Information Flow on Directed Acyclic Graphs

Michael Donders, Sara Miner More, and Pavel Naumov

Department of Mathematics and Computer Science
McDaniel College, Westminster, Maryland 21157, USA
{msd002,smore,pnaumov}@mcdaniel.edu

Abstract. The paper considers a multi-argument *independence* relation between messages sent over the edges of a directed acyclic graph. This relation is a generalization of a relation known in information flow as nondeducibility. A logical system that describes the properties of this relation for an arbitrary fixed directed acyclic graph is introduced and proven to be complete and decidable.

1 Introduction

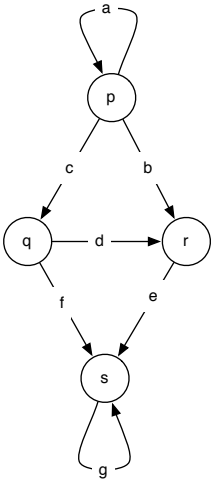


Fig. 1. Graph G_0.

In this paper we study information flow on directed acyclic graphs. We view directed graphs as communication networks in which vertices are parties and directed edges are communication channels. An example of such a graph, G_0, is depicted in Figure 1. We use loop edges to represent values that are computed by the party, but not sent to anyone else. The conditions that parties must observe while communicating over the network will be called *action relations*. The set of action relations for all vertices will be called a *protocol*. Here is a sample protocol $\mathcal{P}_0$ over graph G_0: vertex p picks a random boolean value $a \in \{0,1\}$ and finds two boolean values c and b such that $a \equiv b + c \pmod 2$. It sends value c to vertex q and value b to vertex r. Vertex q finds boolean values d and f such that $c \equiv d + f \pmod 2$ and sends them to vertices r and s, respectively. Vertex r computes the value $e \equiv d + b \pmod 2$ and sends it to vertex s. Vertex s computes value $g \equiv f + e \pmod 2$.

An assignment of values to all channels that satisfies all action relations will be called a *run* of the protocol. Note that for the protocol described above, values c and b are independent in the sense that any possible value of c may occur on the same run with any possible value of b. We denote this by $[c, b]$. This relation between two values was originally introduced by Sutherland [1] and later became known in the study of information flow as *nondeducibility*. Halpern and O'Neill [2] proposed a closely-related notion called f-secrecy. More and Naumov [3] generalized nondeducibility to a relation between an arbitrary set of values and called it independence. For example, values c, b, and d for the above protocol are independent in the sense that any

combination of their possible values may occur on the same run. We denote this relation by $[c, b, d]$. At the same time, it is easy to see that under the above protocol:

$$g \equiv f + e \equiv f + (d + b) \equiv (f + d) + b \equiv c + b \equiv a \pmod 2. \tag{1}$$

Thus, not every combination of values of a and g can occur together on a run. In our notation: $\neg[a, g]$.

The properties mentioned above are specific to the given protocol. If the protocol changes, some of the true properties can become false and vice versa. In this paper, however, we focus on the properties that are true for a given graph no matter which protocol is used. An example of such property for the above graph is $[c, b, f, e] \to [a, g]$. It says that for any protocol under G_0 if values c, b, f and e are independent over this protocol, then values a and g are also independent under the same protocol. We will formally prove this claim in Proposition 3.

The main result of this paper is a sound and complete logical system that describes all propositional properties of the multi-argument relation $[a_1, \ldots, a_n]$ on directed graphs which are acyclic, with the possible exception of loop edges. Previously, More and Naumov obtained similar results for undirected graphs [3] and hypergraphs [4]. Compared to the case of undirected graphs, the logical system described here adds an additional Directed Truncation inference rule.

Our logical system describes information flow properties of a graph, not a specific protocol over this graph. However, this system can be used to reason about the properties of a specific protocol by treating some properties of the protocol as axioms, then using our system to derive additional properties of the protocol.

2 Protocol: A Formal Definition

Throughout this work, by a graph we mean a finite directed graph with cycles of length no more than one or, less formally, "directed acyclic graphs with loops". Such graphs define a partial order on vertices that will be assumed to be the order in which the protocol is executed. The protocol will specify how the values on outgoing edges are related to the values one the incoming edges of each vertex. With this in mind, we will count loops at any vertex v among its outgoing edges $Out(v)$, but *not* among its incoming edges $In(v)$.

Definition 1. *A protocol over a graph $G = \langle V, E\rangle$ is a pair $\langle M, \Delta\rangle$ such that*

1. *$M(e)$ is an arbitrary set of values ("messages") for each edge $e \in E$,*
2. *$\Delta = \{\Delta_v\}_{v\in V}$ is a family of action relations between values of incoming and outgoing edges of the vertex v:*

$$\Delta_v \subseteq \left(\prod_{e\in In(v)} M(e)\right) \times \left(\prod_{e\in Out(v)} M(e)\right).$$

3. **(continuity condition)** *For any possible tuple of values on the incoming edges of a vertex v, there is at least one tuple of values possible on its outgoing edges:*

$$\forall x \in \prod_{e \in In(v)} M(e) \quad \exists y \in \prod_{e \in Out(v)} M(e) \Big((x, y) \in \Delta_v \Big).$$

The continuity condition above distinguishes protocols over directed graphs from protocols over undirected graphs [3].

Definition 2. *A run of a protocol $\mathcal{P} = \langle M, \Delta \rangle$ over graph $G = \langle V, E \rangle$ is any function r on E such that*

1. *$r(e) \in M(e)$ for each $e \in E$,*
2. *$\langle \langle r(e) \rangle_{e \in In(v)}, \langle r(e) \rangle_{e \in Out(v)} \rangle \in \Delta_v$ for each $v \in V$.*

The set of runs of a protocol $\mathcal{P}$ is denoted by $\mathcal{R}(\mathcal{P})$.

Definition 3. *A protocol $\mathcal{P} = \langle M, \Delta \rangle$ over graph $G = \langle E, V \rangle$ is called finite if the set $M(e)$ is finite for each edge $e \in E$.*

We conclude with the definition of a multi-argument version of Sutherland's binary nondeducibility predicate called *independence*. It is identical to the one used by More and Naumov [3, 4].

Definition 4. *A set of edges A is called independent under protocol $\mathcal{P}$ if for any family of runs $\{r_a\}_{a \in A} \subseteq \mathcal{R}(\mathcal{P})$ there is a run $r \in \mathcal{R}(\mathcal{P})$ such that $r(a) = r_a(a)$ for each $a \in A$.*

In the above definition, we refer to the value $r_a(a)$, rather than an arbitrary element of $M(a)$, because there may be some values in $M(a)$ that are not actually used on any given run. In the next section, we will formally define the formulas of our logic system. The atomic formula expressing the independence of a set A will be denoted by $[A]$.

3 Semantics

Informally, by $\Phi(G)$ we denote the set of all propositional properties of independence over a fixed graph $G = \langle V, E \rangle$. Formally, $\Phi(G)$ is a minimal set defined recursively as follows: (i) for any finite set of edges $A \subseteq E$, formula $[A]$ belongs to set $\Phi(G)$, (ii) the false constant $\bot$ belongs to $\Phi(G)$, and (iii) for any formulas ϕ and ψ in $\Phi(G)$, the implication $\phi \to \psi$ also belongs to $\Phi(G)$. Conjunction, disjunction, and negation will be assumed to be defined through connectives $\to$ and $\bot$.

Next, we define the relation $\mathcal{P} \vDash \phi$ between a protocol $\mathcal{P}$ over graph G and a formula $\phi \in \Phi(G)$. Informally, $\mathcal{P} \vDash \phi$ means that formula ϕ is true under $\mathcal{P}$.

Definition 5. *For any protocol $\mathcal{P}$ over a graph G, and any formula $\phi \in \Phi(G)$, we define the relation $\mathcal{P} \vDash \phi$ recursively as follows: (i) $\mathcal{P} \nvDash \bot$, (ii) $\mathcal{P} \vDash [A]$ if the set of edges A is independent under protocol $\mathcal{P}$, (iii) $\mathcal{P} \vDash \phi_1 \to \phi_2$ if $\mathcal{P} \nvDash \phi_1$ or $\mathcal{P} \vDash \phi_2$.*

We will illustrate this definition with the two propositions below. By G_0 we mean the graph depicted earlier in Figure 1.

Proposition 1. *There is a protocol $\mathcal{P}$ over G_0 such that $\mathcal{P} \nvDash [b, f, g] \to [a, g]$.*

Proof. Consider the protocol $\mathcal{P}$ under which the party (represented by vertex) p picks a boolean value a and sends it via edge c to party q . In other words, $a = c$ is the action relation at vertex p. At the same time, the constant value 0 is sent via edge b, which means that $M(b) = \{0\}$. Party q resends value c through edge d and sends the constant 0 through edge f. Party r then resends value d through edge e and, finally, s resends value e through channel g. Under this protocol, $M(b) = M(f) = \{0\}$. Thus, any possible values of edges b, f, and g may occur on the same run. In other words, $\mathcal{P} \vDash [b, f, g]$. At the same time, $a = c = d = e = g$, and $M(a) = M(g) = \{0, 1\}$. Thus, not every combination of values of a and g can occur on the same run. Therefore, $\mathcal{P} \nvDash [a, g]$. □

Note that in the proof of the previous proposition direction of edge d is important. One might expect that the result is not true if the direction of the edge d is reversed. This, however, is not true:

Proposition 2. *There is a protocol $\mathcal{P}$ over G_0 such that $\mathcal{P} \nvDash [c, e, g] \to [a, g]$.*

Proof. Consider the protocol $\mathcal{P}_0$ over G_0 described in the introduction. It was shown earlier through equality (1), that $\mathcal{P}_0 \nvDash [a, g]$. Thus, we only need to prove that $\mathcal{P}_0 \vDash [c, e, g]$. Let c_0, e_0, g_0 be any boolean values. We will show that these values can co-exist on the same run. Indeed, let $f_0 = e_0 + g_0 \pmod 2$, $d_0 = c_0 + f_0 \pmod 2$, $b_0 = d_0 + e_0 \pmod 2$, and $a_0 = c_0 + b_0 \pmod 2$. It is easy to see that values $a_0, b_0, c_0, d_0, e_0, f_0$, and g_0 form a valid run of $\mathcal{P}_0$. □

In this paper, we study the set of formulas that are true under *any* protocol $\mathcal{P}$ as long as the graph G remains fixed. The set of all such formulas will be captured by our logical system for information flow over directed acyclic graphs. This system is described in Section 5.

4 Graph Notation

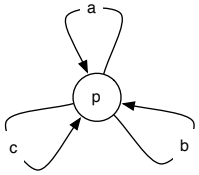
Before the introduction of our formal system, we need to define some graph-related notation that will be used in this system.

A *cut* of a graph is a disjoint partitioning of its vertices into two sets. A *crossing edge* of a cut is an edge whose ends belong to different sets of the partition. For any set of vertices X of a graph G, we use $E(X)$ to denote the set of all edges of G whose ends both belong to X.

Definition 6. *Let G be an arbitrary graph and (X, Y) be an arbitrary cut of G. We define the "truncation" graph G_X of graph G as follows:*

1. *The vertices of graph G_X are the vertices of set X.*
2. *The edges of G_X are all of the edges from $E(X)$ plus the crossing edges of the cut (X, Y) modified in the following way: if, in graph G, a crossing edge c connects vertex $v \in X$ with a vertex in Y, then, in graph G_X, edge c loops from v back into v.*

Each edge e in a truncated graph G_X corresponds to a unique edge in the original graph G. Although the two corresponding edges might connect different vertices in their respective graphs, we will refer to both of them as edge e. For example, graph G'_0 in Figure 2 is obtained from graph G_0 in Figure 1 by truncating along the cut $(\{p, s\}, \{q, r\})$. In the above notation, this truncated graph can be denoted by $(G_0)_{\{p,s\}}$.



Definition 7. *A cut (X, Y) is called "directed" if there are no crossing edges of this cut that lead from Y to X.*

Definition 8. *A gateway between sets of edges A and B in a graph G is a set of edges W such that every undirected path from A to B contains at least one edge from W.*

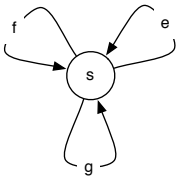


Fig. 2. Graph G'_0.

Note that sets A, B, and W are not necessarily disjoint. Thus, for example, for any set of edges A, set A is a gateway between A and itself. Also, note that the empty set is a gateway between any two components of the graph that are not connected one to another.

5 Formal System: Axioms and Rules

We are now ready to describe our logical system for information flow over directed acyclic graphs. We will write $G \vdash \phi$ to state that formula $\phi \in \Phi(G)$ is provable in this logic. Everywhere below, X, Y denotes the union of sets X and Y. In addition to all propositional tautologies and the Modus Ponens inference rule, the deductive system for this logic consists of the *Small Set* axiom, the *Gateway* axiom, and the *Truncation* and the *Directed Truncation* inference rules:

Small Set Axiom. Any set that contains less than two edges is independent: $G \vdash [A]$, where $A \subseteq E$ and $|A| < 2$.

Gateway Axiom. $G \vdash [A, W] \rightarrow ([B] \rightarrow [A, B])$, where W is a gateway between sets of edges A and B such that $A \cap W = \varnothing$.

Truncation Rule. Let C be the set of all crossing edges of a cut (X, Y) and ϕ be a formula in $\Phi(G_X)$. If $G_X \vdash \phi$, then $G \vdash [C] \rightarrow \phi$.

Directed Truncation Rule. Let (X, Y) be a directed cut and $\phi \in \Phi(G_X)$. If $G_X \vdash \phi$, then $G \vdash \phi$.

The soundness of this system will be demonstrated in Section 6 and its completeness in Section 7. Below, we present a general result to which we will refer during the proof of completeness.

Theorem 1 (monotonicity). *$G \vdash [A] \to [B]$, for any graph G and any subsets $B \subseteq A$ of edges of G.*

Proof. Consider sets B and $\varnothing$. Since there are no paths connecting these sets, any set of edges is a gateway between these sets. In particular $(A \setminus B)$ is such a gateway. Taking into account that sets B and $(A \setminus B)$ are disjoint, by the Gateway axiom, $G \vdash [B, (A \setminus B)] \to ([\varnothing] \to [B])$. By the Small Set axiom, $G \vdash [\varnothing]$. Thus, $G \vdash [B, (A \setminus B)] \to [B]$. By the assumption $B \subseteq A$, we conclude that $G \vdash [A] \to [B]$. □

Next we give two examples of derivations in our logical system. In these examples, by G_0 we mean the graph depicted earlier in Figure 1.

Proposition 3. $G_0 \vdash [c, b, f, e] \to [a, g]$.

Proof. We will start with graph G'_0 depicted in Figure 2. Recall that this graph is obtained from G_0 by truncation with crossing edges c, b, f and e. Note that, in graph G'_0, the empty set is a gateway between sets $\{a\}$ and $\{g\}$. Thus, by the Gateway axiom, $G'_0 \vdash [a] \to ([g] \to [a, g])$. By the Small Set axiom, $G'_0 \vdash [a]$ and $G'_0 \vdash [g]$. Hence, $G'_0 \vdash [a, g]$. By the Truncation rule, $G_0 \vdash [c, b, f, e] \to [a, g]$. □

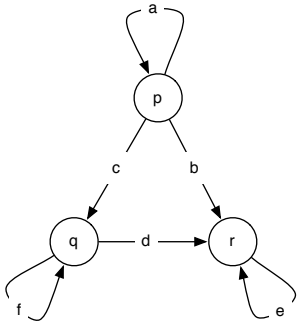


Fig. 3. Graph G''_0.

Proposition 4. $G_0 \vdash [c, b, d] \to [c, e]$.

Proof. Consider graph G''_0 depicted in Figure 3. It is obtained from graph G by a truncation with crossing edges e and f. Note that in graph G''_0 set $\{b, d\}$ is a gateway between sets $\{c\}$ and $\{e\}$. Thus, by the Gateway axiom, $G''_0 \vdash [c, b, d] \to ([e] \to [c, e])$. By the Small Set axiom, $G''_0 \vdash [e]$. Hence, $G''_0 \vdash [c, b, d] \to [c, e]$. By the Directed Truncation rule, $G_0 \vdash [c, b, d] \to [c, e]$. □

6 Soundness

The proof of soundness is non-trivial. For each axiom and inference rule, we provide its justification as a separate theorem.

Theorem 2 (Small Set). *For any graph $G = \langle V, E \rangle$, if $\mathcal{P}$ is an arbitrary protocol over G and subset $A \subseteq E$ has at most one element, then $\mathcal{P} \vDash [A]$.*

Proof. Case 1: $A = \varnothing$. Due to the continuity condition in Definition 1 and because graph G is acyclic, there is at least one run $r \in \mathcal{R}(\mathcal{P})$. Thus, $\mathcal{P} \vDash [\varnothing]$. *Case 2:* $A = \{a_1\}$. Consider any run $r_1 \in \mathcal{R}(\mathcal{P})$. Pick r to be r_1. This guarantees that $r(a_1) = r_1(a_1)$. □

Theorem 3 (Gateway). *For any graph $G = \langle V, E\rangle$, and any gateway W between sets of edges A and B in graph G, if $\mathcal{P} \vDash [A, W]$, $\mathcal{P} \vDash [B]$, and $A \cap W = \varnothing$, then $\mathcal{P} \vDash [A, B]$.*

Proof. Assume $\mathcal{P} \vDash [A, W]$, $\mathcal{P} \vDash [B]$, and $A \cap W = \varnothing$. Let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_k\}$. Consider any $r_1, \ldots, r_{n+k}$. We will show that there is a run $r \in \mathcal{R}(\mathcal{P})$ such that $r(a_i) = r_i(a_i)$ for each $i \leq n$ and $r(b_i) = r_{n+i}(b_i)$ for each $i \leq k$. By the assumption $\mathcal{P} \vDash [B]$, there is a run $r_B \in \mathcal{R}(\mathcal{P})$ such that

$$r_B(b_i) = r_{n+i}(b_i) \qquad \text{for } i \leq k. \tag{2}$$

By assumptions $\mathcal{P} \vDash [A, W]$ and $A \cap W = \varnothing$, there must be a run r_A such that

$$r_A(e) = \begin{cases} r_i(e) & \text{if } e = a_i \text{ for } i \leq n, \\ r_B(e) & \text{if } e \in W. \end{cases} \tag{3}$$

Next, consider graph G' obtained from G by removing all edges in W. By the definition of gateway, no single connected component of graph G' can contain both an edge from A and an edge from $(B \setminus W)$. Let us group all connected components of G' into two subgraphs G'_A and G'_B such that G'_A contains no edges from $(B \setminus W)$ and G'_B contains no edges from A. Components that contain edges neither from A nor from $(B \setminus W)$ can be arbitrarily assigned to either G'_A or G'_B.

By equation (3), runs r_A and r_B on G agree on each edge of gateway W. We will now construct a combined run r by "sewing together" portions of r_A and r_B with the "stitches" placed along gateway W. Formally,

$$r(e) = \begin{cases} r_A(e) & \text{if } e \in G'_A, \\ r_A(e) = r_B(e) & \text{if } e \in W, \\ r_B(e) & \text{if } e \in G'_B. \end{cases} \tag{4}$$

Let us first prove that r is a valid run of the protocol $\mathcal{P}$. For this, we need to prove that it satisfies action relation Δ_v at every vertex v. Without loss of generality, assume that $v \in G'_A$. Hence, on all edges incident with v, run r agrees with run r_A. Thus, run r satisfies Δ_v simply because r_A does.

Next, we will show that $r(a_i) = r_i(a_i)$ for each $i \leq n$. Indeed, by equations (3) and (4), $r(a_i) = r_A(a_i) = r_i(a_i)$. Finally, we need to show that $r(b_i) = r_{n+i}(b_i)$ for each $i \leq k$. This, however, follows easily from equations (2) and (4). □

Theorem 4 (Truncation). *Assume that C is the set of all crossing edges of cut (X, Y) in graph G and ϕ is a formula in $\Phi(G_X)$. If $\mathcal{P}' \vDash \phi$ for each protocol $\mathcal{P}'$ over G_X, then $\mathcal{P} \vDash [C] \to \phi$ for each protocol $\mathcal{P}$ over graph G.*

Proof. Suppose that there is a protocol $\mathcal{P}$ over G such that $\mathcal{P} \vDash [C]$, but $\mathcal{P} \nvDash \phi$. We will construct a protocol $\mathcal{P}'$ over G_X such that $\mathcal{P}' \nvDash \phi$.

Let $\mathcal{P} = \langle M, \Delta \rangle$. Note that, for any edge e, not all values from $M(e)$ are necessarily used in the runs of this protocol. Some values might be excluded by the action relations of $\mathcal{P}$. To construct protocol $\mathcal{P}' = \langle M', \Delta' \rangle$ over truncation G_X, for any edge e of G_X we first define $M'(e)$ as the set of values that are actually used by at least one run of protocol $\mathcal{P}$. Thus, $M'(e) = \{r(e) \mid r \in \mathcal{R}(\mathcal{P})\}$. The action relation Δ'_v at any vertex v of G_X is the same as under protocol $\mathcal{P}$.

Lemma 1. *For any run $r' \in \mathcal{R}(\mathcal{P}')$ there is a run $r \in \mathcal{R}(\mathcal{P})$ such that $r(e) = r'(e)$ for each edge e in truncation G_X.*

Proof. Consider any run $r' \in \mathcal{R}(\mathcal{P}')$. By the definition of M', for any crossing edge $c \in C$, there is a run $r_c \in \mathcal{R}(\mathcal{P})$ such that $r'(c) = r_c(c)$. Since $\mathcal{P} \vDash [C]$, there is a run $r_Y \in \mathcal{R}(\mathcal{P})$ such that $r_Y(c) = r_c(c) = r'(c)$ for each $c \in C$.

We will now construct a combined run $r \in \mathcal{R}(\mathcal{P})$ by "sewing together" r_Y and r' with the "stitches" placed in set C. Recall that we use the notation $E(X)$ to denote edges whose ends are both in set X. Formally, let

$$r(e) = \begin{cases} r'(e) & \text{if } e \in E(X), \\ r'(e) = r_Y(e) & \text{if } e \in C, \\ r_Y(e) & \text{if } e \in E(Y). \end{cases}$$

We just need to show that r satisfies Δ_v at every vertex v of graph G. Indeed, if $v \in Y$, then run r is equal to r_Y on all edges incident with v. Thus, it satisfies the action relation at v because run r_Y does. Alternatively, if $v \in X$, then run r is equal to run r' on all edges incident with v. Since r' satisfies action relation Δ'_v and, by definition, $\Delta'_v \equiv \Delta_v$ for all $v \in X$, we can conclude that r again satisfies condition Δ_v. □

Lemma 2. *For any set of edges Q in graph G_X, $\mathcal{P} \vDash [Q]$ if and only if $\mathcal{P}' \vDash [Q]$.*

Proof. Assume first that $\mathcal{P} \vDash [Q]$ and consider any runs $\{r'_q\}_{q \in Q} \subseteq \mathcal{R}(\mathcal{P}')$. We will construct a run $r' \in \mathcal{R}(\mathcal{P}')$ such that $r'(q) = r'_q(q)$ for every $q \in Q$. Indeed, by Lemma 1, there are runs $\{r_q\}_{q \in Q} \subseteq \mathcal{R}(\mathcal{P})$ that match runs $\{r'_q\}_{q \in Q}$ on all edges in G_X. By the assumption that $\mathcal{P} \vDash [Q]$, there must be a run $r \in \mathcal{R}(\mathcal{P})$ such that $r(q) = r_q(q)$ for all $q \in Q$. Hence, $r(q) = r_q(q) = r'_q(q)$ for all $q \in Q$. Let r' be the restriction of run r to the edges in G_X. Since the action relations of protocols $\mathcal{P}$ and $\mathcal{P}'$ are the same at all vertices in X, we can conclude that $r' \in \mathcal{R}(\mathcal{P}')$. Finally, we notice that $r'(q) = r(q) = r'_q(q)$ for any $q \in Q$.

Next, assume that $\mathcal{P}' \vDash [Q]$ and consider any runs $\{r_q\}_{q \in Q} \subseteq \mathcal{R}(\mathcal{P})$. We will show that there is a run $r \in \mathcal{R}(\mathcal{P})$ such that $r(q) = r_q(q)$ for all $q \in Q$. Indeed, let $\{r'_q\}_{q \in Q}$ be the restrictions of runs $\{r_q\}_{q \in Q}$ to the edges in G_X. Since the action relations of these two protocols are the same at the vertices in X, we can conclude that $\{r'_q\}_{q \in Q} \subseteq \mathcal{R}(\mathcal{P}')$. By the assumption that $\mathcal{P}' \vDash [Q]$, there is a run $r' \in \mathcal{R}(\mathcal{P}')$ such that $r'(q) = r'_q(q) = r_q(q)$ for all $q \in Q$. By Lemma 1, there is a run $r \in \mathcal{R}(\mathcal{P})$ that matches r' everywhere in G_X. Therefore, $r(q) = r'(q) = r_q(q)$ for all $q \in Q$. □

Lemma 3. *For any formula $\psi \in \Phi(G_X)$, $\mathcal{P} \vDash \psi$ if and only if $\mathcal{P}' \vDash \psi$.*

Proof. We use induction on the complexity of ψ. The base case follows from Lemma 2, and the induction step is trivial. □

The statement of Theorem 4 immediately follows from Lemma 3. □

Theorem 5 (Directed Truncation). *Assume that (X, Y) is a directed cut of a graph G and ϕ is a formula in $\Phi(G_X)$. If $\mathcal{P}' \vDash \phi$ for every protocol $\mathcal{P}'$ over truncation G_X, then $\mathcal{P} \vDash \phi$ for every protocol $\mathcal{P}$ over graph G.*

The proof of this theorem is a straightforward modification of the proof of Theorem 4. Specifically, in the proof of Lemma 1, instead of "sewing together" runs r' and r_Y, we use the continuity condition from Definition 1 to extend run $r' \in \mathcal{R}(\mathcal{P}')$ into a run $r \in \mathcal{R}(\mathcal{P})$ that agrees with r' on all vertices in G_X.

7 Completeness

Theorem 6 (completeness). *For any directed graph G, if $\mathcal{P} \vDash \phi$ for all finite protocols $\mathcal{P}$ over G, then $G \vdash \phi$.*

The theorem will be proven by contrapositive. At the core of this proof is the construction of a finite protocol. This protocol will be formed as a composition of several simpler protocols, where each of the simpler protocols is defined recursively. The base case of this recursive definition is the parity protocol defined below. It is a generalization of the protocol described in the introduction.

7.1 Parity Protocol

In the following discussion, we use the overloaded notation $Inc(x)$ to denote the set of objects incident with an object x in a graph, where x may be either an edge or a vertex. That is, if x is an edge, then $Inc(x)$ represents the set of (at most two) vertices which are the ends of edge x. On the other hand, if x is a vertex, then $Inc(x)$ represents the set of edges which have vertex x as an end.

Let $G = \langle V, E \rangle$ be a graph and A be a subset of E. We define the "parity protocol" $\mathcal{P}_A$ over G as follows. The set of values of any edge e in graph G is the set of boolean functions on the ends of e (each loop edge is assumed to have a single end). Thus, a run r of the protocol will be a function that maps an edge into a function from the ends of this edge into boolean values: $r(e)(v) \in \{0, 1\}$, where e is an edge and v is an end of e. It will be more convenient, however, to think about a run as a two-argument function $r(e, v) \in \{0, 1\}$.

Not all assignments of boolean values to the ends of an edge e will be permitted in the parity protocol. Namely, if $e \notin A$, then the sum of all values assigned to the ends of e must be even. This is formally captured by the following condition:

$$\sum_{v \in Inc(e)} r(e, v) \equiv 0 \pmod 2. \tag{5}$$

This means that if an edge $e \notin A$ has two ends, then the values assigned to its two ends must be equal. If edge $e \notin A$ is a loop edge and, thus, has only one end, then the value assigned to this end must be 0. However, if $e \in A$, then no restriction on the assignment of boolean values to the ends of e will be imposed. This defines the set of values $M(e)$ for each edge e under $\mathcal{P}_A$.

The second restriction on the runs will require that the sum of all values assigned to ends incident with any vertex v is also even:

$$\sum_{e \in Inc(v)} r(e, v) \equiv 0 \pmod 2. \tag{6}$$

The latter restriction specifies the action relation Δ_v for each vertex v. We will graphically represent a run by placing boolean values at each end of each edge of the graph. For example, Figure 4 depicts a possible run of the parity protocol $\mathcal{P}_A$ with $A = \{c, b, g\}$ over the graph G_0 from Figure 1.

The finite protocol $\mathcal{P}_A$ is now completely defined, but we still need to prove that it satisfies the continuity condition from Definition 1. This is true, however, only under an additional assumption:

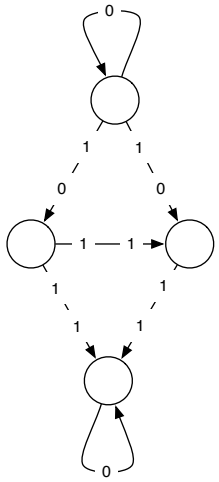


Fig. 4. A run.

Lemma 4. *If set A is such that it contains a loop edge for each sink of graph G, then $\mathcal{P}_A$ satisfies the continuity condition.*

Proof. As long as a vertex has at least one outgoing edge whose boolean value is not fixed, this value an be adjusted to satisfy condition (6). The only edges that have fixed values are loop edges that do not belong to set A. □

Recall that we use the notation $Inc(x)$ to denote the set of objects incident with either an edge x or a vertex x.

Lemma 5. $\sum_{e \in A} \sum_{v \in Inc(e)} r(e, v) \equiv 0 \pmod 2$, *for any run r of the parity protocol $\mathcal{P}_A$.*

Proof. Let $G = \langle V, E \rangle$. Using equations (6) and (5),

$$\sum_{e \in A} \sum_{v \in Inc(e)} r(e, v) \equiv \sum_{e \in E} \sum_{v \in Inc(e)} r(e, v) - \sum_{e \in E \setminus A} \sum_{v \in Inc(e)} r(e, v) \equiv$$
$$\equiv \sum_{v \in V} \sum_{e \in Inc(v)} r(e, v) - \sum_{e \notin A} 0 \equiv \sum_{v \in V} 0 - 0 \equiv 0 \pmod 2.$$

Everywhere below, by a path we will mean a sequence of edges that form a simple (undirected) path.

Definition 9. *For any path $\pi = e_0, e_1, \ldots, e_n$ in a graph G and any run r of the parity protocol $\mathcal{P}_A$, we define run r_π as*

$$r_\pi(e, v) = \begin{cases} 1 - r(e, v) & \text{if } v \in Inc(e_i) \cap Inc(e_{i+1}) \text{ for some } i < n, \\ r(e, v) & \text{otherwise.} \end{cases}$$

Informally, r_π is obtained from r by "flipping" the boolean values on path π at π's "internal" vertices. If a path is cyclic, then all vertices along this path are considered to be internal.

Lemma 6. *For any $r \in \mathcal{P}_A$ and any path π, if π is a cycle or starts and ends with edges that belong to set A, then $r_\pi \in \mathcal{R}(\mathcal{P}_A)$.*

Proof. Run r_π satisfies condition (5) because r_π is different from r at both ends of any non-terminal edge of path π. The same run r_π satisfies condition (6) at every vertex v of the graph, because path π includes either zero or two ends of edges incident at vertex v. □

Lemma 7. *If $|A| > 1$ and graph G is connected, then for any $e \in A$ and any $g \in \{0, 1\}$ there is a run $r \in \mathcal{R}(\mathcal{P}_A)$ such that $\sum_{v \in Inc(e)} r(e, v) \equiv g \pmod 2$.*

Proof. Let $\hat{r}(e, v)$ be a run of the protocol $\mathcal{P}_A$ which is equal to 0 for each end v of each edge e. If $g = 0$, then $\hat{r}$ is the required run r. Assume now that $g = 1$. Since $|A| > 1$ and graph G is connected, there is a path π that connects edge e with an edge $a \in A$ such that $a \neq e$. Notice that $\hat{r}_\pi$ is the desired run r, since $\sum_{v \in Inc(e)} \hat{r}_\pi(e, v) = \sum_{v \in Inc(e)} \hat{r}(e, v) + 1 \equiv g \pmod 2$. □

Lemma 8. *If $|A| > 1$ and graph G is connected, then $\mathcal{P}_A \not\models [A]$.*

Proof. Let $A = \{a_1, \ldots, a_k\}$. Pick any boolean values $g_1, \ldots, g_k$ such that $g_1 + \cdots + g_k \equiv 1 \pmod 2$. By Lemma 7, there are runs $r_1, \ldots, r_k \in \mathcal{R}(\mathcal{P}_A)$ such that $\sum_{v \in a_i} r_i(a_i, v) \equiv g_i \pmod 2$ for any $i \leq k$. If $\mathcal{P}_A \models [A]$, then there is a run $r \in \mathcal{R}(\mathcal{P}_A)$ such that $r(a_i, v) = r_i(a_i, v)$ for each $v \in a_i$ and each $i \leq k$. Therefore, $\sum_{v \in a_1} r(a_1, v) + \cdots + \sum_{v \in a_k} r(a_k, v) = \sum_{v \in a_1} r_1(a_1, v) + \cdots + \sum_{v \in a_k} r_k(a_k, v) \equiv g_1 + \cdots + g_k \equiv 1 \pmod 2$. This contradicts Lemma 5. □

Lemma 9. *If A and B are sets of edges of a graph $G = \langle V, E \rangle$, such that each connected component of the graph $\langle V, E \setminus B \rangle$ contains at least one edge from A, then $\mathcal{P}_A \models [B]$.*

Proof. Let $B = \{b_1, \ldots, b_k\}$. Consider any runs $r_1, \ldots, r_k \in \mathcal{R}(\mathcal{P}_A)$. We will prove that there is a run $r \in \mathcal{R}(\mathcal{P}_A)$ such that $r(b_i, v) = r_i(b_i, v)$ for any $v \in Inc(b_i)$ and any $i \leq k$. We will start with a run $\hat{r}(e, v)$ equal to 0 for each end v of each edge e and modify it to satisfy the condition $\hat{r}(b_i, v) = r_i(b_i, v)$ for every $i \leq k$ and every $v \in Inc(b_i)$. Our modification will consist of repeating the following procedure for each $i \leq k$ and each $v \in Inc(b_i)$ such that $\hat{r}(b_i, v) \neq r_i(b_i, v)$:

1. If $b_i \in A$, then, by the assumption of the lemma, there must be a path a_0,e_1,e_2,$\ldots, e_n$ in the graph $\langle V, E \setminus B \rangle$ that connects an edge $a_0 \in A$ with vertex v. Consider path $\pi = a_0$,e_1,e_2, $\ldots$, e_n, b_i in graph G. By Lemma 6, $\hat{r}_\pi \in \mathcal{R}(\mathcal{P}_A)$. Note that $\hat{r}_\pi$ matches $\hat{r}$ exactly on both ends of each edge b_j, where $j \neq i$. Furthermore, if b_i is not a loop edge, then $\hat{r}_\pi$ also matches $\hat{r}$ exactly on the end of edge b_i which is not incident with vertex v. However, $\hat{r}_\pi(b_i, v) = 1 - \hat{r}(b_i, v) = r_i(b_i, v)$, as desired. Pick $\hat{r}_\pi$ to be the new $\hat{r}$.

2. If $b_i \notin A$, then, by (5), $\sum_{v \in Inc(b_i)} \hat{r}(b_i, v) \equiv 0 \equiv \sum_{v \in Inc(b_i)} r_i(b_i, v) \pmod 2$. At the same time, by our assumption, $\hat{r}(b_i, v) \neq r_i(b_i, v)$. Thus another end $u \in Inc(b_i)$ must exist and be such that $u \neq v$ and $\hat{r}(b_i, u) \neq r_i(b_i, u)$. Note that vertices u and v may belong to either the same connected component or to two different connected components of graph $\langle V, E \setminus B \rangle$. We will consider these two subcases separately.
   (a) Suppose u and v belong to the same connected component of graph $\langle V, E \setminus B \rangle$. Thus, there must be a path π' in that graph which connects an edge containing vertex u with an edge containing v. Now, consider a cyclic path in graph $G = \langle V, E \rangle$ that starts at edge b_i, via vertex u connects to path π', goes through the whole path π', and via vertex v connects back to b_i. Call this cyclic path π.
   (b) Suppose u and v belong to different connected components of graph $\langle V, E \setminus B \rangle$. Thus, by the assumption of the lemma, graph $\langle V, E \setminus B \rangle$ contains a path $\pi_u = a_u, \ldots, e_u$ that connects an edge $a_u \in A$ with an edge e_u containing end u. By the same assumption, graph $\langle V, E \setminus B \rangle$ must also contain a path $\pi_v = e_v, \ldots, a_v$ that connects an edge e_v, containing end v, with an edge $a_v \in A$. Let $\pi = \pi_u, b_i, \pi_v$.

   Note that $\hat{r}_\pi$ matches $\hat{r}$ exactly on all ends of each edge b_j where $j \neq i$. However, $\hat{r}_\pi(b_i, v) = 1 - \hat{r}(b_i, v) = r_i(b_i, v)$, as desired. In addition, $\hat{r}_\pi(b_i, u) = 1 - \hat{r}(b_i, u) = r_i(b_i, u)$. Furthermore, by Lemma 6, $\hat{r}_\pi \in \mathcal{R}(\mathcal{P}_A)$. Pick $\hat{r}_\pi$ to be the new $\hat{r}$.

Let r be $\hat{r}$ with all the modifications described above. These modifications guarantee that $r(b_i, v) = \hat{r}(b_i, v) = r_i(b_i, v)$ for each $i \leq k$ and each $v \in b_i$. □

7.2 Recursive Construction

In this section we will generalize the parity protocol through a recursive construction. First, however, we will establish a technical result that we will need for this construction.

Lemma 10 (protocol extension). *For any cut (X, Y) of graph $G = \langle V, E \rangle$ and any finite protocol $\mathcal{P}'$ on truncation G_X, there is a finite protocol $\mathcal{P}$ on G such that for any set $Q \subseteq E$, $\mathcal{P} \vDash [Q]$ if and only if $\mathcal{P}' \vDash [Q \cap E(G_X)]$.*

Proof. To define protocol $\mathcal{P}$ we need to specify a set of values $M(e)$ for each edge $e \in E$ and the set of action relations for each vertex p in graph G. If $e \in E(G_X)$, then let $M(e)$ be the same as in protocol $\mathcal{P}$. Otherwise, $M(e) = \{\epsilon\}$, where ϵ is an arbitrary element. The action relations at the vertices in X are as in protocol $\mathcal{P}'$, and the action relations at the vertices in Y are equal to the boolean constant $True$. It is easy to see that because the continuity condition in Definition 1 holds for $\mathcal{P}'$, it also holds for $\mathcal{P}$. This completes the definition of $\mathcal{P}$.
($\Rightarrow$) : Suppose that $Q \cap E(G_X) = \{q_1, \ldots, q_k\}$. Consider any $r'_1, \ldots, r'_k \in \mathcal{R}(\mathcal{P}')$. Define runs $r_1, \ldots, r_k$ as follows. For any edge e:

$$r_i(e) = \begin{cases} r'_i(e) & \text{if } e \in E(G_X), \\ \varepsilon & \text{if } e \notin E(G_X). \end{cases}$$

Note that runs r_i and r'_i, by definition, are equal on any edge incident with any vertex in graph G_X. Thus, r_i satisfies the action relations at any such vertex. Hence, since the action relations at all other vertices are trivially satisfied, $r_i \in \mathcal{R}(\mathcal{P})$ for each $i \in \{1, \ldots, k\}$. By the continuity condition in Definition 1 and the fact that G is acyclic, there must be at least one run of protocol $\mathcal{P}$ (even if $k = 0$). Call this run r_0. By the assumption that $\mathcal{P} \vDash [Q]$, there is a run $r \in \mathcal{R}(\mathcal{P})$ such that for any edge e,

$$r(e) = \begin{cases} r_i(e) \text{ if } e = q_i, \\ r_0(e) \text{ if } e \in Q \setminus E(G_X). \end{cases}$$

Define r' to be a restriction of r on graph G_X. Note that r' satisfies all action relations of $\mathcal{P}'$. Thus, $r' \in \mathcal{R}(\mathcal{P}')$. At the same time, $r'(q_i) = r_i(q_i) = r'_i(q_i)$.
($\Leftarrow$) : Suppose that $Q = \{q_1, \ldots, q_k\}$. Consider any runs $r_1, \ldots, r_k \in \mathcal{R}(\mathcal{P})$, and let $r'_1, \ldots, r'_k$ be their respective restrictions to graph G_X. Since, for any $i \in \{1, \ldots, k\}$, run r'_i satisfies the action relations of $\mathcal{P}'$ at any vertex of G_X, we can conclude that $r'_1, \ldots, r'_k \in \mathcal{R}(\mathcal{P}')$. By the assumption that $\mathcal{P}' \vDash [Q \cap E(G_X)]$, there is a run $r' \in \mathcal{R}(\mathcal{P}')$ such that $r'(q_i) = r'_i(q_i)$ for any i such that $q_i \in Q \cap E(G_X)$. In addition, $r'(q) = \varepsilon = r'_i(q)$ for any $q \in Q \setminus E(G_X)$. Hence, $r'(q_i) = r'_i(q_i)$ for any $i \in \{1, \ldots, k\}$. For any edge e, define run r as follows:

$$r(e) = \begin{cases} r'(e) \text{ if } e \in E(G_X), \\ \varepsilon \quad\;\; \text{if } e \notin E(G_X). \end{cases}$$

Note that r satisfies the action relations of $\mathcal{P}$ at all vertices. Thus, $r \in \mathcal{R}(\mathcal{P})$. In addition, $r(q_i) = r'(q_i) = r'_i(q_i)$ for all $i \in \{1, \ldots, k\}$. □

We will now prove another key lemma in our construction. The proof of this lemma recursively defines a generalization of the parity protocol.

Lemma 11. *For any sets $A, B_1, \ldots, B_n$ of edges of G, if $G \nvdash \bigwedge_{1 \le i \le n} [B_i] \to [A]$, then there is a finite protocol $\mathcal{P}$ over G such that $\mathcal{P} \vDash [B_i]$ for all $1 \le i \le n$ and $\mathcal{P} \nvDash [A]$.*

Proof. We use induction on the number of vertices of graph G.
Case 1. If $|A| \le 1$, then, by the Small Set axiom, $G \vdash [A]$. Hence, $G \vdash \bigwedge_{1 \le i \le n} [B_i] \to [A]$, which is a contradiction.
Case 2. Suppose that the edges of graph G can be partitioned into two non-trivial disconnected sets X and Y. That is, no edge in X is adjacent with a edge in Y. Thus, the empty set is a gateway between $A \cap X$ and $A \cap Y$. By the Gateway axiom, $G \vdash [A \cap X] \to ([A \cap Y] \to [A])$. Hence, taking into account the assumption $G \nvdash \bigwedge_{1 \le i \le n} [B_i] \to [A]$, either $G \nvdash \bigwedge_{1 \le i \le n} [B_i] \to [A \cap X]$ or $G \nvdash \bigwedge_{1 \le i \le n} [B_i] \to [A \cap Y]$. Without loss of generality, we will assume the former. By Theorem 1, $G \nvdash \bigwedge_{1 \le i \le n} [B_i \cap X] \to [A \cap X]$. Consider the sets P_X and P_Y of all vertices in components X and Y respectively. Note that (P_X, P_Y) is a cut of G that has no crossing edges. Let G_X be the result of the truncation of G along this cut. By the Directed Truncation rule, $G_X \nvdash \bigwedge_{1 \le i \le n} [B_i \cap X] \to [A \cap X]$. By the Induction Hypothesis, there is a protocol $\mathcal{P}'$ on G_X such that $\mathcal{P}' \nvDash [A \cap X]$

and $\mathcal{P}' \vDash [B_i \cap X]$, for any $i \leq n$. Therefore, by Lemma 10, there is a protocol $\mathcal{P}$ on G such that $\mathcal{P} \nvDash [A]$ and $\mathcal{P} \vDash [B_i]$ for any $i \leq n$.

Case 3. Suppose that graph G has a non-trivial directed cut (X, Y) such that $E(Y) \cap A = \varnothing$. Thus, by Theorem 1, $G \nvdash \bigwedge_{1 \leq i \leq n} [B_i \cap E(X)] \to [A]$. By the Directed Truncation rule, $G_X \nvdash \bigwedge_{1 \leq i \leq n} [B_i \cap E(X)] \to [A]$. By the Induction Hypothesis, there is a protocol $\mathcal{P}'$ over G_X such that $\mathcal{P}' \vDash [B_i \cap E(X)]$ for all $1 \leq i \leq n$ and $\mathcal{P}' \nvDash [A]$. Therefore, by Lemma 10, there is a protocol $\mathcal{P}$ on G such that $\mathcal{P} \nvDash [A]$ and $\mathcal{P} \vDash [B_i]$ for any $i \leq n$.

Case 4. Suppose there is $i_0 \leq n$ such that if all edges in B_{i_0} are removed from graph G, then at least one connected component of the resulting network G' does not contain an element of A. We will denote this connected component by Q. Let $W \subseteq B_{i_0}$ be the set of edges in G that connect a vertex from Q with a vertex not in Q. Any path connecting a edge in $E(Q)$ with a edge not in $E(Q)$ will have to contain a edge from W. In other words, W is a gateway between $E(Q)$ and the complement of $E(Q)$ in G. Hence, W is also a gateway between $A \cap E(Q)$ and $A \setminus E(Q)$. Therefore, by the Gateway axiom, taking into account that $(A \cap E(Q)) \cap W \subseteq E(Q) \cap W = \varnothing$,

$$G \vdash [A \cap E(Q), W] \to ([A \setminus E(Q)] \to [A]). \tag{7}$$

Recall now that by the assumption of this case, component Q of graph G' does not contain any elements of A. Hence, $A \cap E(Q) \subseteq B_{i_0}$. At the same time, $W \subseteq B_{i_0}$. Thus, from statement (7) and Theorem 1,

$$G \vdash [B_{i_0}] \to ([A \setminus E(Q))] \to [A]). \tag{8}$$

By the assumption of the lemma,

$$G \nvdash \bigwedge_{1 \leq i \leq n} [B_i] \to [A]. \tag{9}$$

From statements (8) and (9), $G \nvdash \bigwedge_{1 \leq i \leq n} [B_i] \to [A \setminus E(Q))]$. By the laws of propositional logic, $G \nvdash [B_{i_0}] \to (\bigwedge_{1 \leq i \leq n} [B_i] \to [A \setminus E(Q)])$. Note that if $\overline{Q}$ is the complement of set Q, then $(\overline{Q}, Q)$ is a cut of graph G and W is the set of all crossing edges of this cut. Since $W \subseteq B_{i_0}$, by Theorem 1, $G \nvdash [W] \to (\bigwedge_{1 \leq i \leq n} [B_i] \to [A \setminus E(Q)])$. Again by Theorem 1, $G \nvdash [W] \to (\bigwedge_{1 \leq i \leq n} [B_i \setminus E(Q)] \to [A \setminus E(Q)])$. Let $G_{\overline{Q}}$ be the truncation of graph G along the cut $(\overline{Q}, Q)$. By the Truncation rule, $G_{\overline{Q}} \nvdash \bigwedge_{1 \leq i \leq n} [B_i \setminus E(Q)] \to [A \setminus E(Q)]$.

By the Induction Hypothesis, there is a protocol $\mathcal{P}'$ on $G_{\overline{Q}}$ such that $\mathcal{P}' \nvDash [A \setminus E(Q)]$ and $\mathcal{P}' \vDash [B_i \setminus E(Q)]$ for any $i \leq n$. Therefore, by Lemma 10, there is a protocol $\mathcal{P}$ on G such that $\mathcal{P} \nvDash [A]$ and $\mathcal{P} \vDash [B_i]$ for any $i \leq n$.

Case 5. Assume now that (i) $|A| > 1$, (ii) graph G is connected, (iii) graph G has no non-trivial directed cuts (X, Y) such that $E(Y) \cap A = \varnothing$, and (iv) for any $i \leq n$, if graph G' is obtained from G by the removal of all edges in B_i then each connected component of G' contains at least one element of A. Note that condition (iii) implies that A contains at least one loop edge at every sink vertex in graph G. Consider the parity protocol $\mathcal{P}_A$ over G. By Lemma 8, $\mathcal{P}_A \nvDash [A]$. By Lemma 9, $\mathcal{P}_A \vDash [B_i]$ for any $i \leq n$. □

7.3 Protocol Composition

In this section, we define a composition of several protocols and finish the proof of the completeness theorem.

Definition 10. *For any protocols $\mathcal{P}^1 = (M^1, \Delta^1), \ldots, \mathcal{P}^n = (M^n, \Delta^n)$ over a graph G, we define the Cartesian composition $\mathcal{P}^1 \times \mathcal{P}^2 \times \cdots \times \mathcal{P}^n$ to be a pair (M, Δ) such that*

1. $M(e) = M^1(e) \times \cdots \times M^n(e)$,
2. $\Delta_p(\langle e_1^1, \ldots, e_1^n \rangle, \ldots, \langle e_k^1, \ldots, e_k^n \rangle) = \bigwedge_{1 \le i \le n} \Delta_p^i(e_1^i, \ldots, e_k^i)$.

For each composition $\mathcal{P} = \mathcal{P}^1 \times \mathcal{P}^2 \times \cdots \times \mathcal{P}^n$, let $\{r(e)\}_i$ denote the ith component of the value of secret e over run r.

Lemma 12. *For any $n > 0$ and any finite protocols $\mathcal{P}^1, \ldots, \mathcal{P}^n$ over a graph G, $\mathcal{P} = \mathcal{P}^1 \times \mathcal{P}^2 \times \cdots \times \mathcal{P}^n$ is a finite protocol over G.*

Proof. The validity of the continuity condition for $\mathcal{P}$ follows from the continuity conditions for protocols $\mathcal{P}^1, \ldots, \mathcal{P}^n$. □

Lemma 13. *For any $n > 0$, for any protocol $\mathcal{P} = \mathcal{P}^1 \times \mathcal{P}^2 \times \cdots \times \mathcal{P}^n$ over a graph $G = \langle V, E \rangle$, and for any set of edges Q, $\mathcal{P} \vDash [Q]$ if and only if $\forall i$ $(\mathcal{P}^i \vDash [Q])$.*

Proof. Let $Q = \{q_1, \ldots, q_\ell\}$.
($\Rightarrow$) : Assume $\mathcal{P} \vDash [Q]$ and pick any $i_0 \in \{1, \ldots, n\}$. We will show that $\mathcal{P}^{i_0} \vDash [Q]$. Pick any runs $r'_1, \ldots, r'_\ell \in \mathcal{R}(\mathcal{P}^{i_0})$. For each $i \in \{1, \ldots, i_0 - 1, i_0 + 1, \ldots, n\}$, select an arbitrary run $r^i \in \mathcal{R}(\mathcal{P}^i)$. Such runs exist because graph G is acyclic and all protocols satisfy the continuity condition. We then define a series of composed runs r_j for $j \in \{1, \ldots, \ell\}$ by

$$r_j(e) = \langle r^1(e), \ldots, r^{i_0 - 1}(e), r'_j(e), r^{i_0 + 1}(e), \ldots, r^n(e) \rangle,$$

for each edge $e \in E$. Since the component parts of each r_j belong in their respective sets $\mathcal{R}(\mathcal{P}^i)$, the composed runs are themselves members of $\mathcal{R}(\mathcal{P})$. By our assumption, $\mathcal{P} \vDash [Q]$, thus there is $r \in \mathcal{R}(\mathcal{P})$ such that $r(q_i) = r_i(q_i)$ for any $i_0 \in \{1, \ldots, \ell\}$. Finally, we consider the run r^*, where $r^*(e) = \{r(e)\}_{i_0}$ for each $e \in E$. That is, we let the value of r^* on e be the i_o^{th} component of $r(e)$. By the definition of composition, $r^* \in \mathcal{R}(\mathcal{P}^{i_0})$, and it matches the original $r'_1, \ldots, r'_\ell \in \mathcal{R}(\mathcal{P}^{i_0})$ on edges $q_1, \ldots, q_\ell$, respectively. Hence, we have shown that $\mathcal{P}^{i_0} \vDash [Q]$.
($\Leftarrow$) : Assume $\forall i$ $(\mathcal{P}^i \vDash [Q])$. We will show that $\mathcal{P} \vDash [Q]$. Pick any runs $r_1, \ldots, r_\ell \in \mathcal{R}(\mathcal{P})$. For each $i \in \{1, \ldots, n\}$, each $j \in \{1, \ldots, \ell\}$, and each edge e, let $r_j^i(e) = \{r_j(e)\}_i$. That is, for each e, define a run r_j^i whose value on edge e equals the ith component of $r_j(e)$. Note that by the definition of composition, for each i and each j, r_j^i is a run in $\mathcal{R}(\mathcal{P}^i)$. Next, for each $i \in \{1, \ldots, n\}$, we use the fact that $\mathcal{P}^i \vDash [Q]$ to construct a run $r^i \in \mathcal{R}(\mathcal{P}^i)$ such that $r^i(q_j) = r_j^i(q_j)$. Finally, we compose these n runs $r^1, \ldots, r^n$ to get run $r \in \mathcal{R}(\mathcal{P})$. We note that the value of each edge q_j on r matches the the value of q_j in run $r_j \in \mathcal{R}(\mathcal{P})$, demonstrating that $\mathcal{P} \vDash [Q]$. □

We are now ready to prove the completeness theorem, which was stated earlier as Theorem 6:

Theorem 6. *For any graph $G = \langle V, E\rangle$, if $\mathcal{P} \models \phi$ for all finite protocols $\mathcal{P}$ over G, then $G \vdash \phi$.*

Proof. We give a proof by contradiction. Let X be a maximal consistent set of formulas from $\Phi(G)$ that contains $\neg\phi$. Let $\{A_1, \ldots, A_n\} = \{A \subseteq E \mid [A] \notin X\}$ and $\{B_1, \ldots, B_k\} = \{B \subseteq E \mid [B] \in X\}$. Thus, due to the maximality of set X, we have $G \nvdash \bigwedge_{1\leq j\leq k}[B_j] \to [A_i]$, for every $i \in \{1, \ldots, n\}$. We will construct a protocol $\mathcal{P}$ such that $\mathcal{P} \not\models [A_i]$ for any $i \in \{1, \ldots, n\}$ and $\mathcal{P} \models [B_j]$ for any $j \in \{1, \ldots, k\}$.

First consider the case where $n = 0$. Pick any symbol ϵ and define $\mathcal{P}$ to be $\langle M, \Delta\rangle$ such that $M(e) = \{\epsilon\}$ for any $e \in E$ and action relation Δ_p to be the constant $True$ at any vertex p. By Definition 4, $\mathcal{P} \models [C]$ for any $C \subseteq E$.

We will assume now that $n > 0$. By Theorem 11, there are finite protocols $\mathcal{P}^1, \ldots, \mathcal{P}^n$ such that $\mathcal{P}^i \not\models [A_i]$ and $\mathcal{P}^i \models [B_j]$ for all $j \in \{1, \ldots, k\}$. Consider the composition $\mathcal{P}$ of protocols $\mathcal{P}^1, \ldots, \mathcal{P}^n$. By Theorem 13, $\mathcal{P} \not\models [A_i]$ for any $i \in \{1, \ldots, n\}$ and $\mathcal{P} \models [B_j]$ for any $j \in \{1, \ldots, j\}$.

Since X is a maximal consistent set, by induction on the structural complexity of any formula $\psi \in \Phi(G)$, one can show now that $\psi \in X$ if and only if $\mathcal{P} \models \psi$. Thus, $\mathcal{P} \models \neg\phi$. Therefore, $\mathcal{P} \not\models \phi$, which is a contradiction. □

Corollary 1. *The set $\{(G, \phi) \mid G \vdash \phi\}$ is decidable.*

Proof. The complement of this set is recursively enumerable due to the completeness of the system with respect to finite protocols. □

8 Conclusion

In this paper, we captured the properties of information flow that can be described in terms of the independence relation $[A]$. This is not the only relation that can be used to describe properties of information flow on a graph. Another natural relation is the functional dependency relation $A \rhd B$ between two sets of edges. This relation is true if the values of edges in set A functionally determine the values of all edges in set B. A complete axiomatization of this relation when graph G is not fixed was given by Armstrong [5]. This logical system has become known in the database literature as Armstrong's axioms [6, p. 81]. Beeri, Fagin, and Howard [7] suggested a variation of Armstrong's axioms that describe properties of multi-valued dependency.

A complete axiomatization of relation $A \rhd B$ for a fixed *undirected* graph was given by More and Naumov [8]. It consists of Armstrong's axioms and a version of the Gateway axiom discussed in this paper, but contains no inference rules other than Modus Ponens. It appears, however, that this result can not be easily generalized to directed acyclic graphs. Thus, an axiomatization of relation $A \rhd B$ for directed acyclic graphs remains an open problem.

References

1. Sutherland, D.: A model of information. In: Proceedings of Ninth National Computer Security Conference. (1986) 175–183
2. Halpern, J.Y., O'Neill, K.R.: Secrecy in multiagent systems. ACM Trans. Inf. Syst. Secur. **12**(1) (2008) 1–47
3. Miner More, S., Naumov, P.: On interdependence of secrets in collaboration networks. In: Proceedings of 12th Conference on Theoretical Aspects of Rationality and Knowledge (Stanford University, 2009). (2009) 208–217
4. Miner More, S., Naumov, P.: Hypergraphs of multiparty secrets. In: 11th International Workshop on Computational Logic in Multi-Agent Systems CLIMA XI (Lisbon, Portugal), LNAI 6245, Springer (2010) 15–32
5. Armstrong, W.W.: Dependency structures of data base relationships. In: Information processing 74 (Proc. IFIP Congress, Stockholm, 1974). North-Holland, Amsterdam (1974) 580–583
6. Garcia-Molina, H., Ullman, J., Widom, J.: Database Systems: The Complete Book. Second edn. Prentice-Hall (2009)
7. Beeri, C., Fagin, R., Howard, J.H.: A complete axiomatization for functional and multivalued dependencies in database relations. In: SIGMOD '77: Proceedings of the 1977 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM (1977) 47–61
8. More, S.M., Naumov, P.: Functional dependence of secrets in a collaboration network. CoRR **arXiv:1011.0399v1 [cs.LO]** (2010)