TUM

TECHNISCHE UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK

The Diameter of Binary Multithreaded Programs:

Preliminary Results

Alexander Malkis, Yutaka Nagashima, Steffen Borgwardt und Claudia Eckert

TUM-I1361

The Diameter of Binary Multithreaded Programs: Preliminary Results

Alexander Malkis Yutaka Nagashima Steffen Borgwardt Claudia Eckert

14 August 2013

Abstract

We consider the class of finite-state multithreaded programs equipped with a shared-memory interleaving semantics and study the possibility of useful asymptotic bounds on the (finite) diameter of the transition graph of a program from this class. This diameter gives an upper bound on the length of a shortest certificate for the existence of a bug in such a program and as such is connected to the complexity of bug-finding. We here study the dependency of the diameter on the number of threads for the so-called *binary* programs, which have two local states per thread and two shared states. We prove several upper bounds on these diameters, and provide exact values for special cases. Our bounds are based on a detailed study of the possible transition relations of the threads, and the exact values are algorithmically computed by a mechanical enumeration. Our results present an initial investigation questioning whether the growing number of threads is the main reason for the high complexity of analyzing multithreaded programs – which is often taken for granted in the literature.

1 Introduction

"The most notorious problem of explicit-state model checking of parallel systems is the so-called 'state space explosion', pertaining to the fact that the size of the state space is exponential in the number of parallel [threads]" [10]. Although the state space of each real-world multithreaded program is finite, it is "frequently much too large to be exhaustively explored" [4]. Comments like these appear frequently in the literature on formal methods for concurrency. A long line of research on program analysis is devoted to different approaches to avoid the state-space explosion or mitigate its effects. (For example, see [1, 2, 5, 6] in addition to the above.)

The state-space explosion phenomenon suggests that the *model-checking problem*, which is the actual software-engineering task we are interested in, might also have exponential complexity in the number of threads. Questioning this conclusion is our point of departure. To isolate the contribution of the number of threads n, we keep the number of shared states and local states constant, vary n, and look at how large the *diameter* of an n-threaded program can become. Similar to [3], we abstract from the textual representation of analyzed programs in programming languages.

This paper presents the first steps by looking at the so-called binary programs; a *binary* program is a multithreaded program with two local states per thread and two shared states.

2 Preliminaries

2.1 General Conventions

Let $\mathbb{N}^+$ be the set of positive integers, $\mathbb{N}_0$ be the set of non-negative integers and $\mathbb{R}$ the set of real numbers. Unless otherwise stated, the universe of quantified variables is $\mathbb{N}_0$. For a simple notation, we view natural numbers as ordinals, i.e., $\forall\, m, n\colon m < n \Leftrightarrow m \in n$.
In logical statements the symbol $\Rightarrow$ means implication and $\Longleftrightarrow$ means equivalence. The underscore _ denotes an innermost existentially quantified variable; underscores at different places correspond to different variables. E.g., if φ and ψ are binary predicate symbols, then "$\varphi(\_, 1) \wedge \psi(\_, 0)$" means "$(\exists\, x\colon \varphi(x, 1)) \wedge (\exists\, x\colon \psi(x, 0))$".
The symbol *xor* means "exclusive or": *a xor b* holds if and only if exactly one of a and b holds.
We write $\lightning$ for "Contradiction!".

2.2 Program Notation

For $n \in \mathbb{N}^+$, an (n-threaded) *program* is the tuple

$$P \;=\; (\mathrm{Glob}, \mathrm{Loc}, \to_0, \dots, \to_{n-1})$$

such that Glob and Loc are sets and

$$\forall\, i < n\colon\ \to_i \subseteq (\mathrm{Glob} \times \mathrm{Loc})^2\,.$$

Elements of Glob are called *shared states*, elements of Loc *local states*, elements of Glob × Loc *thread states*, elements of $\bigcup_{i<n} \to_i$ *thread transitions*. A *program state* is an element of

$$\mathrm{State} \quad = \quad \mathrm{Glob} \times \mathrm{Loc}^n\,.$$

A transition graph induced by P is a graph $G(P) = (\mathrm{State}, \longrightarrow)$ where

$$(g,l) \longrightarrow (g',l') \quad :\Longleftrightarrow \quad (\exists\, i < n : (g,l_i) \to_i (g',l'_i) \wedge (\forall j \in n \setminus \{i\} : l_j = l'_j))\,.$$

The *length* of a path $(s_0, \dots, s_m)$ in such a transition graph is

$$\mathrm{length}((s_0, \dots, s_m)) \quad = \quad m.$$

The *distance* form a state s to a state s' is

$$d(s,s') \;=\; \min\{m \mid \exists \text{ path } (s_0, \dots, s_m) \text{ in } (\mathrm{State}, \longrightarrow)\colon\ s_0 = s \wedge s_m = s'\}\,,$$

where $\min \emptyset := \infty$. The *diameter* of a transition graph as above is defined as

$$\mathrm{diam}(\mathrm{State}, \longrightarrow) \quad = \quad \max\{d(s,s') \mid s,s' \in \mathrm{State} \wedge d(s,s') < \infty\}\,,$$

where $\max \emptyset := -1$. Finally, we define the *diameter* of a program P as the diameter of its transition graph:

$$\mathrm{diam}(P) \;=\; \mathrm{diam}(G(P))\,.$$

2.3 Conjectures

Next we formally state the conjectures of interest.

2.3.1 General Conjecture

There is no family $(P_n)_{n\geq 1}$ of n-threaded programs such that
- P_n is n-threaded ($n \geq 1$),
- $\exists\, c \in \mathbb{R}\colon (c > 1 \wedge \forall\, n \in \mathbb{N}^+\colon \mathrm{diam}(P_n) \geq c^n)$,
- Glob is finite and the same for all P_n ($n \geq 1$) and Loc is finite and the same for all P_n ($n \geq 1$).

The above conjecture was mentioned in previous work of the first author [8, Problem 14], [9, Problem 14], and [7, Problem 1.12.3].

2.3.2 Special Conjecture

The following conjecture is an instance of the general one.

There is no family $(P_n)_{n\geq 1}$ of n-threaded programs such that
- P_n is n-threaded ($n \geq 1$),
- $\exists\, c \in \mathbb{R}\colon (c > 1 \wedge \forall\, n \in \mathbb{N}^+\colon \mathrm{diam}(P_n) \geq c^n)$,
- $|\mathrm{Glob}| = |\mathrm{Loc}| = 2$ in all P_n ($n \geq 1$).

In the following, we concentrate on resolving the special conjecture.

In this case, all our programs are *binary*: without loss of generality we assume that $\mathrm{Glob} = \{\alpha, \beta\}$ and $\mathrm{Loc} = \{0,1\}$, where α and β are different, fixed constants. We write the program $(\{\alpha,\beta\}, \{0,1\}, \to_0, \dots, \to_{n-1})$ shortly as $(\to_i)_{i<n}$.

For a short notation, we sometimes omit commas and parentheses when writing states of threads and states of the whole program: we will sometimes write thread states $(g,l) \in \mathrm{Glob} \times \mathrm{Loc}$ as gl and program states $(g,(l_0,\dots,l_{n-1})) \in \mathrm{Glob} \times \mathrm{Loc}^n$ as strings $gl_0 \dots l_{n-1}$.

Given a n-threaded program, the *depth* of a state is its distance from $(\alpha, n \times \{0\})$, formally $\mathrm{depth}(s) := d((\alpha, n \times \{0\}), s)$ for $s \in \mathrm{State}$.

The maximal possible diameter for an n-threaded program is denoted by

$$\mathrm{diamax}(n) \;=\; \max\{\mathrm{diam}(P) \mid P \text{ is an } n\text{-threaded (binary) program}\}\,.$$

3 Examples

Now we show a few n-threaded programs whose diameter is $\mathrm{diamax}(n)$ for small values of n.

3.1 $n = 1$

Consider the program with exactly one thread and transitions $\alpha 0 \to_0 \alpha 1$, $\alpha 1 \to_0 \beta 1$, and $\beta 1 \to_0 \beta 0$. This program has diameter 3. Since the total number of program states is 4, no other single-threaded program can have a larger diameter. In total there are six programs with initial state $\alpha 0$ that have diameter 3.

3.2 $n = 2$

Consider the two-threaded program that consists of exactly the following thread transitions.
Thread 0: $\alpha 0 \to_0 \alpha 1$, $\alpha 1 \to_0 \beta 0$, $\beta 0 \to_0 \alpha 0$,
Thread 1: $\alpha 1 \to_1 \beta 0$, $\beta 0 \to_1 \beta 1$.
It has diameter 7, and, since the total number of states in any two-threaded program is 8, no other two-threaded program can have a larger diameter.

3.3 $n = 3$

Consider the three-threaded program that consists of exactly the following thread transitions.
Thread 0: $\alpha 1 \to_0 \alpha 0$, $\alpha 1 \to_0 \beta 1$, $\beta 0 \to_0 \beta 1$.
Thread 1: $\alpha 0 \to_1 \alpha 1$, $\alpha 1 \to_1 \beta 0$, $\beta 1 \to_1 \alpha 0$.
Thread 2: $\alpha 1 \to_2 \beta 0$, $\beta 0 \to_2 \beta 1$, $\beta 1 \to_2 \alpha 0$.
This program has diameter 13. We will see in Lemma 5 why no other three-threaded program has a higher diameter.

4 Lower Bound

Theorem 1. $\forall\, n \geq 1 \colon \mathrm{diamax}(n) \geq 3n$.

Proof sketch. Fix $n \geq 1$. We construct an n-threaded program with a state at depth $3n$. Let the transitions of thread 0 be $\alpha 0 \to_0 \alpha 1 \to_0 \beta 0 \to_0 \beta 1$. Let the only transition of thread t be $\beta 0 \to_t \alpha 1$ for each $t \in n \setminus \{0\}$. ∎

5 Upper Bound

Given a program P with diameter k and a path $\sigma = (s_0, \ldots, s_m)$ in the transition graph of P, we will say that σ *realizes* the diameter of P, if its length is $m = k$ and there is no shorter path from s_0 to s_k.

Lemma 2. *Given an n-threaded program P of diameter k and a path σ in P that realizes the diameter of P, there is a program P' whose diameter is also k and that has a path σ' that realizes the diameter of P' and starts in $(\alpha, (0, \ldots, 0))$.*

Proof sketch. Reindex the local and shared states appropriately. ∎

Definition 3. *An n-threaded program P is called* beautiful *if there is a shortest path σ from $(\alpha, (0, \ldots, 0))$ to some state such that* $\mathrm{length}(\sigma) = \mathrm{diamax}(n)$ *and all thread transitions occur on σ. A* diamax path *of an n-threaded beautiful program is a shortest path σ from $(\alpha, (0, \ldots, 0))$ to some state such that* $\mathrm{length}(\sigma) = \mathrm{diamax}(n)$ *and all thread transitions occur on σ. A program is called* very beautiful *if it is beautiful and has the minimal number of thread transitions among all beautiful programs with the same number of threads. Formally:*
An n-threaded program $P = (\to_i)_{i<n}$ is called very beautiful
$:\Leftrightarrow$ *P is beautiful* $\wedge$ $\forall$ *n-threaded beautiful program* $(\to'_i)_{i<n} \colon \sum_{i<n} |\to_i| \leq \sum_{i<n} |\to'_i|$.

Lemma 4. *In a beautiful program every thread has a transition that changes 0 to 1.*
Formally:

$$\forall\ n \in \mathbb{N}^+,\ \text{n-threaded beautiful program } (\to_i)_{i<n},\ i < n\ \exists a, b \in \{\alpha, \beta\} \colon (a, 0) \to_i (b, 1).$$

Proof sketch. If a thread in a beautiful program does not have such a thread transition, we can add it and change the thread slightly so that the new program has a strictly larger diameter. ∎

Lemma 5. $\mathrm{diamax}(3) \leq 13$

Proof sketch. Fix a very beautiful 3-threaded program. Fix a diamax path. The program has 16 states. Show that at least two states do not belong to the path by a case split on the present thread transitions. ■

Lemma 6. $\forall n \geq 1 \colon \mathrm{diamax}(n) \leq 2^{n+1} - n + 1$.

Proof sketch. Fix an n-threaded beautiful program P and some diamax path σ of P. Show that at least $n-2$ states do not lie on σ. ■

Lemma 7. $\forall\, n \in \mathbb{N}^+ \setminus \{2\} \colon \mathrm{diamax}(n) \leq 2^{n+1} - n$.

Proof sketch. Case $n = 1$ is handled in § 3; case $n = 3$ in Lemma 5. Now let $n \geq 4$. Fix an n-threaded beautiful program P and some diamax path σ of P. Show that at least $n-1$ states do not lie on σ. ■

6 Computation Results

Finally, we show the results of computer-generated diamax values.

We have constructed a program that, by a (somewhat involved) enumeration, searches for all beautiful programs with at most 5 threads and computes the diameter of each of the beautiful programs. The results of this enumeration are presented below:

Threads n	1	2	3	4	5
Known $\mathrm{diamax}(n)$	3	7	13	14	$\geq$15
Time needed for computation, s	0	0	5.44	8538.15	$\geq$15778463

The program uses Lemmas 2 and 4 to reduce the search space. The times are given for a parallel computation performed on a single machine with two Intel® Xeon® X5690 CPUs clocked at 3.46 GHz each with a total of 16 processes running in parallel.

References

[1] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008. ISBN: 978-0-262-02649-9.

[2] Edmund Melson Clarke, Orna Grumberg, and Doron A. Peled. *Model checking*. MIT Press, Dec. 1999. ISBN: 978-0-262-03270-4.

[3] Cormac Flanagan and Shaz Qadeer. "Thread-modular model checking." In: *SPIN*. LNCS 2648. Springer, 2003, pp. 213–224.

[4] Patrice Godefroid. *Partial-order methods for the verification of concurrent systems – an approach to the state-explosion problem*. Vol. 1032. Lecture Notes in Computer Science. Springer, 1996.

[5] Gerard Johan Holzmann. "The model checker SPIN." In: *IEEE transactions on software engineering*. Vol. 23. 5. May 1997, pp. 279–295.

[6] Gerard Johan Holzmann. *The SPIN model checker: primer and reference manual*. Addison-Wesley, 2003. ISBN: 0-321-22862-6. URL: `http://www.spinroot.com`.

[7] Alexander Malkis. "Cartesian Abstraction and Verification of Multithreaded Programs." PhD thesis. Albert-Ludwigs-Universität Freiburg, 2010.

[8] Alexander Malkis, Andreas Podelski, and Andrey Rybalchenko. "Thread-Modular Verification and Cartesian Abstraction." TV'06. 2006.

[9] Alexander Malkis, Andreas Podelski, and Andrey Rybalchenko. "Thread-Modular Verification Is Cartesian Abstract Interpretation." In: *ICTAC*. Ed. by Kamel Barkaoui, Ana Cavalcanti, and Antonio Cerone. Vol. 4281. Lecture Notes in Computer Science. Springer, 2006, pp. 183–197.

[10] Petr Ročkai. "Model checking software." PhD thesis. Masaryk university, Jan. 2015.