for any $u \leq 2^{100}$. The following lemma simply expresses that one can generate an approximately normalized floating point approximation of any unnormalized distribution by normalizing it numerically.

**Lemma 5.** *Let $a_1, a_2, \ldots, a_k \in \mathbb{D}(u)$ and let $\tilde{p}_i = a_i \oslash^u \left(\bigoplus^u_{j=1\ldots k} a_j\right)$. Also, let $p_i = \tilde{p}_i / \sum_{j=1..k} \tilde{p}_j$. Then $(p_1, \ldots, p_k) \in \mathbb{P}(u)$ and $(\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_k)$ is an approximately normalized representation of this distribution. Also, $p_i$ is $(u, 2k)$-close to $a_i / \sum_j a_j$.*

*Proof.* By repeated use of Lemma 4, for each $i$, $a_i / \sum_j a_j$ is $(u, k)$-close to $\tilde{p}_i$. Therefore, by Lemma 2, we have that 1 is $(u, k)$-close to $\sum_i \tilde{p}_i$. Therefore, $(\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_k)$ satisfies the condition for being an approximately normalized representation. Also $p_i = \tilde{p}_i / \sum_j \tilde{p}_j$ is $(u, k)$-close to $\tilde{p}_i = \tilde{p}_i / 1$ by Lemma 2. Then, by Lemma 1, $p_i$ is $(u, 2k)$ close to $a_i / \sum_j a_j$.

The following lemma expresses that every probability density function is well-approximated by an element of $\mathbb{P}(u)$.

**Lemma 6.** *Let $q = (q_1, q_2, \ldots, q_k)$ be a probability density function. There exists $p = (p_1, p_2, \ldots, p_k)$ in $\mathbb{P}(u)$ so that for all $i$, $p_i$ and $q_i$ are $(u, 2k + 2)$-close.*

*Proof.* For $i = 1, \ldots, k$, let $a_i$ be a number in $\mathbb{D}(u)$ which is $(u, 1)$-close to $q_i$, as guaranteed by Lemma 3. By Lemma 2, $\sum_i a_i$ is $(u, 1)$-close to $\sum_i q_i = 1$. Therefore, $a_i / \sum_j a_j$ is $(u, 2)$-close to $q_i$, again by Lemma 2. Now let $p_i$ be the distribution in $\mathbb{P}(u)$ defined by applying Lemma 5 to $(a_i)$. The statement of the lemma gives us that $p_i$ is $(u, 2k)$-close to $a_i / \sum_j a_j$ which is $(u, 2)$-close to $q_i$, so by Lemma 1, $p_i$ is $(u, 2k + 2)$-close to $q_i$.

**Absorbing Markov Chains and Concurrent Reachability Games**

An *absorbing Markov chain* is given by a finite set of transient states $\{1, \ldots, N\}$ and a finite set of absorbing states $\{N+1, \ldots, N+S\}$ and transition probabilities $p_{ij}, i \in \{1, \ldots, N\}, j \in \{1, \ldots, N + S\}$ with the property that for each transient state $k_0$, there are states $k_1, k_2, \ldots, k_l$ so that $p_{k_i k_{i+1}} > 0$ for all $i$ and so that $k_l$ is absorbing. We say that the chain is *loop-free* if $p_{ii} = 0$ for all $i \in \{1, \ldots, N\}$. Given an absorbing Markov chain, the *absorption probability* $a_{ij}$ where $i$ is transient and $j$ is absorbing, is the probability that the chain is eventually absorbed in state $j$, given that it is started in state $i$.

We shall use the following theorem of Solan [14, Theorem 6] stating that the absorption probabilities of a Markov chain only change slightly when transition probabilities are perturbed (Solan's theorem is actually much more general; the statement below is its specialization to absorbing Markov chains).

**Theorem 2.** *Let $M$ and $\tilde{M}$ be absorbing Markov chains with identical sets of transient states $\{1, 2, \ldots, N\}$ and absorbing states $\{N + 1, \ldots, N + S\}$ and transition probabilities $p_{kl}$, $\tilde{p}_{kl}$ respectively. Assume that for all $k, l \in \{1, \ldots, N\}$ we have $\delta(p_{kl}, \tilde{p}_{kl}) \leq \epsilon$. Let $a_{kl}, \tilde{a}_{kl}$ denote the absorption probabilities in the two chains. Then, for each $k \in \{1, \ldots, N\}$ and each $l \in \{N+1, \ldots, N+S\}$, we have $|a_{kl} - \tilde{a}_{kl}| \leq 4N\epsilon$.*

The formalities concerning concurrent reachability games were given in the introduction. We shall use the following theorem of Hansen, Koucky and Miltersen [8, Theorem 4].

**Theorem 3.** *For any concurrent reachability games with a total number of $A \geq 10$ actions in the entire game (collecting actions in all positions belonging to both players), and any $0 < \varepsilon < \frac{1}{2}$, Player 1 has an $\epsilon$-optimal stationary strategy with all non-zero probabilities involved being at least $\varepsilon^{2^{30A}}$.*

## 3 The State Reduction Algorithm

In this section, we present an adaptation of the state reduction algorithm of Sheskin [13] and Grassman *et al.* [6] for computing steady-state probabilities in Markov chains. The algorithm is (straightforwardly) adapted to compute absorption probabilities instead of steady-state probabilities. Also, we adapt an analysis due to O'Cinneide [11] for the finite precision version of the algorithm. The adaptation is presented as a "theory of computation" flavored statement as Theorem 4.

**Lemma 7.** *There is a polynomial time algorithm MAKE-LOOP-FREE that*

- *takes as input the transition probability matrix of an absorbing Markov chain $M$ with $N$ transient states $\{1, 2, .., N\}$ and $S$ absorbing states $\{N+1, .., N+S\}$, with each transition probability distribution of $M$ being in $\mathbb{P}(u)$ and being given by an approximately normalized representation using $u$-bit floating point numbers, for an arbitrary $u \geq 1000(N+S)^2$,*
- *outputs the transition probability matrix of an absorbing loop-free Markov chain $M'$ with $N$ transient states $\{1, .., N\}$ and $S$ absorbing states $\{N+1, .., N+S\}$ and with each transition probability distribution of $M'$ being in $\mathbb{P}(u)$ and being represented by an approximately normalized representation using $u$-bit floating point numbers,*
- *with the smallest (negative) exponent among all floating point numbers in the output being at most one smaller than the smallest (negative) exponent among all floating point numbers in the input,*
- *and with the property that each absorption probability $a_{ij}$ of $M$ (for $i = 2 \ldots N, j = N+1 \ldots N+S$) differs from the corresponding absorption probability $a'_{ij}$ of $M'$ by at most $20(N+S)^3 2^{-u}$.*

*Proof.* We assume $N \geq 1$ and $S \geq 2$, otherwise the problem is trivial. Let the transition probabilities of the chain $M$ be denoted $p_{ij}$. From each outcome of $M$ seen as a sequence of states, consider removing all repeated occurrences of states (e.g., "2 2 6 6 6 5 5 .." becomes "2 6 5.."). This induces a probability distribution on sequences which is easily seen to be the distribution generated by a loop-free Markov chain $\bar{M}$ with transition probabilities $q_{ij}$ for $i = 1, \ldots, N, j = 1, \ldots S+N, i \neq j$, with $q_{ij} = p_{ij}/q_i$ where $q_i = \sum_{k \neq i} p_{ik}$. Clearly, $\bar{M}$ has the same absorption probabilities as $M$. The algorithm MAKE-LOOP-FREE constructs an approximation $M'$ to $\bar{M}$ as indicated by the pseudocode.

By Lemma 5, the output is a family of approximately normalized floating point representations of probability distributions $q'_{ij} = \tilde{q}_{ij} / \sum_k \tilde{q}_{ik}$. Let $M'$ be the Markov chain $M'$ with transition probabilities $q'_{ij}$. We need to show that $M'$ has absorption probabilities close to the absorption probabilties of the chain $M$. For this, we need to bound the relative distance between $q'_{ij}$ and $q_{ij}$. For this, note that:

- (i) Each $\tilde{p}_{ij}$ is $(u, N+S)$-close to $p_{ij}$ by definition of approximately normalized representation.
- (ii) Each $\tilde{q}_i$ is $(u, 2N+2S-2)$-close to $q_i$ by (i) and $N+S-2$ applications of Lemma 4.
- (iii) Each $\tilde{q}_{ij}$ is $(u, 2(N+S)(N+S-1))$-close to $q_{ij}$ by (ii) and Lemma 4.
- (iv) Each $q'_{ij}$ is $(u, N+S-1)$-close to $\tilde{q}_{ij}$ by Lemma 5 and the definition of approximately normalized representation.
- (v) Each $q'_{ij}$ is $(u, 2(N+S)(N+S-1)+N+S-1)$-close to $q_{ij}$ by (iii), (iv) and Lemma 1, i.e. at least $(u, 2(N+S)^2)$-close.

Theorem 2 now implies that the absorption probabilities of $M'$ differ from the corresponding absorption probabilities of $M$ by at most $4N\epsilon$ where $\epsilon =$

**Algorithm 1.** MAKE-LOOP-FREE

**Input:** $(\tilde{p}_{ij})_{i\in\{1,\ldots,N\},j\in\{1,\ldots,N+S\}}$, where for each $i = 1,\ldots,N$, $(\tilde{p}_{ij})_{j\in\{1,\ldots,N\}}$ is an approximately normalized $u$-bit floating point representation of a probability distribution $p_i$.
**for** $i = 1 \to N$ **do**
  $\tilde{q}_i \leftarrow \bigoplus^u_{l\in\{1,\ldots,N+S\}\setminus\{i\}} \tilde{p}_{il}$
  **for** $j = 1 \to N + S$ **do**
    **if** $i = j$ **then**
      $\tilde{q}_{ij} \leftarrow 0$
    **else**
      $\tilde{q}_{ij} \leftarrow \tilde{p}_{ij} \oslash^u \tilde{q}_i$
    **end if**
  **end for**
**end for**
**return** $(\tilde{q}_{ij})_{i\in\{1,\ldots,N\},j\in\{1,\ldots,N+S\}}$

$(\frac{1}{1-2^{-u+1}})^{2(N+2)^2} - 1$. Since $u \geq 1000(N + S)^2$, we have $(\frac{1}{1-2^{-u+1}})^{2(N+S)^2} \leq 1 + 5(N + S)^2 2^{-u}$, so $\epsilon \leq 5(N + S)^2 2^{-u}$, and $4N\epsilon \leq 20(N + S)^3 2^{-u}$. Finally, to show that the exponents in the output are at most one smaller than the exponents in the input, note that we actually have that $q_{ij} \geq p_{ij}$. Since $\tilde{q}_{ij}$ closely approximates $q_{ij}$ and $\tilde{p}_{ij}$ closely approximates $p_{ij}$, it is not possible for $\tilde{q}_{ij}$ to be smaller than $\tilde{q}_{ij}$ by a factor of more than two, from which the claim follows.

**Lemma 8.** *There is a polynomial time algorithm APPROX-STATE-RED that*

– *takes as input the transition probability matrix of an absorbing loop-free Markov chain $M$ with $N$ transient states $\{1, 2, .., N\}$ and $S$ absorbing states $\{N+1, .., N+S\}$, with each transition probability distribution of $M$ being in $\mathbb{P}(u)$ and being given by an approximately normalized representation using $u$-bit floating point numbers, for an arbitrary $u \geq 1000(N + S + 1)^2$,*
– *outputs the transition probability matrix of an absorbing loop-free Markov chain $M'$ with $N-1$ transient states $\{2, .., N\}$ and $S$ absorbing states $\{N+1, .., N+S\}$ and with each transition probability distribution of $M'$ being in $\mathbb{P}(u)$ and being represented by an approximately normalized representation using $u$-bit floating point numbers,*
– *with the smallest (negative) exponent among all floating point numbers in the output being at most one smaller than the smallest (negative) exponent among all floating point numbers in the input,*
– *and with the property that each absorption probability $a_{ij}$ of $M$ (for $i = 2\ldots N, j = N+1\ldots N+S$) differs from the corresponding absorption probability $a'_{ij}$ of $M'$ by at most $80(N + S)^3 2^{-u}$.*

*Proof.* Let the transition probabilities of the chain $M$ be denoted $p_{ij}$. From each outcome of $M$ as a sequence of states, consider removing all occurrences of the state 1. This induces a probability distribution on sequences which is easily seen (recalling that $M$ is loop free) to be the distribution generated by a Markov chain $\tilde{M}$ with transition probabilities $q_{ij}$, for $i, j = 2, .., N$ with $q_{ij} = p_{ij} + p_{i1}p_{1j}$. Clearly, $\tilde{M}$ has the same absorption probabilities as $M$. From each outcome of $\tilde{M}$ since as a sequence of states, consider removing all repeated occurrences of states (e.g., "2 2 6 6 6 5 5 .." becomes "2 6 5.."). This induces a probability distribution on sequences which is easily seen to be the distribution generated by a loop-free Markov chain $\bar{M}$ with transition probabilities $r_{ij}$ for $i, j = 2, \ldots, N, i \neq j$, with $r_{ij} = q_{ij}/q_i$ where $q_i = \sum_{k \neq i} q_{ik}$. Clearly, $\bar{M}$ has the same absorption probabilities as $\tilde{M}$, and hence as $M$. The algorithm APPROX-STATE-RED constructs an approximation $M'$ to $\bar{M}$ as indicated by the pseudocode.

**Algorithm 2.** APPROX-STATE-RED

**Input:** $(\tilde{p}_{ij})_{i \in \{1,\ldots,N\}, j \in \{1,\ldots,N+S\}}$, where for each $i = 1, \ldots, N$, $(\tilde{p}_{ij})_{j \in \{1,\ldots,N\}}$ is an approximately normalized $u$-bit floating point representation of a probability distribution $p_i$, and $\tilde{p}_{ii} = 0$.

```
for i, j = 2 → N do
    if i = j then
        q̃_ij ← 0
    else
        q̃_ij ← p̃_ij ⊕^u (p̃_i,1 ⊗^u p̃_1,j)
    end if
end for
for i = 2 → N do
    q̃_i ← ⊕^u_{l∈{2,...,N+S}\{i}} q̃_il
    for j = 2 → N + S do
        if i = j then
            r̃_ij ← 0
        else
            r̃_ij ← q̃_ij ⊘^u q̃_i
        end if
    end for
end for
return (r̃_ij)_{i∈{1,...,N}\{1}, j∈{1,...,N+S}\{1}}
```

By Lemma 5, the output is a family of approximately normalized floating point representations of probability distributions $r'_{ij} = \tilde{r}_{ij} / \sum_k \tilde{r}_{ik}$. Let $M'$ be the Markov chain $M'$ with transition probabilities $r'_{ij}$. We need to show that $M'$ has absorption probabilities close to the absorption probabilties of the chain $M$. For this, we need to bound the relative distance between $r'_{ij}$ and $r_{ij}$. For this, note that:

- (i) Each $\tilde{p}_{ij}$ is $(u, N+S)$-close to $p_{ij}$ by definition of approximately normalized representation.
- (ii) Each $\tilde{q}_{ij}$ is $(u, 2(N+S)+2)$-close to $q_{ij}$ by (i) and two applications of Lemma 4.
- (iii) Each $\tilde{q}_i$ is $(u, 3(N+S)-1)$-close to $q_i$ by (ii) and $N+S-3$ applications of Lemma 4.

- (iv) Each $\tilde{r}_{ij}$ is $(u, 8(N+S)^2)$-close to $r_{ij}$ by (iii) and Lemma 4.
- (v) Each $r'_{ij}$ is $(u, N+S-1)$-close to $\tilde{r}_{ij}$ by Lemma 5 and the definition of approximately normalized representation.
- (vi) Each $r'_{ij}$ is $(u, 9(N+S)^2)$-close to $r_{ij}$ by (iv), (v) and Lemma 1.

Theorem 2 now implies that the absorption probabilities of $M'$ differ from the corresponding absorption probabilities of $M$ by at most $4N\epsilon$ where $\epsilon = (\frac{1}{1-2^{-u+1}})^{9(N+S)^2} - 1$. Since $u \geq 1000(N+S+1)^2$, we have $(\frac{1}{1-2^{-u+1}})^{9(N+S)^2} \leq 1+20(N+S)^2 2^{-u}$, so $\epsilon \leq 20(N+S)^2 \cdot 2^{-u}$, and $4N\epsilon \leq 80(N+S)^3 2^{-u}$. Finally, to show that the exponents in the output are at most one smaller than the exponents in the input, note that we actually have that $r_{ij} \geq p_{ij}$. Since $\tilde{r}_{ij}$ closely approximates $r_{ij}$ and $\tilde{p}_{ij}$ closely approximates $p_{ij}$, it is not possible for $\tilde{r}_{ij}$ to be smaller than $\tilde{p}_{ij}$ by a factor of more than two, from which the claim follows.

**Theorem 4.** *There is a polynomial time algorithm APPROX-ABSORPTION that takes as input the transition probability matrix of an absorbing Markov chain $M$ with $n$ states, with each transition probability distribution of $M$ being in $\mathbb{P}(u)$ and being given by an approximately normalized representation using $u$-bit floating point numbers for some $u \geq 1000n^2$, and outputs for each transient state $i$ and each absorbing state $j$, an approximation to the absorption probability $a_{ij}$ given in $u$-bit floating point notation and with additive error at most $80n^4 2^{-u}$.*

*Proof.* For each absorption probability to be estimated, relabel states so that transient states are labeled 1,2,..$N$, with the transient state of interest being $N$. Then apply MAKE-LOOP-FREE of Lemma 7 once, and then APPROX-STATE-RED of Lemma 8 $N-1$ times, eliminating all transient states but the one of interests. As the final Markov chain is loop-free and has only one transient state, its absorption probabilities are equal to its transition probabilities and are approximations with the desired accuracies to the absorption probabilities of the orginal chain by the two lemmas.

## 4 Approximating Values in the Polynomial Time Hierarchy

In this section, we prove our main result, Theorem 1. Let $L_1$ be the language of tuples $\langle M, 1^u, \alpha \rangle$, where $M$ is an absorbing Markov chain with two absorbing states *goal* and *trap* and a distinguished start state *start*, the parameter $u$ satisfies the conditions of Theorem 4, $\alpha$ is the standard fixed point binary notation of a number between 0 and 1 (which we will also call $\alpha$), and when APPROX-ABSORPTION of Theorem 4 is applied to $M$, the approximation returned for the probability of being absorbed in *goal* when the chain is started in *start* is at least $\alpha$. Then, since APPROX-ABSORPTION is a polynomial time algorithm, we have that $L_1 \in \mathsf{P}$. Given a concurrent reachability game $G$, if Player 1's strategy is fixed to $x$ and Player 2's strategy is fixed to $y$, we get a Markov chain. If we collapse all states in this Markov chain from which the goal position 0 will be reached with probability 0 into a single state -1, we get an absorbing Markov chain with two absorbing states, 0 (*goal*) and -1 (*trap*). Let this Markov chain be denoted $M(G, x, y)$. Let $L_2$ be the language of tuples $\langle G, 1^u, i, x, \alpha\rangle$, so that $G$ is a concurrent reachability game, $i$ is either 1 or 2, $x$ is a stationary strategy for Player $i$ with each involved probability distribution being represented approximately normalized using $u$-bit floating point notation, and $\alpha$ is the standard fixed point binary representation of a number between 0 and 1, so that *for all* pure strategies $y$ of Player $3 - i$, we have that if $i = 1$ then $\langle M(G, x, y), 1^u, \alpha\rangle \in L_1$ and if $i = 2$, then $\langle M(G, y, x), 1^u, \alpha\rangle \in L_1$. Then, by construction, and since a pure strategy $y$ has a bit representation bounded in size by the bit representation of the game, $L_2 \in \mathsf{coNP}$.

We are now ready to show that APPROX-CRG-VALUE can be solved in $\mathsf{TFNP[NP]}$ by presenting an appropriate Turing machine $M$. The machine $M$ uses the language $L_2$ (or its complement, a language in $\mathsf{NP}$) as its oracle and does the following on input $\langle G, 1^k\rangle$: Let $N$ be the number of non-terminal positions of $G$ and $m$ the largest number of actions for a player in any state. Let $u^* = 1000km(N+2)^3$ and let $e^* = (k+4)2^{30A}+1$, where $A$ is the maximum of 10 and the total number of actions in $G$, collecting in each state all actions of both players. The machine nondeterministically guesses an integer $j$ between 0 and $2^{k+1}$ and a strategy profile $(x, y)$ for the two players with each involved probability distribution being in $\mathbb{P}(u^*)$ and with probabilities having exponents at least $-e^*$ (note that $e^*$ has polynomially many bits in the standard binary representation, so it is possible for $M$ to do this). If $\langle G, 1^{u^*}, 1, x, (j-1)2^{-k-1}\rangle \in L_2$ and $\langle G, 1^{u^*}, 2, y, (j+1)2^{-k-1}\rangle \in L_2$ the machine outputs $j2^{-k-1}$, otherwise it outputs *fail*. We argue that $M$ does the job correctly: Suppose $M$ outputs a number $j2^{-k-1}$. In this case, $M$ has guessed a strategy $x$ for Player 1 and a strategy $y$ for Player 2, so that $\langle G, 1^{u^*}, 1, x, (j-1)2^{-k-1}\rangle \in L_2$ and $\langle G, 1^{u^*}, 2, y, (j+1)2^{-k+1}\rangle \in L_2$. Such a strategy $x$ guarantees that *goal* is reached with probability at least $(j-1)2^{-k-1}$ minus the additive error of the estimate provided by APPROX-ABSORPTION, that is, with probability larger that $(j-2)2^{-k-1}$. Similarly, the strategy $y$ guarantees that *goal* is reached with probability at most $(j+2)2^{-k-1}$. So the value of the game is in the interval $[(j-2)2^{-k-1}, (j+2)2^{-k-1}]$ and the approximation $j2^{-k-1}$ is indeed $2^{-k}$-accurate. Finally, we show that $M$ does not output *fail* on all computation paths. Consider a path where $M$ guesses $j$, where $j$ is a number so that the value $v$ of the game is in $[(2j-1)2^{-k-2}, (2j+1)2^{-k-2}]$. Let $x^*$ be an $2^{-k-4}$-optimal stationary strategy for Player 1 with all non-zero probabilities involved being bigger than $2^{-(k+4)2^{30A}}$, as guaranteed by Theorem 3. Let $x$ be the stationary strategy that in each state is given by the probability distribution from $\mathbb{P}(u^*)$ obtained by applying Lemma 6 to the distribution in each state of $x^*$. The relative distance between probabilities according to $x$ and probabilities according to $x^*$ is at most $\gamma = 1/(1-2^{-u^*})^{2m+2}$, by that lemma. The exponents involved in an approximately normalized representation of $x'$ are all at least $-e^*$, by $x'$ closely approximating $x^*$. For each pure reply $y$ of Player 2, Theorem 2 yields that the probability of the process being absorbed in *goal* in the chain $M(x', y)$ is at least $v - 2^{-k-4} - 4N\gamma \geq v - 2^{-k-3}$. When APPROX-ABSORPTION is applied to $M(x, y)$, its estimate for this probability has error much smaller than $2^{-k-3}$, that is, its estimate is larger than $v - 2^{-k-2}$. That is, $\langle G, 1^{u^*}, 1, x, (j-1)2^{-k-1}\rangle \in L_2$. A similar construction yields a $y$ so that $\langle G, 1^{u^*}, 2, y, (j+1)2^{-k-1}\rangle \in L_2$. If $M$ guesses $j, x, y$, it does not output *fail*. This completes the proof.

## References


1. Chatterjee, K., de Alfaro, L., Henzinger, T.A.: Strategy improvement for concurrent reachability games. In: Third International Conference on the Quantitative Evaluation of Systems, QEST 2006, pp. 291–300. IEEE Computer Society (2006)
2. Chatterjee, K., Majumdar, R., Jurdziński, M.: On Nash equilibria in stochastic games. In: Marcinkowski, J., Tarlecki, A. (eds.) CSL 2004. LNCS, vol. 3210, pp. 26–40. Springer, Heidelberg (2004)
3. de Alfaro, L., Henzinger, T.A., Kupferman, O.: Concurrent reachability games. Theor. Comput. Sci. 386(3), 188–217 (2007)
4. Etessami, K., Yannakakis, M.: Recursive concurrent stochastic games. Logical Methods in Computer Science 4(4) (2008)
5. Everett, H.: Recursive games. In: Kuhn, H.W., Tucker, A.W. (eds.) Contributions to the Theory of Games Vol. III, vol. 39, Annals of Mathematical Studies. Princeton University Press (1957)
6. Grassmann, W.K., Taksar, M.I., Heyman, D.P.: Regenerative analysis and steady state distributions for Markov chains. Operations Research, 1107–1116 (1985)
7. Hansen, K.A., Ibsen-Jensen, R., Miltersen, P.B.: The complexity of solving reachability games using value and strategy iteration. In: Kulikov, A., Vereshchagin, N. (eds.) CSR 2011. LNCS, vol. 6651, pp. 77–90. Springer, Heidelberg (2011)
8. Hansen, K.A., Koucky, M., Miltersen, P.B.: Winning concurrent reachability games requires doubly exponential patience. In: 24th Annual IEEE Symposium on Logic in Computer Science (LICS 2009), pp. 332–341. IEEE (2009)
9. Himmelberg, C.J., Parthasarathy, T., Raghavan, T.E.S., Vleck, F.S.V.: Existence of p-equilibrium and optimal stationary strategies in stochastic games. Proc. Amer. Math. Soc. 60, 245–251 (1976)
10. Megiddo, N., Papadimitriou, C.H.: On total functions, existence theorems and computational complexity. Theor. Comput. Sci. 81(2), 317–324 (1991)
11. O'Cinneide, C.A.: Entrywise perturbation theory and error analysis for Markov chains. Numerische Mathematik 65(1), 109–120 (1993)
12. Parthasarathy, T.: Discounted and positive stochastic games. Bull. Amer. Math. Soc. 77, 134–136 (1971)
13. Sheskin, T.J.: A Markov chain partitioning algorithm for computing steady state probabilities. Operations Research, 228–235 (1985)
14. Solan, E.: Continuity of the value of competitive Markov decision processes. Journal of Theoretical Probability 16, 831–845 (2003)
15. Wilkinson, J.H.: A priori error analysis of algebraic processes. In: Proceedings International Congress Math., pp. 629–639. Izdat. Mir, Moscow (1968)