Concurrency Semantics for the Geiger-Paz-Pearl Axioms of Independence

Sara Miner More, Pavel Naumov, and Benjamin Sapp

Department of Mathematics and Computer Science
McDaniel College, Westminster, Maryland 21157, USA
{smore,pnaumov,brs004}@mcdaniel.edu

Abstract

Independence between two sets of random variables is a well-known relation in probability theory. Its origins trace back to Abraham de Moivre's work in the 18th century. The propositional theory of this relation was axiomatized by Geiger, Paz, and Pearl.

Sutherland introduced a relation in information flow theory that later became known as "non-deducibility." Subsequently, the first two authors generalized this relation from a relation between two arguments to a relation between two sets of arguments and proved that it is completely described by essentially the same axioms as independence in probability theory.

This paper considers a non-interference relation between two groups of concurrent processes sharing common resources. Two such groups are called non-interfering if, when executed concurrently, the only way for them to reach deadlock is for one of the groups to deadlock internally. The paper shows that a complete axiomatization of this relation is given by the same Geiger-Paz-Pearl axioms.

1998 ACM Subject Classification F.0 Theory of Computation

Keywords and phrases Independence, concurrency, information flow, axiomatization

Digital Object Identifier 10.4230/LIPIcs.CSL.2011.443

1 Introduction

In this paper, we show that the same logical principles describe independence in three very different domains: probability, information flow, and concurrency.

1.1 Independence in Probability Theory

In probability theory, two events are called independent if the probability of their intersection is equal to the product of their probabilities. It is believed [6] that this notion was first introduced by de Moivre [2, 3]. If $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_m\}$ are two disjoint sets of random variables with finite ranges of values, then these two sets of variables are called independent if for any values $v_1, \dots, v_n$ and any values $w_1, \dots, w_m$, events $\bigwedge_{i \le n}(a_i = v_i)$ and $\bigwedge_{i \le m}(b_i = w_i)$ are independent. We denote this relation by $A \parallel B$. This definition can be generalized to independence of sets of variables with infinite ranges through the independence of appropriate σ-algebras.

A complete axiomatization of propositional properties of the independence relation between two sets of random variables was given by Geiger, Paz, and Pearl[1] [8]:

[1] The axiom names shown here are ours.

1. Empty Set: $A \parallel \varnothing$,
2. Symmetry: $A \parallel B \to B \parallel A$,
3. Monotonicity: $A \parallel B, C \to A \parallel B$,
4. Exchange: $A, B \parallel C \to (A \parallel B \to A \parallel B, C)$,

where here and everywhere below A, B means the union of sets A and B. Furthermore, Studený [14] showed that *conditional* probabilistic independence does not have a complete finite axiomatization.

1.2 Independence in Information Flow

Sutherland [15] introduced a relation between two pieces of information, which we will call "secrets", that later became known as the "nondeducibility" relation. Two secrets are in this relation if any possible value of the first secret is consistent with any possible value of the second secret. More and Naumov [13] generalized this relation to a relation $A \parallel B$ between two sets of secrets and called it independence: sets of secrets A and B are independent if each possible combination of the values of secrets in A is consistent with each possible combination of the values of secrets in B. This relation also satisfies the Empty Set, Symmetry, Monotonicity, and Exchange axioms given above.

Describing the probabilistic semantics of relation $A \parallel B$, Geiger, Paz, and Pearl [8] assumed that sets A and B are disjoint since independence of a variable from itself is not a very intuitive idea. Under More and Naumov's semantics of secrets [13], however, $A \parallel A$ means that all secrets in set A have constant values which are known to everyone. More and Naumov called such secrets "public knowledge" and considered the relation $A \parallel B$ on sets of secrets where sets A and B are not necessary disjoint. They introduced a logical system that consists of the above Empty Set, Symmetry, Monotonicity, and Exchange axioms, as well as the following additional axiom:

5. Public Knowledge: $A \parallel A \to (B \parallel C \to A, B \parallel C)$.

They proved the completeness of this system with respect to a semantics of secrets. By analyzing their completeness proof, one can easily observe that if sets A and B are assumed to be disjoint, then the original four-axiom system of Geiger, Paz, and Pearl is complete with respect to the same semantics of secrets.

Cohen [1] presented a related notion called *strong dependence*. More recently, Halpern and O'Neill [9] introduced f-secrecy to reason about multiparty protocols. In our notation, f-secrecy is a version of the nondeducibility predicate whose left or right side contains a certain function of the secret rather than the secret itself. More and Naumov also axiomatized a variation of the independence relation between secrets over graphs [11, 5] and hypergraphs [12].

1.3 Independence in Concurrency Theory

In this paper, we propose a third semantics for the Geiger-Paz-Pearl axioms of independence. Under this semantics, independence is interpreted as "non-interference" between two sets of concurrent processes. Suppose that α and β are two such processes. We say that these processes interfere if they can deadlock when executed together. That is, there is a reachable state in which neither process can make a transition to another state, but at least one of the two processes can make a transition if the other process is not present.

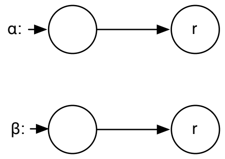

Figure 1 Two interfering processes.

One of the simplest examples of two such processes α and β is shown in Figure 1. Processes α and β both have initial states that require no resources and a second state in which the process requires a resource r.

Suppose that process α makes a transition from the initial state to the second state. Then the whole system reaches deadlock although process β still would be able to make a transition in the absence of process α.

In this paper we will study the relation $A \parallel B$ between two *sets* of processes A and B. We will say that a set of processes A interferes with a set of processes B if these two sets can reach a deadlocked state where either set A or set B is not internally deadlocked.

For example, consider a variation of Dijkstra's dining philosopher problem depicted in Figure 2. It consists of three processes α, β, and γ, representing three dining philosophers. Each philosopher has access to two out of three resources r, s, and t, representing three forks in the dining philosophers problem. Each philosopher can acquire its two resources in any order, but needs both of them in order to "eat". Once a philosopher becomes full, he leaves the table and the process terminates.[2]

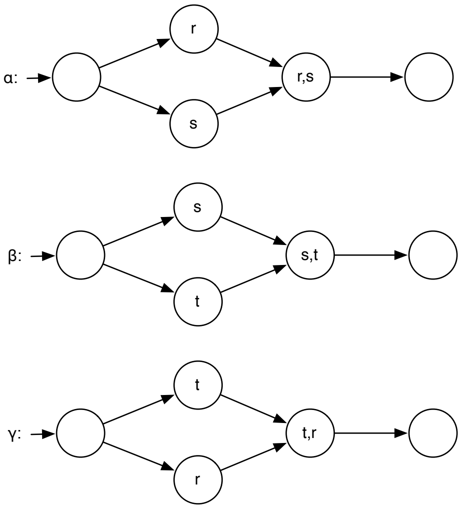

Figure 2 Three dining philosophers.

Let us first consider the concurrent execution of just two of these processes: α and β. Of course, if process α, for example, acquires resource s, then process β will need to wait until this resource is released before it will be able to finish. However, note that in any state of the composition of these two processes, at least one of the processes can make a transition, until both processes arrive at their respective final states. Thus, processes α and β do not interfere. We denote this non-interference by $\alpha \parallel \beta$.

The situation changes when all three processes are executed concurrently. If process α acquires resource r, process β acquires resource s, and process γ acquires resource t, then the system enters a deadlocked state in which none of the processes can make a transition. Yet, note that each process running alone can make a transition. In fact, any *pair* of processes running concurrently can make a transition in the absence of the third process. This means, for example, that the single process α interferes with the set of processes $\{\beta, \gamma\}$. In our notation, this can be expressed as $\neg(\alpha \parallel \beta, \gamma)$.

The main technical results of this paper are the soundness and completeness of the Geiger-Paz-Pearl logical system with respect to the non-interference semantics of concurrent processes sketched above. The significant implication of these results is that *the same non-trivial set of axioms captures the properties of independence in three very different settings: probability, information flow, and concurrency.*

2 Semantics

In order to prove formal results about process interference, we need to specify a mathematical model of concurrency. A number of models and formalisms for concurrent systems have been

[2] This is the form in which, with five philosophers rather than three, the problem was described by Hoare [10]. In Dijkstra's original version, "the life of a philosopher consists of an alternation of thinking and eating" ([4], p. 131), and, thus, graphs representing philosophers are cyclic.

developed. Among them are Petri nets, I/O automata, bigraphs, μ-calculus, and process calculi such as CCS, LOTOS, CADP, and Concurrency Workbench. (See, for example, Garavel [7], for a more recent review). Most of these were designed to be expressive enough to capture, at least potentially, reasoning about real-world systems. Since our ultimate goal is the completeness theorem, the *less expressive* our definition of concurrency, the stronger our result. Thus, instead of choosing one of the existing formalisms, we identified the minimal formalism sufficient for our proof of completeness. Specifically, we have chosen to define a process as a finite directed acyclic graph in which vertices are labeled by sets of resources. Figures 1 and 2 above depict examples of such processes. The concurrent execution of several such processes is captured in Definition 3 on page 447. We assume that concurrent processes defined using this formalism can also be captured, if needed, in other, richer, languages such as those mentioned above.

▶ **Definition 1.** A process is $\pi = (V, E, q, R, r)$, where

1. (V, E) is a finite directed acyclic graph (DAG). Vertices (elements of set V) will also be called "states" of the process.
2. $q \in V$ is a designated "initial" state of the process.
3. R is an arbitrary finite set of "resources" available to the process. Some of these resources may not actually be used by the process.
4. r is a "resource requirement" function from V to 2^R that specifies the resources used in each state. This function will be assumed to satisfy the following two conditions:
   a. $r(q) = \varnothing$,
   b. if $(v, w) \in E$, then $|r(w) \setminus r(v)| < 2$.

There are several aspects of our formalism that we would like to comment on.

Acquiring one resource at a time. Part 4 (b) of Definition 1 requires each process to acquire no more than one resource per transition. This is not a real restriction on the type of processes that we consider, but rather a restriction on how specific the description of a process should be. One always can introduce intermediate states in order to satisfy this requirement. For example, in Figure 3, instead of DAG α, one should specify DAG β, DAG γ, or DAG δ. This technical requirement is used in the proof of soundness of the Monotonicity axiom. Furthermore, in the conclusion, we will give an explicit example demonstrating that the Monotonicity axiom is false without this requirement.

Initial state. We assume that each process has a unique initial state. Additionally, we disallow processes which require resources in initial states so that each set of processes can be started concurrently. These are very technical limitations. If either condition is not satisfied, an artificial new initial state can always be added in order to satisfy it.

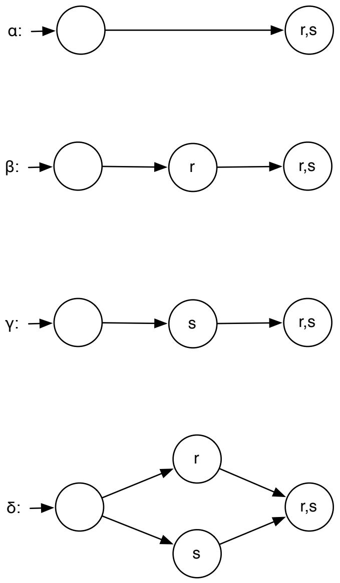

Figure 3 DAG α is not specific enough to be viewed as a process.

Resources at sink state. One might argue that, since all of our processes are finite DAGs, it is natural to assume that all processes must release all resources once they reach a sink state. We agree that this is a reasonable assumption to consider. However, our more general approach will allow us to treat the concurrent execution of any set of several processes as a single process.[3]

Acyclic graphs. By representing a process as a finite DAG, we exclude from consideration any process that can run forever or that terminates after a number of steps which was unknown a priori. Considering such processes would create a whole new set of questions about fairness, livelock, etc. that would shift focus away from the deadlock interference that we consider in this paper. This certainly could be a direction for future work.

Resource multiplicity. Although our formalism does not allow for multiple copies of the same resource, one may still capture such processes by introducing distinct copies of these resources and additional states of the process for different combinations of them, as is done in the example in Figure 6 in the conclusion.

Resource production. Some models of concurrency, such as Petri nets, assume that processes not only "acquire" resources, but also "produce" new resources or additional copies of a resource not previously available in the system. Such processes are outside of the scope of this work, because the Monotonicity axiom does not hold for them.

▶ **Definition 2.** For any process $\pi = (V, E, q, R, r)$ and any state $v \in V$, we say that π is "alive" in v if there is $w \in V$ such that $(v, w) \in E$.

In other words, π is alive in v if v is not a sink of the directed acyclic graph (V, E). If π is alive in v, we will write $Alive_v(\pi)$.

For any $\pi = (V, E, q, R, r)$, by $State(\pi)$ we mean the set of all vertices V. By $State^R(\pi)$ we mean the set of all vertices of directed graph (V, E) reachable from the process' initial state q. By $Trans(\pi)$ we mean the set of transitions E. By $Res(\pi)$ we mean the set of resources R. By a family of processes $\{\pi_i\}_{i\in I}$ we mean any *multiset* of processes. That is, we allow some of the processes in the family to be equal.

The following is a key definition of this paper that formally captures the notion of concurrent execution of a family of processes.

▶ **Definition 3.** For any family of processes $\{\pi_i\}_{i\in I}$, such that $\pi_i = (V_i, E_i, q_i, R_i, r_i)$, the product of these processes $\prod_{i\in I} \pi_i$ is a tuple $\pi = (V, E, q, R, r)$, such that

1. V is a set of all tuples $\langle v_i\rangle_{i\in I} \in \prod_{i\in I} V_i$, where $r_i(v_i) \cap r_j(v_j) = \varnothing$ for all $i, j \in I$ such that $i \neq j$,
2. E is the set of all pairs $(\langle v_i\rangle_{i\in I}, \langle w_i\rangle_{i\in I}) \in V \times V$ such that there is $i_0 \in I$ for which $(v_{i_0}, w_{i_0}) \in E_{i_0}$ and $v_i = w_i$ for each $i \neq i_0$,
3. $q = \langle q_i\rangle_{i\in I}$,
4. $R = \bigcup_{i\in I} R_i$,
5. $r(\langle v_i\rangle_{i\in I}) = \bigcup_{i\in I} r(v_i)$.

Note the similarity between this definition and the Cartesian product of finite automata. A technical difference is in the fact that we disallow the simultaneous transitions of multiple processes. However, such simultaneous transitions can always be represented by a series of single transitions executed consecutively.

If set I is empty, then, as follows from the above definition, V consists of a single element – the 0-length tuple. We will denote this tuple by $\star$. The process which is the product of an empty family of processes will be denoted by ϵ. Thus, $\star \in State(\epsilon)$. If $I = \{i_1, \ldots, i_n\}$, then

[3] Even if the original processes release all resources in sink states, the concurrent execution of such processes may have sinks (deadlock states) in which some resources are not released.

we may informally denote $\prod_{i\in I}\pi_i$ by $\pi_{i_1}\times\cdots\times\pi_{i_n}$. However, since formally an element of $\prod_{i\in I}\pi_i$ is a function on I, the product is a commutative and associative operation.

▶ **Theorem 4.** *For any family of processes $\{\pi_i\}_{i\in I}$, the tuple $\prod_{i\in I}\pi_i$ is a process.* ◀

▶ **Definition 5.** A family of processes $\{\pi_i\}_{i\in I}$ is called "non-interfering" if for any $\langle v_i\rangle_{i\in I}\in State^R(\prod_{i\in I}\pi_i)$,

$$(\exists i\in I\ Alive_{v_i}(\pi_i))\to Alive_{\langle v_i\rangle_{i\in I}}\left(\prod_{i\in I}\pi_i\right).$$

▶ **Definition 6.** For any set I, the set of formulas $\Phi(I)$ is defined recursively: (i) $\bot\in\Phi(I)$, (ii) $(A\parallel B)\in\Phi(I)$, where A and B are two disjoint subsets of I, (iii) $\phi\to\psi\in\Phi(I)$, where $\phi,\psi\in\Phi(I)$.

▶ **Definition 7.** For any family of processes $\mathcal{P}=\{\pi_i\}_{i\in I}$ and any formula $\phi\in\Phi(I)$, we define the binary relation $\mathcal{P}\vDash\phi$ as follows:

1. $\mathcal{P}\nvDash\bot$,
2. $\mathcal{P}\vDash\phi\to\psi$ if and only if $\mathcal{P}\nvDash\phi$ or $\mathcal{P}\vDash\psi$,
3. $\mathcal{P}\vDash A\parallel B$ if and only if the two-element family of processes $\{\prod_{a\in A}\pi_a,\prod_{b\in B}\pi_b\}$ is non-interfering.

   See Section 6.3 for a discussion of an n-ary version of the predicate $A\parallel B$.

3 Axioms

▶ **Definition 8.** The logic of concurrency, in addition to propositional tautologies and the Modus Ponens inference rule, consists of the following axioms:

1. Empty Set: $A\parallel\varnothing$,
2. Symmetry: $A\parallel B\to B\parallel A$,
3. Monotonicity: $A\parallel B,C\to A\parallel B$,
4. Exchange: $A,B\parallel C\to(A\parallel B\to A\parallel B,C)$.

We use notation $X\vdash\phi$ to denote that formula ϕ is provable in our logical system using the set of additional axioms X.

4 Soundness

The proof of soundness of the Geiger-Paz-Pearl axioms with respect to the non-interference semantics is not trivial. We state the soundness of each axiom as a separate theorem.

▶ **Theorem 9** (Empty Set). *No process α interferes with process ϵ.*

Proof. Consider any state $\langle a,\star\rangle\in State^R(\alpha\times\epsilon)$ such that $Alive_a(\alpha)$ or $Alive_\star(\epsilon)$. Note that $\star$ is the only state of process ϵ and, thus, $Alive_\star(\epsilon)$ is false. Hence, $Alive_a(\alpha)$. Thus, there is $a'\in State^R(\alpha)$ such that $(a,a')\in Trans(\alpha)$. Hence, $(\langle a,\star\rangle,\langle a',\star\rangle)\in Trans(\alpha\times\epsilon)$. Therefore, $Alive_{\langle a,\star\rangle}(\alpha\times\epsilon)$. ◀

▶ **Theorem 10** (Symmetry). *If process α does not interfere with process β, then process β does not interfere with process α.*

Proof. Consider any $\langle b,a\rangle\in State^R(\beta\times\alpha)$ such that $Alive_b(\beta)$ or $Alive_a(\alpha)$. By the assumption of the theorem, $Alive_{\langle a,b\rangle}(\alpha\times\beta)$. Since the product is a commutative operation, $Alive_{\langle b,a\rangle}(\beta\times\alpha)$. ◀

Next, we will prove the soundness of the Monotonicity axiom. It will be more convenient to prove the soundness of a slightly more general statement: $A, B \parallel C, D \rightarrow A \parallel C$.

▶ **Theorem 11** (Monotonicity). *For all processes $\alpha, \beta, \gamma, \delta$, if process $\alpha \times \beta$ does not interfere with process $\gamma \times \delta$, then process α does not interfere with process γ.*

Proof. Assume that process $\alpha \times \beta$ does not interfere with process $\gamma \times \delta$ and consider any state

$$\langle a, c\rangle \in State^R(\alpha \times \gamma) \qquad (1)$$

such that $Alive_a(\alpha)$ or $Alive_c(\gamma)$. Without loss of generality, we will assume $Alive_a(\alpha)$. Thus, there is $a' \in State(\alpha)$ such that $(a, a') \in Trans(\alpha)$.

We need to show that $Alive_{\langle a,c\rangle}(\alpha \times \gamma)$. Indeed, assume that process $\alpha \times \gamma$ is deadlocked in state $\langle a, c\rangle$. Since $(a, a') \in Trans(\alpha)$, we must conclude that there is some resource $\rho_0 \in r(a) \setminus r(a')$ such that $\rho_0 \in r(c)$. By Definition 1, $|r(a') \setminus r(a)| < 2$ and, hence, $r(a') \setminus r(a) = \{\rho_0\}$.

Let b_0 and d_0 be the initial states of processes β and δ, respectively. Assumption (1) implies that $\langle a, b_0, c, d_0\rangle \in State^R(\alpha \times \beta \times \gamma \times \delta)$. Let process $\alpha \times \beta \times \gamma \times \delta$ transition from state $\langle a, b_0, c, d_0\rangle$ until it reaches a deadlock state $u \in State^R(\alpha \times \beta \times \gamma \times \delta)$. Since processes α and γ are themselves deadlocked in state $\langle a, c\rangle$, all transitions from $\langle a, b_0, c, d_0\rangle$ to u must have been made by processes β and δ. Thus, $u = \langle a, b, c, d\rangle$ for some $b \in State^R(\beta)$ and $d \in State^R(\delta)$. In other words,

$$\langle a, b, c, d\rangle \in State^R(\alpha \times \beta \times \gamma \times \delta)$$

and

$$\neg Alive_{\langle a,b,c,d\rangle}(\alpha \times \beta \times \gamma \times \delta). \qquad (2)$$

The first of the above statements, by Definition 3, implies that $(r(a) \cup r(c)) \cap r(b) = \varnothing$. Recall that $r(a') \setminus r(a) = \{\rho_0\} \subseteq r(c)$. Thus,

$$r(a') \cap r(b) = ((r(a') \setminus r(a)) \cup (r(a') \cap r(a))) \cap r(b) \subseteq (r(c) \cup r(a)) \cap r(b) = \varnothing.$$

Hence $\langle a', b\rangle \in State(\alpha \times \beta)$. Recall that $(a, a') \in Trans(\alpha)$. Thus, $Alive_{\langle a,b\rangle}(\alpha \times \beta)$. By the assumption of the theorem, process $\alpha \times \beta$ does not interfere with process $\gamma \times \delta$. Hence, $Alive_{\langle\langle a,b\rangle,\langle c,d\rangle\rangle}((\alpha \times \beta) \times (\gamma \times \delta))$. Due to the associativity of the product operation, $Alive_{\langle a,b,c,d\rangle}(\alpha \times \beta \times \gamma \times \delta)$, which contradicts (2). ◀

▶ **Theorem 12** (Exchange). *For all processes α, β, γ, if process $\alpha \times \beta$ does not interfere with process γ and process α does not interfere with process β, then process α does not interfere with process $\beta \times \gamma$.*

Proof. Assume that process $\alpha \times \beta$ does not interfere with process γ and process α does not interfere with process β. Consider any $\langle a, \langle b, c\rangle\rangle \in State^R(\alpha \times (\beta \times \gamma))$. We need to prove that $Alive_{\langle a,\langle b,c\rangle\rangle}(\alpha \times (\beta \times \gamma))$ if either $Alive_a(\alpha)$ or $Alive_{\langle b,c\rangle}(\beta \times \gamma)$. Let us consider these two cases separately.

Case I. If $Alive_a(\alpha)$, then $Alive_{\langle a,b\rangle}(\alpha \times \beta)$, since process α does not interfere with process β. Thus, $Alive_{\langle\langle a,b\rangle,c\rangle}((\alpha \times \beta) \times \gamma))$, because process $\alpha \times \beta$ does not interfere with process γ. Therefore, since the product operation is associative, $Alive_{\langle a,\langle b,c\rangle\rangle}(\alpha \times (\beta \times \gamma))$.

Case II. If $Alive_{\langle b,c\rangle}(\beta \times \gamma)$, then there is $\langle b', c'\rangle \in State(\beta \times \gamma)$ such that $(\langle b, c\rangle, \langle b', c'\rangle) \in Trans(\beta \times \gamma)$ and either $b' = b$ or $c' = c$. Again, we need to consider two separate cases.

First, assume that $b' = b$. Hence, $Alive_c(\gamma)$. Thus, $Alive_{\langle\langle a,b\rangle,c\rangle}((\alpha \times \beta) \times \gamma))$, because process $\alpha \times \beta$ does not interfere with process γ. Therefore, since product is an associative operation, $Alive_{\langle a,\langle b,c\rangle\rangle}(\alpha \times (\beta \times \gamma))$.

Finally, suppose that $c' = c$. Hence, $Alive_b(\beta)$. Thus, $Alive_{\langle a,b\rangle}(\alpha \times \beta)$, because process α does not interfere with process β. Hence, $Alive_{\langle\langle a,b\rangle,c\rangle}((\alpha \times \beta) \times \gamma))$, because process $\alpha \times \beta$ does not interfere with process γ. Therefore, since product is an associative operation, $Alive_{\langle a,\langle b,c\rangle\rangle}(\alpha \times (\beta \times \gamma))$. ◀

5 Completeness

In this section we will prove the completeness of the Geiger-Paz-Pearl axioms with respect to non-interference semantics. This result is stated in Theorem 25. We start, however, with a sequence of lemmas in which we assume a fixed *finite* index set I and a fixed maximal consistent set of formulas $X \subseteq \Phi(I)$.

5.1 Critical Sets

▶ **Definition 13.** A set $C \subseteq I$ is called critical if there is a disjoint partition $C_1 \sqcup C_2$ of C, called a "critical partition", such that

1. $X \nvdash C_1 \parallel C_2$,
2. $X \vdash C_1 \cap D \parallel C_2 \cap D$, for any $D \subsetneq C$.

▶ **Lemma 14.** *Any critical partition is a non-trivial partition.*

Proof. It will be sufficient to prove that for any set A, we have $X \vdash A \parallel \varnothing$ and $X \vdash \varnothing \parallel A$. The first statement is an instance of Empty Set axiom, the second statement follows from Empty Set and Symmetry axioms. ◀

▶ **Lemma 15.** *$X \nvdash A \parallel B$, for any non-trivial (but not necessarily critical) partition $A \sqcup B$ of a critical set C.*

Proof. Suppose $X \vdash A \parallel B$ and let $C_1 \sqcup C_2$ be a critical partition of C. By the Monotonicity and Symmetry axioms, $X \vdash A \cap C \parallel B \cap C$. Thus,

$$X \vdash A \cap C_1, A \cap C_2 \parallel B \cap C_1, B \cap C_2. \tag{3}$$

Since $A \sqcup B$ is a non-trivial partition of C, sets A and B are both non-empty. Thus, $A \subsetneq C$ and $B \subsetneq C$. Hence, by the definition of a critical set, $X \vdash A \cap C_1 \parallel A \cap C_2$ and $X \vdash B \cap C_1 \parallel B \cap C_2$.

Note that $A \cap C$ is not empty since $A \sqcup B$ is a non-trivial partition of C. Thus, either $A \cap C_1$ or $A \cap C_2$ is not empty. Without loss of generality, assume that $A \cap C_1 \neq \varnothing$. From (3) and our earlier observation that $X \vdash A \cap C_1 \parallel A \cap C_2$, the Exchange axiom yields

$$X \vdash A \cap C_1 \parallel A \cap C_2, B \cap C_1, B \cap C_2.$$

By the Symmetry axiom,

$$X \vdash A \cap C_2, B \cap C_1, B \cap C_2 \parallel A \cap C_1. \tag{4}$$

The assumption $A \cap C_1 \neq \varnothing$ implies that $(A \cap C_2) \cup (B \cap C_1) \cup (B \cap C_2) \subsetneq C$. Hence, by the definition of a critical set, $X \vdash B \cap C_1 \parallel A \cap C_2, B \cap C_2$. By Symmetry axiom, $X \vdash A \cap C_2, B \cap C_2 \parallel B \cap C_1$. From (4) and the above statement, using the Exchange axiom,

$X \vdash A \cap C_2, B \cap C_2 \parallel A \cap C_1, B \cap C_1$. Since $A \sqcup B$ is a partition of C, we can conclude that $X \vdash C_2 \parallel C_1$. By the Symmetry axiom, $X \vdash C_1 \parallel C_2$, which contradicts the assumption that $C_1 \sqcup C_2$ is a critical partition.

◀

▶ **Lemma 16.** *For any two disjoint subsets $A, B \subseteq I$, if $X \nvdash A \parallel B$, then there is a critical partition $C_1 \sqcup C_2$, such that $C_1 \subseteq A$ and $C_2 \subseteq B$.*

Proof. Consider the partial order $\preceq$ on set $2^A \times 2^B$ such that $(E_1, E_2) \preceq (F_1, F_2)$ if and only if $E_1 \subseteq F_1$ and $E_2 \subseteq F_2$. Define $\mathcal{E} = \{(E_1, E_2) \in 2^A \times 2^B \mid X \nvdash E_1 \parallel E_2\}$. Note that $(A, B) \in \mathcal{E}$, because $X \nvdash A \parallel B$. Thus, $\mathcal{E}$ is a non-empty finite set. Take (C_1, C_2) to be a minimal element of set $\mathcal{E}$ with respect to partial order $\preceq$. ◀

5.2 Critical Set at a Dinner Table

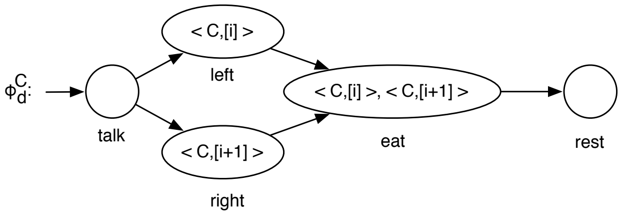

Figure 4 Critical set process ϕ_d^C.

For each critical set $C = \{c_1, \dots, c_n\}$, we formally define the family of "dining philosophers" processes $\mathcal{P}_C = \{\phi_d^C\}_{d \in C} = \{(V, E, q, R^C, r_d^C)\}_{d \in C}$, shown in Figure 4, as follows

1. $V = \{talk, left, right, eat, rest\}$,
2. set E consists of edges $(talk, left)$, $(talk, right)$, $(left, eat)$, $(right, eat)$, $(eat, rest)$,
3. $q = talk$,
4. $R^C = \{C\} \times \mathbb{Z}_n$, or the set of all congruence classes in $\mathbb{Z}_n$ labeled with the critical set C. We will need this label later to distinguish resources of processes corresponding to different critical sets.
5. If $d = c_i$, then $r_d^C(talk) = r_d^C(rest) = \varnothing$, $r_d^C(left) = \{(C, [i])\}$, $r_d^C(right) = \{(C, [i+1])\}$, and $r_d^C(eat) = \{(C, [i]), (C, [i+1])\}$.

▶ **Lemma 17.** *For any $D \subseteq C$,*

$$\langle left \rangle_{d \in D} \in State^R\left(\prod_{d \in D} \phi_d^C\right).$$

Proof. Starting from the initial state $\langle talk \rangle_{d \in D}$, each of the processes $\{\phi_d^C\}_{d \in D}$ can make a transition into state $left$. ◀

▶ **Lemma 18.** *For any $D \subseteq C$,*

$$Alive_{\langle left \rangle_{d \in D}}\left(\prod_{d \in D} \phi_d^C\right) \quad \textit{iff} \quad D \neq C.$$

Proof. ($\Rightarrow$): Suppose $D = C$ and consider $\langle left\rangle_{c\in C}$, a state of process $\prod_{c\in C}\phi_c^C$. In this state no process can make a transition because all resources are already held.
($\Leftarrow$): Let $D \neq C$. Thus, there are more resources than processes. By the Pigeonhole Principle, if not all processes are in *rest* states, than at least one process has both of its resources available and, thus, can make a transition. ◀

▶ **Lemma 19.** *For any critical set C and any two disjoint subsets $A, B \subseteq C$, process $\prod_{a\in A}\phi_a^C$ and process $\prod_{b\in B}\phi_b^C$ interfere if and only if $A \sqcup B$ is a non-trivial partition of C.*

Proof. ($\Rightarrow$): Suppose that $A \sqcup B$ is not a non-trivial partition of C. Thus, either $A \cup B \subsetneq C$ or one of sets A and B is empty. In both of these cases, we need to prove that processes $\prod_{a\in A}\phi_a^C$ and $\prod_{b\in B}\phi_b^C$ do not interfere.
Case I. Suppose that $A \cup B \subsetneq C$. Consider any state

$$\langle s_a, s_b\rangle \in State^R\left(\prod_{a\in A}\phi_a^C \times \prod_{b\in B}\phi_b^C\right),$$

such that $Alive_{s_a}(\prod_{a\in A}\phi_a^C)$ or $Alive_{s_b}(\prod_{b\in B}\phi_b^C)$. Thus, $\langle s_a, s_b\rangle$ is not the state in which all ϕ-processes are already in state *rest*.

Since $A \cup B \subsetneq C$, there are more resources than ϕ-processes in the product $\prod_{a\in A}\phi_a^C \times \prod_{b\in B}\phi_b^C$. Thus, by the Pigeonhole Principle and since not all ϕ-processes are in *rest* states, at least one process has both of its resources available and, thus, can make a transition. Therefore,

$$Alive_{\langle s_a, s_b\rangle}\left(\prod_{a\in A}\phi_a^C \times \prod_{b\in B}\phi_b^C\right).$$

Case II. If one of sets A and B is empty, then the desired property follows from Theorem 9.
($\Leftarrow$): Consider state $\langle\langle left\rangle_{a\in A}, \langle left\rangle_{b\in B}\rangle$. By Lemma 17,

$$\langle\langle left\rangle_{a\in A}, \langle left\rangle_{b\in B}\rangle \in State^R\left(\prod_{a\in A}\phi_a^C \times \prod_{b\in B}\phi_b^C\right).$$

By Lemma 18, however,

$$Alive_{\langle left\rangle_{a\in A}}\left(\prod_{a\in A}\phi_a^C\right), \qquad Alive_{\langle left\rangle_{b\in B}}\left(\prod_{b\in B}\phi_b^C\right)$$

and

$$\neg Alive_{\langle\langle left\rangle_{a\in A}, \langle left\rangle_{b\in B}\rangle}\left(\prod_{a\in A}\phi_a^C \times \prod_{b\in B}\phi_b^C\right).$$

Therefore, processes $\prod_{a\in A}\phi_a^C$ and $\prod_{b\in B}\phi_b^C$ interfere. ◀

▶ **Lemma 20.** *For any two disjoint subsets $A, B \subseteq C$, if $X \vdash A \parallel B$, then processes $\prod_{a\in A}\phi_a^C$ and $\prod_{b\in B}\phi_b^C$ do not interfere.*

Proof. Suppose that processes $\prod_{a\in A}\phi_a^C$ and $\prod_{b\in B}\pi_b^C$ interfere. Thus, by Lemma 19, sets A and B form a non-trivial disjoint partition of set C. Hence, by Lemma 15, $X \nvdash A \parallel B$. ◀

▶ **Lemma 21.** *For any two families of processes $\{\alpha_j\}_{j\in J}$ and $\{\beta_j\}_{j\in J}$ such that sets $Res(\alpha_{j_1} \times \beta_{j_1})$ and $Res(\alpha_{j_2} \times \beta_{j_2})$ are disjoint for any $j_1 \neq j_2$, processes $\prod_{j\in J}\alpha_j$ and $\prod_{j\in J}\beta_j$ are non-interfering if and only if processes α_j and β_j are non-interfering for each $j \in J$.*

Proof. ($\Rightarrow$): Suppose that there is some j_0 such that processes α_{j_0} and β_{j_0} interfere. Thus, there is a state $\langle a,b\rangle \in State^R(\alpha_{j_0}\times\beta_{j_0})$ such that $\neg Alive_{\langle a,b\rangle}(\alpha_{j_0}\times\beta_{j_0})$ and either $Alive_a(\alpha_{j_0})$ or $Alive_b(\beta_{j_0})$. Without loss of generality, we will assume that $Alive_a(\alpha_{j_0})$.

Let $\langle q_j\rangle_{j\in J}$ be the initial state of process $\prod_{j\in J}(\alpha_j\times\beta_j)$. Define

$$q'_j = \begin{cases} \langle a,b\rangle & \text{if } j = j_0,\\ q_j & \text{otherwise.}\end{cases}$$

Since $\langle a,b\rangle \in State^R(\alpha_{j_0}\times\beta_{j_0})$, we can conclude that $\langle q'_j\rangle_{j\in J} \in State^R(\prod_{j\in J}(\alpha_j\times\beta_j))$. Let process $\prod_{j\in J}(\alpha_j\times\beta_j)$ start at state $\langle q'_j\rangle_{j\in J}$ and run until it reaches a state $\langle q''_j\rangle_{j\in J}\in State^R(\prod_{j\in J}(\alpha_j\times\beta_j))$ such that $\neg Alive_{\langle q''_j\rangle_{j\in J}}(\prod_{j\in J}(\alpha_j\times\beta_j))$. Let $q''_j = \langle a''_j, b''_j\rangle$. Because the product is a commutative and associative operation,

$$\neg Alive_{\langle\langle a''_j\rangle_{j\in J},\langle b''_j\rangle_{j\in J}\rangle}\left(\left(\prod_{j\in J}\alpha_j\right)\times\left(\prod_{j\in J}\beta_j\right)\right) \tag{5}$$

Since $\neg Alive_{\langle a,b\rangle}(\alpha_{j_0}\times\beta_{j_0})$, we can claim that $\langle a''_{j_0}, b''_{j_0}\rangle = q''_{j_0} = q'_{j_0} = \langle a,b\rangle$. Recall now our assumption that $Alive_a(\alpha_{j_0})$. Thus, $Alive_{a''_{j_0}}(\alpha_{j_0})$. Since, by the assumption of the lemma, any process α_j, where $j\neq j_0$, does not share resources with with process α_{j_0}, we can conclude $Alive_{\langle a''_j\rangle_{j\in J}}(\prod_{j\in J}\alpha_j)$. This, in conjunction with (5), implies that processes $\prod_{j\in J}\alpha_j$ and $\prod_{j\in J}\beta_j$ interfere.

($\Leftarrow$): Suppose that processes $\prod_{j\in J}\alpha_j$ and $\prod_{j\in J}\beta_j$ interfere. Thus, there is a state

$$\langle\langle a_j\rangle_{j\in J},\langle b_i\rangle_{j\in J}\rangle \in State^R\left(\left(\prod_{j\in J}\alpha_j\right)\times\left(\prod_{j\in J}\beta_j\right)\right)$$

such that

$$\neg Alive_{\langle\langle a_j\rangle_{j\in J},\langle b_j\rangle_{j\in J}\rangle}\left(\left(\prod_{j\in J}\alpha_j\right)\times\left(\prod_{j\in J}\beta_j\right)\right) \tag{6}$$

but either $Alive_{\langle a_j\rangle_{j\in J}}(\prod_{j\in J}\alpha_j)$ or $Alive_{\langle b_j\rangle_{j\in J}}(\prod_{j\in J}\beta_j)$. Without loss of generality, assume that $Alive_{\langle a_j\rangle_{j\in J}}(\prod_{j\in J}\alpha_j)$. Thus, there is an $j_0\in J$ such that $Alive_{a_{j_0}}(\alpha_{j_0})$. Hence, $Alive_{\langle a_{j_0},b_{j_0}\rangle}(\alpha_{j_0}\times\beta_{j_0})$, because, by the assumption of the lemma, processes α_{j_0} and β_{j_0} do not interfere. Thus, there is a state $\langle a',b'\rangle$ such that $(\langle a_{j_0},b_{j_0}\rangle,\langle a',b'\rangle)\in Trans(\alpha_{j_0}\times\beta_{j_0})$. Since, by the assumption of the lemma, process $\alpha_{j_0}\times\beta_{j_0}$ does not share resources with any process $\alpha_j\times\beta_j$ such that $j\neq j_0$, the same transition is available to process $\prod_{j\in J}(\alpha_j\times\beta_j)$. Thus, $Alive_{\langle\langle a_j,b_j\rangle\rangle_{j\in J}}(\prod_{j\in J}(\alpha_j\times\beta_j))$. Due to the commutativity and associativity of the product, $Alive_{\langle\langle a_j\rangle_{j\in J},\langle b_j\rangle_{j\in J}\rangle}\left((\prod_{j\in J}\alpha_j)\times(\prod_{j\in J}\beta_j)\right)$, which contradicts (6). ◀

▶ **Definition 22.** $\mathcal{P} = \{\prod_{C\ni i}\phi_i^C\}_{i\in I}$, where the product is computed over all critical subsets C of I that contain i.

▶ **Lemma 23.** *For any disjoint subsets $A\subseteq I$ and $B\subseteq I$, $X\vdash A\parallel B$ if and only if $\mathcal{P}\vDash A\parallel B$.*

Proof. ($\Rightarrow$): Let $\mathcal{P}\nvDash A\parallel B$. Thus, processes $\prod_{a\in A}\prod_{C\ni a}\phi_a^C$ and $\prod_{b\in B}\prod_{C\ni b}\phi_b^C$ interfere. Hence, since the product operation is commutative and associative, processes $\prod_{C\in\mathcal{C}}\prod_{a\in A\cap C}\phi_a^C$ and $\prod_{C\in\mathcal{C}}\prod_{b\in B\cap C}\phi_b^C$ interfere, where $\mathcal{C}$ is the set of all critical subsets of set I.

For any two different critical sets C_1 and C_2, the set of resources available to process $\prod_{a\in A\cap C_1}\phi_a^{C_1}$ is $\cup_a R^{C_1}=\{C_1\}\times\mathbb{Z}_{|C_1|}$ and the set of resources available to process $\prod_{b\in B\cap C_2}\phi_b^{C_2}$ is $\cup_b R^{C_2}=\{C_2\}\times\mathbb{Z}_{|C_2|}$. These two sets of resources are disjoint since $C_1\neq C_2$.

By Lemma 21, there must be a critical set $C_0\in\mathcal{C}$ such that processes $\prod_{a\in A\cap C_0}\phi_a^{C_0}$ and $\prod_{b\in B\cap C_0}\phi_b^{C_0}$ interfere. Hence, by Lemma 20, $X\nvdash A\cap C_0\parallel B\cap C_0$. By the Monotonicity axiom, $X\nvdash A\cap C_0\parallel B$. By the Symmetry axiom, $X\nvdash B\parallel A\cap C_0$. Again by the Monotonicity axiom, $X\nvdash B\parallel A$. By the Symmetry axiom, $X\nvdash A\parallel B$, which is a contradiction.

$(\Leftarrow)$: Let $X\nvdash A\parallel B$. By Lemma 16, there is a critical set C such that $(A\cap C)\sqcup(B\cap C)$ is a critical partition of C. By Lemma 14, partition $(A\cap C)\sqcup(B\cap C)$ is a non-trivial partition of the critical set C. Thus, by Lemma 19, processes $\prod_{a\in A\cap C}\phi_a^C$ and $\prod_{b\in B\cap C}\phi_b^C$ interfere. By Lemma 21, processes $\prod_{C\in\mathcal{C}}\prod_{a\in A\cap C}\phi_a^C$ and $\prod_{C\in\mathcal{C}}\prod_{b\in B\cap C}\phi_b^C$ interfere. Since the product is a commutative and associative operation, processes $\prod_{a\in A}\prod_{C\ni a}\phi_a^C$ and $\prod_{b\in B}\prod_{C\ni b}\phi_b^C$ interfere. Therefore, $\mathcal{P}\nvDash A\parallel B$. ◀

▶ **Lemma 24.** *For any $\psi\in\Phi(I)$, $X\vdash\psi$ if and only if $\mathcal{P}\vDash\psi$.*

Proof. We use induction on structural complexity of formula ψ. The base case follows from Lemma 23, and the inductive case, after taking into account Definition 7, is straightforward. ◀

▶ **Theorem 25** (completeness)**.** *For any ϕ, if $\nvdash\phi$, then there is a family of processes $\mathcal{P}=\{\pi_i\}_{i\in I}$ such that $\mathcal{P}\nvDash\phi$.*

Proof. Assume that $\nvdash\phi$. Let I be the (finite) set of all indices used in formula ϕ and X be a maximal consistent subset of $\Phi(I)$ that contains formula $\neg\phi$. By Lemma 24, $\mathcal{P}\nvDash\phi$. ◀

6 Conclusions

6.1 The Monotonicity Axiom, Revisited

We will show that the Monotonicity axiom is not sound if the *acquire one resource at a time* condition is removed from Definition 1. Indeed, consider three "processes" specified by the DAGs in Figure 5. It will be sufficient to show that processes α and $\beta\times\gamma$ do not interfere, but processes α and β do interfere.

▶ **Theorem 26.** *Processes α and $\beta\times\gamma$ do not interfere.*

Proof. Consider any state $\langle a,\langle b,c\rangle\rangle\in State^R(\alpha\times(\beta\times\gamma))$ and assume that $\neg Alive_{\langle a,\langle b,c\rangle\rangle}(\alpha\times(\beta\times\gamma))$. We need to show that $\neg Alive_a(\alpha)$ and $\neg Alive_{\langle b,c\rangle}(\beta\times\gamma)$. Indeed, notice that the graph of process $\alpha\times(\beta\times\gamma)$ has only two sinks, thus the tuple $\langle a,\langle b,c\rangle\rangle$ has only two possible values.

Case 1: $\langle a,\langle b,c\rangle\rangle=\langle 2,\langle 3,2\rangle\rangle$. Note that $\neg Alive_2(\alpha)$ and $\neg Alive_{\langle 3,2\rangle}(\beta\times\gamma)$.
Case 2: $\langle a,\langle b,c\rangle\rangle=\langle 2,\langle 1,2\rangle\rangle$. Note again that $\neg Alive_2(\alpha)$ and $\neg Alive_{\langle 1,2\rangle}(\beta\times\gamma)$. ◀

▶ **Theorem 27.** *Processes α and β interfere.*

Proof. Consider state $\langle 2,1\rangle\in State^R(\alpha\times\beta)$ and notice that $\neg Alive_{\langle 2,1\rangle}(\alpha\times\beta)$, but $Alive_1(\beta)$. ◀

6.2 The Public Knowledge Axiom

In Definition 6, we assumed that for $A \parallel B$ to be a valid formula, sets A and B must be disjoint. In the case of independence of secrets, More and Naumov [13] did not make this assumption. They noticed that $A \parallel A$ implies that each secret in set A has a constant and, thus, "publicly known" value. This led to an additional *Public Knowledge* axiom for independence:

$$A \parallel A \rightarrow (B \parallel C \rightarrow A, B \parallel C). \tag{7}$$

This Public Knowledge axiom, together with the Empty Set, Symmetry, Monotonicity, and Exchange axioms, forms a sound and complete system for the independence of secrets in information flow.

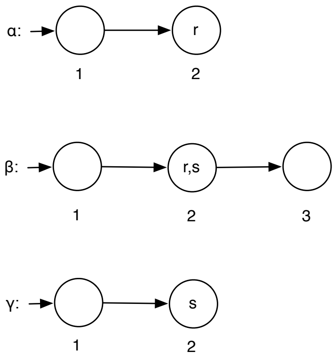

Figure 5 Monotonicity "counter-example."

Although Geiger, Paz, and Pearl [8] assumed that sets A and B were disjoint, this assumption is not important in their work. Indeed, it is easy to see that $A \parallel A$, under probability semantics, means that each variable in set A is constant almost everywhere. This means that the Public Knowledge axiom is also valid under the probability semantics. Finally, a review of the Geiger-Paz-Pearl completeness proof shows that a similar argument can be made in this more general setting if the Public Knowledge axiom is added to the system.

The case of concurrency semantics, however, is less straightforward. It depends on exactly what it means when the same process appears on both sides of $A \parallel B$. If $v \in A \cap B$, then one option is to assume that different occurrences of v refer to the same instance of a process. The other option is to assume that they refer to two different instances of the process. The former option requires them to have the same DAG and to be in the same states at any given time. The latter means that they have the same DAG, but could be in different states at any given time.

Under the first interpretation, $A \parallel A$ implies that each of processes in A cannot require any resources in reachable states, since otherwise both copies of the process would need to acquire the same resource. If the set of processes A does not require any resources in any of its reachable states, then it cannot affect interference between the other processes. Thus, the Public Knowledge axiom is sound. Moreover, the proof in this paper can be modified to show that the logical system formed by Empty Set, Symmetry, Monotonicity, Exchange, and Public Knowledge is complete under this interpretation.

Under the second interpretation, however, the Public Knowledge axiom is not sound. Indeed, consider the three processes α, β, and γ that have access to three resources r, s, and t. Each of the processes needs any two out of the three resources in order to terminate. Formally, all three of these processes have the same cube-shaped DAG, which is depicted in Figure 6. In some sense, this is a modified version of the Dining Philosopher's problem, where each of the philosophers α, β, and γ can use any two out of the three forks on the table.

Note that formula $\beta \parallel \gamma$ is true, because there are more resources than processes, thus, by the Pigeonhole Principle, at any state of $\beta \times \gamma$, either all processes have already terminated or one of the them has enough resources available to make a transition. For the same reason, formula $\alpha \parallel \alpha$ is also true as long as α on the left-hand-side and α on the right-hand-side refer to to two different instances of α.

Finally, formula $\alpha, \beta \parallel \gamma$ is false, because if processes α, β, and γ, respectively, acquire resources r, s, and t, then the system deadlocks. Therefore, the Public Knowledge axiom (7) is not sound if A, B, and C represent processes α, β, and γ.

6.3 An n-ary Non-interference Relation

In this paper, we considered the non-interference relation $A \parallel B$ between two sets of process. This binary relation can be generalized naturally to the n-ary relation $A_1 \parallel A_2 \parallel \cdots \parallel A_n$ between n sets of processes by changing part 4 of Definition 7 to

4. *$\mathcal{P} \vDash A_1 \parallel A_2 \parallel \cdots \parallel A_n$ if and only if the n-element family of processes $\{\prod_{a \in A_i} \pi_a\}_{i \le n}$ is non-interfering.*

It turns out, however, that the n-ary non-interference relation can be expressed through the binary non-interference relation studied in this paper. For example, in the case where $n = 3$, the following result holds:

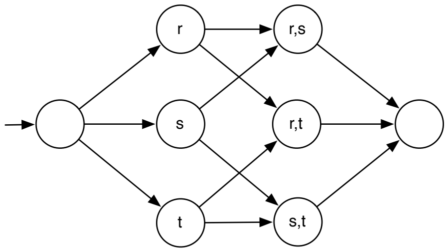

Figure 6 Cube DAG.

▶ **Theorem 28.** *For any family of processes $\mathcal{P} = \{\pi_i\}_{i \in I}$ and any subsets A, B, and C of set I,*

$$\mathcal{P} \vDash (A \parallel B \parallel C) \iff (A \parallel B, C) \wedge (B \parallel C).$$

Proof. In the following proof, we let α denote $\prod_{i \in A} \pi_i$, β denote $\prod_{i \in B} \pi_i$, and γ denote $\prod_{i \in C} \pi_i$.

$(\Rightarrow)$: First, assume that $\mathcal{P} \nvDash A \parallel B, C$. Thus, there is a state $\langle a, b, c\rangle \in State^R(\alpha \times \beta \times \gamma)$ such that $\neg Alive_{\langle a,b,c\rangle}(\alpha \times \beta \times \gamma)$, but either $Alive_a(\alpha)$ or $Alive_{\langle b,c\rangle}(\beta \times \gamma)$. The latter implies that either $Alive_b(\beta)$ or $Alive_c(\gamma)$. Hence, we can conclude that at least one of the following three statements is true: $Alive_a(\alpha)$, $Alive_b(\beta)$, or $Alive_c(\gamma)$. By the assumption $\mathcal{P} \vDash A \parallel B \parallel C$, we can conclude that $Alive_{\langle a,b,c\rangle}(\alpha \times \beta \times \gamma)$, which is a contradiction.

Second, suppose that $\mathcal{P} \nvDash B \parallel C$. Thus, there is a state $\langle b, c\rangle \in State^R(\beta \times \gamma)$ such that $\neg Alive_{\langle b,c\rangle}(\beta \times \gamma)$, but either $Alive_b(\beta)$ or $Alive_c(\gamma)$. Let a_0 be the initial state of process α. Thus, $\langle a_0, b, c\rangle \in State^R(\alpha \times \beta \times \gamma)$. Let process $\alpha \times \beta \times \gamma$ make as many transitions as possible from state $\langle a_0, b, c\rangle$ until it reaches a state $\langle a', b', c'\rangle$ such that $\neg Alive_{\langle a',b',c'\rangle}(\alpha \times \beta \times \gamma)$. Note that $\neg Alive_{\langle b,c\rangle}(\beta \times \gamma)$ implies that $b' = b$ and $c' = c$. Thus, $\neg Alive_{\langle a',b,c\rangle}(\alpha \times \beta \times \gamma)$. However, we proved earlier that $Alive_b(\beta)$ or $Alive_c(\gamma)$. This contradicts our assumption that $\mathcal{P} \vDash A \parallel B \parallel C$.

$(\Leftarrow)$: Let $\mathcal{P} \nvDash A \parallel B \parallel C$. Thus, there is a state $\langle a, b, c\rangle \in State^R(\alpha \times \beta \times \gamma)$ such that $\neg Alive_{\langle a,b,c\rangle}(\alpha \times \beta \times \gamma)$, but $Alive_a(\alpha)$, $Alive_b(\beta)$, or $Alive_c(\gamma)$.

If $Alive_a(\alpha)$, then, by the assumption $\mathcal{P} \vDash A \parallel B, C$, we can conclude that $Alive_{\langle a,b,c\rangle}(\alpha \times \beta \times \gamma)$, which is a contradiction.

If $Alive_b(\beta)$ or $Alive_c(\gamma)$, then $Alive_{\langle b,c\rangle}(\beta \times \gamma)$, by the assumption that $\mathcal{P} \vDash B \parallel C$. Thus, because $\mathcal{P} \vDash A \parallel B, C$, we have $Alive_{\langle a,b,c\rangle}(\alpha \times \beta \times \gamma)$, which is a contradiction. ◀

7 Acknowledgments

The authors would like to thank Joseph Halpern for pointing to the connection between the Geiger, Paz, and Pearl results on probabilistic independence and the first two authors' work on independence in information flow; and Sergei Artemov for the encouragement to look for new semantics of the axioms of independence.

References

1 Ellis Cohen. Information transmission in computational systems. In *Proceedings of Sixth ACM Symposium on Operating Systems Principles*, pages 113–139. Association for Computing Machinery, 1977.

2 Abraham de Moivre. De mensura sortis seu; de probabilitate eventuum in ludis a casu fortuito pendentibus. *Philosophical Transactions (1683-1775)*, 27:pp. 213–264, 1711.

3 Abraham de Moivre. *Doctrine of Chances.* 1718.

4 Edsger W. Dijkstra. Hierarchical ordering of sequential processes. *Acta Inf.*, 1:115–138, 1971.

5 Michael Donders, Sara Miner More, and Pavel Naumov. Information flow on directed acyclic graphs. In L. Beklemishev and R. de Queiroz, editors, *Proceedings of 18th Workshop on Logic, Language, Information and Computation.* Springer, 2011. (to appear).

6 Moivre, Abraham, de. In *The New Encyclopædia Britannica*, volume 8, page 226. Encyclopædia Britannica, 15th edition, 1998.

7 Hubert Garavel. Reflections on the future of concurrency theory in general and process calculi in particular. *Electr. Notes Theor. Comput. Sci.*, 209:149–164, 2008.

8 Dan Geiger, Azaria Paz, and Judea Pearl. Axioms and algorithms for inferences involving probabilistic independence. *Inform. and Comput.*, 91(1):128–141, 1991.

9 Joseph Y. Halpern and Kevin R. O'Neill. Secrecy in multiagent systems. *ACM Trans. Inf. Syst. Secur.*, 12(1):1–47, 2008.

10 C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 26(1):100–106, 1983.

11 Sara Miner More and Pavel Naumov. On interdependence of secrets in collaboration networks. In *Proceedings of 12th Conference on Theoretical Aspects of Rationality and Knowledge (Stanford University, 2009)*, pages 208–217, 2009.

12 Sara Miner More and Pavel Naumov. Hypergraphs of multiparty secrets. In *11th International Workshop on Computational Logic in Multi-Agent Systems CLIMA XI (Lisbon, Portugal), LNAI 6245*, pages 15–32. Springer, 2010.

13 Sara Miner More and Pavel Naumov. An independence relation for sets of secrets. *Studia Logica*, 94(1):73–85, 2010.

14 Milan Studený. Conditional independence relations have no finite complete characterization. In *Information Theory, Statistical Decision Functions and Random Processes. Transactions of the 11th Prague Conference vol. B*, pages 377–396. Kluwer, 1990.

15 David Sutherland. A model of information. In *Proceedings of Ninth National Computer Security Conference*, pages 175–183, 1986.