Brief Announcement: On Decidability of 2-Process Affine Models

Petr Kuznetsov

LTCI, Télécom Paris, Institut Polytechnique Paris, France
petr.kuznetsov@telecom-paris.fr

Thibault Rieutord

CEA LIST, PC 174, Gif-sur-Yvette, France
thibault.rieutord@cea.fr

Abstract

Affine models of computation, defined as subsets of iterated immediate-snapshot runs, capture a wide variety of shared-memory systems: wait-freedom, t-resilience, k-concurrency, and fair shared-memory adversaries. The question of whether a given task is solvable in a given affine model is, in general, undecidable.

In this paper, we focus on affine models defined for a system of two processes. We show that task computability of 2-process affine models is decidable and presents a complete hierarchy of five equivalence classes of 2-process affine models.

2012 ACM Subject Classification Theory of computation → Distributed computing models

Keywords and phrases Affine tasks, Decidability

Digital Object Identifier 10.4230/LIPIcs.DISC.2020.54

Related Version A full version of the paper is available at `https://arxiv.org/abs/2008.02099`.

1 Introduction

The question of whether a task is solvable in a *wait-free* manner, i.e., in the asynchronous read-write shared-memory model with no restrictions on who and when can fail, is known to be undecidable for systems with more than 3 processes [3, 5]. We can still, however, study the *relative* computability of models of computation. The framework of *affine models* was introduced to capture task computability of various restrictions of the wait-free model [4].

More precisely, an *affine task* A on $n+1$ processes can be represented as a pure n-dimensional sub-complex of a finite number of iterations of the *standard chromatic subdivision*, i.e., $A \subseteq \mathrm{Chr}^k \mathbf{s}, k \in \mathbb{N}$, where all facets of A are of dimension n. Many shared-memory models such as t-resilience [8], k-concurrency [2] or fair adversaries [7] are characterized via affine tasks. Task computability of an *affine model* of A, denoted by A^*, is defined as follows: A^* solves a task $(\mathcal{I}, \Delta, \mathcal{O})$ if and only if there is a natural integer $b \in \mathbb{N}$ and a simplicial map $\delta\colon A^b(\mathcal{I}) \to \mathcal{O}$ such that δ is carried by Δ, i.e., $\forall s \in I, \delta(A^b(\mathcal{I})) \subseteq \Delta(s)$. A natural challenge is therefore to compare relative task computability of affine models:

A^* is *stronger* than B^*, i.e., $A^* \succeq_{\mathcal{A}} B^*$, if all tasks solvable in B^* can be solved in A^*.

Hence, we can state our problem as follows:

Given two affine tasks, A and B, is the question of whether $A^* \succeq_{\mathcal{A}} B^*$ decidable?

Equivalently, we can study decidability of the question whether A^* *solves* B, i.e., whether A solves the *simplex agreement* task on B [1]. Indeed, suppose that A^* solves B, inductively, for any $b \in \mathbb{N}$, A^* solves B^b. Thus, any task solvable in B^* can be solved in A^*.

34th International Symposium on Distributed Computing (DISC 2020).
Editor: Hagit Attiya; Article No. 54; pp. 54:1–54:3
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

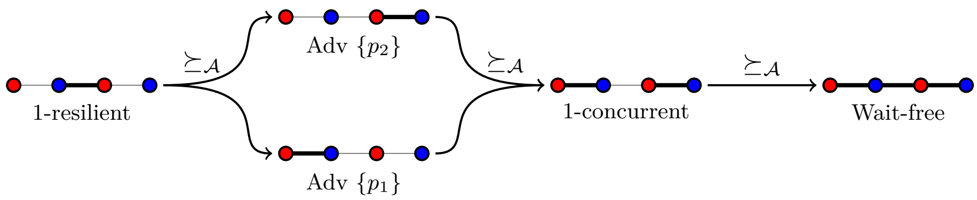

Figure 1 Relations between canonical affine tasks and corresponding models.

In this paper, we first present a framework for studying the decidability question above in 2-process affine models. We provide a complete hierarchy of 2-process affine models, including most, if not all, 2-process shared-memory models. We show that these models break down into five equivalence classes, where each class is equipped with a *representative* defined as a subset of a single iteration of the standard chromatic subdivision. The order depicted in Figure 1 defines the complete hierarchy of relative task computability of these five equivalence classes.

An intriguing question is whether this approach can be applied to higher-dimensional systems. One approach could be to focus on models defined using *link-connected* affine tasks.

2 Equivalence classes

Property selection. We define equivalence classes of 2-process affine tasks via a simple predicate on a set of properties. The power a 2-process system relies on the properties of *solo executions*, i.e., the endpoints of the corresponding affine task. Assuming a fixed input state, there is only one such endpoint v_0 (resp., v_1) of process p_0 (resp., p_1). We identify the following (obviously disjoint and forming a partition) classes of 2-process affine tasks:

1. There is a path from v_0 to v_1.
2. Both v_0 and v_1 belong to the task, but there is no path between v_0 to v_1.
3. Only v_0 belongs to the task.
4. Only v_1 belongs to the task.
5. Neither v_0 nor v_1 belongs to the task.

Equivalence for solving tasks. We show that for any tasks A and B in the same class, A^* solves B.

- If v_0 and v_1 are connected, then A and B are both iterations of the standard chromatic subdivision. By the simplicial approximation theorem, there exists a simplicial map that maps an iteration of A onto B. Thus $A^* \succeq_{\mathcal{A}} B^*$.
- Suppose now that there is no path between v_0 and v_1. Then simplices of A can be split into connected components. Let A_0 (resp., A_1) be the (possibly empty) connected component including v_0 (resp., v_1). We can then simply map every facets of A_0 (resp. A_1) to the facet of B containing v_0 (resp. v_1), and every facets of remaining connected components to any facet of B. This allows us to solve B using simply one iteration of A and, thus, $A^* \succeq_{\mathcal{A}} B^*$.

Canonical tasks. In each equivalence class, we can then select a characterizing representative, which we call a *canonical task*. We will show that the partial order on these canonical tasks (depicted in Figure 1) captures the relative power of the corresponding equivalence classes. Figure 2 depicts the five canonical affine tasks. The affine model of a canonical task is also called canonical.

(a) No v_1, no v_2 **(b)** Only v_1 **(c)** Only v_2 **(d)** v_1 and v_2 **(e)** Connected

Figure 2 Representative affine tasks for the five classes.

3 Comparing equivalence classes

To prove that the partial order in Figure 1 indeed corresponds to relative task computability power of affine models, we need to show that: (1) iterations of an affine task cannot belong to a higher equivalence class; (2) carrier-preserving simplicial maps can only send tasks to those in smaller or equal classes; and (3) canonical affine models follow this order. It is easy to check that (1) and (2) imply that equivalent affine models belong to the same class. As all models in a class are equivalent, comparing canonical tasks (3) is, hence, sufficient to compare all models. Moreover, (1) and (2) also imply that models in a class cannot solve tasks in higher classes; consequently, (3) reduces to showing that a higher canonical model is stronger than a smaller one.

- **Iterating affine tasks.** An iteration of an affine task replaces each simplex with a set of simplices defined by the task for the corresponding processes. Hence, a vertex with a carrier of dimension 0 is replaced by a vertex with a carrier of dimension 0. A path is also stable under iteration in this setting as the existence of a path is equivalent to having a complete subdivision.
- **Simplicial maps.** A carrier-preserving simplicial map must send the simplex with carrier p_0 (resp., p_1) to the simplex with carrier p_0 (resp.,p_1). Therefore, models from all classes but the smallest one can only be mapped to models in smaller classes. It is also easy to check that a path between the two endpoints must be mapped to a path between them.
- **Comparing canonical models.** If neither v_0 nor v_1 belongs to the task, we can map all facets to any other task facet. Hence, this class is the strongest one. For other canonical tasks, the order follows a direct task inclusion, which implies the solvability of canonical tasks in smaller classes (the solution being the identity map).

Formal statements and proofs of the claims listed above can be found in [6].

References

1 Elizabeth Borowsky and Eli Gafni. A simple algorithmically reasoned characterization of wait-free computation (extended abstract). In *PODC*, pages 189–198, 1997.

2 Eli Gafni, Yuan He, Petr Kuznetsov, and Thibault Rieutord. Read-write memory and k-set consensus as an affine task. In *OPODIS*, pages 6:1–6:17, 2016.

3 Eli Gafni and Elias Koutsoupias. Three-processor tasks are undecidable. *SIAM J. Comput.*, 28(3):970–983, 1999.

4 Eli Gafni, Petr Kuznetsov, and Ciprian Manolescu. A generalized asynchronous computability theorem. In *PODC*, pages 222–231, 2014.

5 Maurice Herlihy and Sergio Rajsbaum. The decidability of distributed decision tasks (extended abstract). In *STOC*, pages 589–598, 1997.

6 Petr Kuznetsov and Thibault Rieutord. On decidability of 2-process affine models, 2020. `arXiv:2008.02099`.

7 Petr Kuznetsov, Thibault Rieutord, and Yuan He. An asynchronous computability theorem for fair adversaries. In *PODC*, pages 387–396, 2018.

8 Vikram Saraph, Maurice Herlihy, and Eli Gafni. Asynchronous computability theorems for t-resilient systems. In *DISC*, pages 428–441, 2016.