Information and Computation **166**, 35–60 (2001)
doi:10.1006/inco.2000.3015, available online at http://www.idealibrary.com on IDEAL®

On Fairness Notions in Distributed Systems

II. Equivalence-Completions and Their Hierarchies*

Yuh-Jzer Joung†

Department of Information Management, National Taiwan University, Taipei 106, Taiwan
E-mail: joung@ccms.ntu.edu.tw

Received October 4, 1996

This is the second part of a two-part paper in which we discuss the implementability of *fairness notions* in distributed systems where asynchronous processes interact via multiparty interactions. We focus here on equivalence-robust fairness notions where equivalence computations are either all fair or all unfair. Francez *et al.* (1992, *Formal Aspects Comput.* **4**, 582–591) propose a notion of *completion* to transform a non-equivalence-robust fairness notion to an equivalence-robust one while maintaining several properties of the source. However, a completion may not preserve *strong feasibility*—a necessary and sufficient condition for a completion to be implementable. In this paper, we study the system requirement for a completion to be strongly feasible and determine the strongest implementable completion for every given fairness notion. Moreover, for most systems we obtain a fairness notion, which we refer to as SG^+, such that SG^+ is the strongest fairness notion that is both implementable and equivalence-robust. We also provide a comprehensive comparison of SG^+ and several well-known fairness notions and their minimal and maximal completions. Finally, we show that if equivalence-robustness is dropped, then in general it is impossible to define a fairness notion that is implementable and stronger than all other implementable fairness notions, unless the system consists of only one interaction. This implies plenty of leeway in the design of fairness notions suitable for various applications.

INTRODUCTION

This is the second part of a two-part paper in which we discuss the implementability of *fairness notions* in distributed systems where asynchronous processes interact via multiparty interactions. In Part I [5] we have presented a necessary and sufficient criterion for determining the implementability of fairness notions. We focus here on equivalence-robust fairness notions where equivalent computations are either all fair or all unfair.

1. EQUIVALENCE-ROBUSTNESS AND COMPLETIONS

Intuitively, equivalence-robustness ensures that different observations of the same partial-order computation obtain the same property of the system [6]. It thus serves as a natural bridge over the gap between *interleaving semantics* and *partial-order semantics*, which is highly desirable in distributed languages [3]. Furthermore, as we have shown in Part I, under strong feasibility equivalence-robustness suffices to guarantee the implementability of a fairness notion.

As it turns out, however, several important fairness notions are strongly feasible but are not equivalence-robust. For example, consider the notion of *strong interaction fairness* (SIF), which requires an interaction that is infinitely often enabled to be executed infinitely often. Assume a system $\mathbb{IS}$ with three interactions x_{12}, x_{13}, and x_{24} depicted in Fig. 1, where x_{12} involves p_1 and p_2, x_{13} involves p_1 and p_3, and x_{24} involves p_2 and p_4. SIF is strongly feasible for the system because a nonblocking scheduler satisfying SIF can be constructed by always choosing as the continuation the enabled interaction that is executed the least often; tie is broken arbitrarily. Then the computation $\pi = (p_1 p_3 x_{13} p_2 p_4 x_{24})^\omega$

*A preliminary version of this paper appeared as Y.-J. Joung, 1996, On strong feasibilities of equivalence-Completions, *in* "Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing," Philadelphia, PA, pp. 156–165. This research was supported by the National Science Council, Taipei, Taiwan, Grants NSC 85-2213-E-002-059 and NSC 86-2213-E-002-053, and by the 1997 Research Award of College of Management, National Taiwan University.

† The author is currently visiting Laboratory for Computer Science, Massachusetts Institute of Technology (1999–2000).

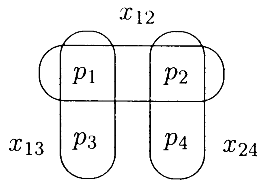


FIG. 1. A system of four processes p_1, p_2, p_3, and p_4 and three interactions x_{12}, x_{13}, and x_{24}.

satisfies SIF, but its equivalent computation $\psi = (p_1 p_3 p_2 x_{13} p_4 x_{24})^\omega$ does not because x_{12} is now enabled in every state immediately after p_2 is ready but it is never executed.

Francez *et al.* [3] propose a notion of *completion* to transform a non-equivalence-robust fairness notion to an equivalence-robust one while maintaining most properties of the source. To understand completions, consider Fig. 2. In this figure, run($\mathbb{IS}$) denotes the set of all possible computations of a system $\mathbb{IS}$, while $\mathbb{C}(\mathbb{IS})$ denotes the set of computations allowed by a fairness notion $\mathbb{C}$. Each partition represents an equivalence class induced by the equivalence relation considered above by permuting independent actions. Of these equivalence classes, X_i's are said to be *purely fair* because they are contained in $\mathbb{C}(\mathbb{IS})$, while Y_j's are *purely unfair* because they do not intersect $\mathbb{C}(\mathbb{IS})$. The classes Z_k's are *mixed* as they contain both fair and unfair computations. A completion has to resolve the fairness of the mixed classes. Thus, the minimal completion (i.e., the strongest completion) can be obtained by treating all mixed classes as unfair, whereas the maximal completion (i.e., the weakest completion) can be obtained by treating all mixed classes as fair. A semantic comparison of the two can be found in [3]. In general, fewer liveness properties can be assumed for programs using the weakest completion, while the strongest completion has exactly the opposite characteristics.

Unfortunately, a completion may not necessarily preserve strong feasibility, meaning that it may not even be implementable. To see this, consider again the system shown in Fig. 1. The computation

$$\pi = (p_1 p_3 x_{13} p_2 p_4 x_{24})^\omega$$

is *inevitable* to all strongly feasible fairness notions of $\mathbb{IS}$, meaning that they must consider π as fair. This is because at any point of the computation at most one interaction is enabled. Thus according to the strong feasibility criterion, when only one interaction is enabled, there must be a continuation allowing the interaction to be executed. (Otherwise the system could be deadlocked if, say when x_{13} is enabled, the scheduling algorithm chooses to wait for more interactions to be enabled while p_2 and p_4 instead are busy doing their local actions forever.) So any completion of SIF, e.g., the minimal completion, that excludes the equivalence class of π would not be strongly feasible.

In this paper we determine, for any given fairness notion $\mathbb{C}$, the strongest strongly feasible completion of $\mathbb{C}$. Recall that strong feasibility is sufficient and necessary to guarantee the implementability of a completion. So we are looking for a strongest implementable completion of $\mathbb{C}$. Our results show that if no interaction contains an interaction (an interaction x *contains* y if $x \neq y$ and the set of participants of y is a subset of x),[1] then the strongest implementable completion of $\mathbb{C}$ exists; otherwise, in general no such completion is possible.

Furthermore, there exists a fairness notion, which we refer to as SG^+, such that when interactions are not allowed to contain interactions, SG^+ is the strongest implementable fairness notion satisfying equivalence-robustness. In other words, all other implementable and equivalence-robust fairness notions must be weaker than SG^+, and all other equivalence-robust fairness notions that are stronger than SG^+ or incomparable with SG^+ must not be implementable. We also compare SG^+ with several existing fairness notions. The results indicate that SG^+ is equivalent to SIF^+ and SPF^+ and is stronger than WIF and WPF^+, where SIF^+, SPF^+, and WPF^+ are the maximal completions of SIF, SPF, and WPF, respectively. In particular, when interactions are CSP-like bipartied, SG^+ is also equivalent to SPF. So *SPF is the strongest equivalence-robust property one can observe from a CSP-like program executing in any asynchronous environment*. Conversely, if interactions can contain interactions, then in general the strongest implementable and equivalence-robust fairness notion does not exist.

[1] Note that although x contains y, the establishment of y does *not* depend on the establishment of x. In the literature, some languages (e.g., IP and Script) allow participants of an interaction v to establish a "subinteraction" within v. So the subinteraction can be established only when v has been established.

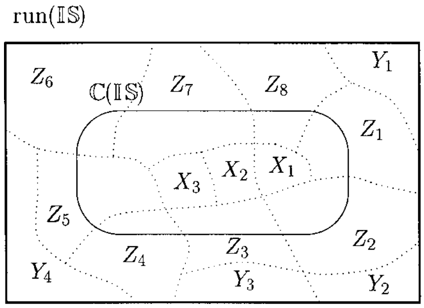


FIG. 2. The fairness-equivalence partitioning [3].

Finally, we show that if equivalence-robustness is not required, then no system can have a strongest implementable fairness notion, unless the system consists of only one interaction. So, in general, for any implementable fairness notion $\mathbb{C}$, there exists another implementable fairness notion $\mathbb{C}'$ such that $\mathbb{C}'$ is either strictly stronger than $\mathbb{C}$ or is incomparable with $\mathbb{C}$. This implies plenty of leeway in the design of fairness notions suitable for various applications.

The rest of the paper is organized as follows. Section 2 provides some preliminaries. Section 3 considers the implementability of completions. Section 4 presents a comprehensive analysis of several commonly used fairness notions and their minimal and maximal completions. The impossibility result of a strongest implementable fairness notion is the subject of Section 5. Section 6 concludes.

2. PRELIMINARIES

We shall follow the same notations as those given in Part I. In addition, we shall use $eq(\pi)$ to denote the equivalence class of π, i.e., the set of runs that are equivalent to π. Moreover, for notational simplicity we shall use a set P in a run to denote an arbitrary sequence of ready transitions, one by each process in P, where each ready transition is of the form $p_i \cdot \{x \in \mathsf{I} \mid p_i \in P_x\}$. For example, assume that $P_{x_{12}} = \{p_1, p_2\}$ and $P_{x_{23}} = \{p_2, p_3\}$. Then $P_{x_{12}} x_{12} P_{x_{23}} x_{23}$ represents a partial run in which p_1 and p_2 become ready (in arbitrary order), they execute x_{12}, and then p_2 and p_3 become ready (again, in arbitrary order) and execute x_{23}. Likewise, $(P_{x_{12}} \cup P_{x_{23}}) x_{12} P_{x_{12}} x_{23}$ represents a partial run in which p_1, p_2, and p_3 become ready, p_1 and p_2 execute x_{12} and become ready again, and then p_2 and p_3 execute x_{23}.

In addition to the definitions given in Part I, the following are used in this artical.

Definition 2.1. A fairness notion $\mathbb{C}_1$ for $\mathbb{IS}$ is *stronger* than $\mathbb{C}_2$ (or, alternatively, $\mathbb{C}_2$ is *weaker* than $\mathbb{C}_1$) if $\mathbb{C}_1(\mathbb{IS}) \subseteq \mathbb{C}_2(\mathbb{IS})$. $\mathbb{C}_1$ is *strictly stronger* than $\mathbb{C}_2$ if $\mathbb{C}_1(\mathbb{IS}) \subsetneq \mathbb{C}_2(\mathbb{IS})$. $\mathbb{C}_1$ and $\mathbb{C}_2$ are *incomparable* if $\mathbb{C}_1(\mathbb{IS}) \not\subseteq \mathbb{C}_2(\mathbb{IS})$ and $\mathbb{C}_2(\mathbb{IS}) \not\subseteq \mathbb{C}_1(\mathbb{IS})$; they are *equivalent* if $\mathbb{C}_1(\mathbb{IS}) = \mathbb{C}_2(\mathbb{IS})$.

Definition 2.2. [1]. A fairness notion $\mathbb{C}$ is *equivalence-robust* for $\mathbb{IS}$ iff $\forall \pi \in \mathbb{C}(\mathbb{IS})$, $eq(\pi) \subseteq \mathbb{C}(\mathbb{IS})$.

The following is a restatement of strongly feasibility.

Definition 2.3. A fairness notion $\mathbb{C}$ is *strongly feasible* for $\mathbb{IS}$ iff there exists a nonblocking scheduler S such that $r(S, A) \in \mathbb{C}(\mathbb{IS})$ for every adversary A.

An immediate consequence of these definitions is the following.

Lemma 2.1. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M})$.

1. *If* $\mathsf{I} \neq \emptyset$, *then for any strongly feasible fairness notion* $\mathbb{C}$, $\mathbb{C}(\mathbb{IS}) \neq \emptyset$.
2. *If* $\mathbb{C}_1$ *is strongly feasible for* $\mathbb{IS}$ *and* $\mathbb{C}_1(\mathbb{IS}) \subseteq \mathbb{C}_2(\mathbb{IS})$, *then* $\mathbb{C}_2$ *is also strongly feasible for* $\mathbb{IS}$.
3. *It may be the case that both* $\mathbb{C}_1$ *and* $\mathbb{C}_2$ *are strongly feasible for* $\mathbb{IS}$, *but* $\mathbb{C}_1 \cap \mathbb{C}_2$ *is not, where the notion* $\mathbb{C}_1 \cap \mathbb{C}_2$ *is defined by* $\mathbb{C}_1(\mathbb{IS}) \cap \mathbb{C}_2(\mathbb{IS})$.

Proof. The first two follow directly from the definitions. For the third, let $\mathbb{IS} = (\{p_1, p_2, p_3\}, \{x_{12}, x_{23}\}, \mathsf{M}^{\forall})$, where $P_{x_{12}} = \{p_1, p_2\}$ and $P_{x_{23}} = \{p_2, p_3\}$. Let $\mathbb{C}_1$ be defined as follows: in a state where both x_{12} and x_{23} are enabled, x_{12} must be scheduled for execution (i.e., x_{12} has a higher priority than x_{23}) and $\mathbb{C}_2$ be defined as "x_{23} has a higher priority than x_{12}." Then, both $\mathbb{C}_1$ and $\mathbb{C}_2$ are strongly feasible, but $\mathbb{C}_1 \cap \mathbb{C}_2$ is not because no scheduler can make a move if the adversary lets the three processes be ready simultaneously. ■

We have shown in Part I that SIF is strongly feasible. Then by Lemma 2.1, any fairness notion that is weaker than SIF is also strongly feasible. Examples include SPF, WPF, and WIF

LEMMA 2.2. *The following four fairness notions are all strongly feasible*: *SIF, SPF, WPF, and WIF.*

On the other hand, as we have shown in Part I, both U-fairness and hyperfairness are not strongly feasible.

In general, to prove that $\mathbb{C}$ is not strongly feasible for $\mathbb{IS}$, we must show that for every nonblocking scheduler S, there is an adversary A such that $r(S, A) \notin \mathbb{C}(\mathbb{IS})$. However, for most systems there exists an adversary such that every scheduler versus it must generate the same run. So if the run is not included in $\mathbb{C}(\mathbb{IS})$, then obviously $\mathbb{C}$ cannot be strongly feasible. For example, if an adversary always lets at most one interaction be enabled, then any nonblocking scheduler versus the adversary must generate the same run: whichever interaction is enabled, it must be selected for execution. The resulting runs are called *singular* in Part I, and are inevitable to every strongly feasible fairness notion of $\mathbb{IS}$. We shall use SG($\mathbb{IS}$) to denote the set of singular runs of $\mathbb{IS}$. An immediate consequence of the definition is the following.

LEMMA 2.3. *For every strongly feasible fairness notion* $\mathbb{C}$, SG($\mathbb{IS}$) $\subseteq \mathbb{C}(\mathbb{IS})$.

Note that for every $\mathbb{IS}$, SG($\mathbb{IS}$) $\neq \emptyset$, unless $\mathbb{IS}$ contains more than one interaction and for every interaction x, there exists another interaction y such that y is contained in x. Moreover, if SG($\mathbb{IS}$) $\neq \emptyset$, then for every fairness notion $\mathbb{C}$ it is never the case that SG($\mathbb{IS}$) $\subseteq \mathbb{C}(\mathbb{IS})$ and SG($\mathbb{IS}$) $\subseteq \overline{\mathbb{C}}(\mathbb{IS})$, where $\overline{\mathbb{C}}$ is the complement of $\mathbb{C}$ defined by $\overline{\mathbb{C}}(\mathbb{IS}) = \text{run}(\mathbb{IS}) - \mathbb{C}(\mathbb{IS})$. Therefore, if $\mathbb{C}$ is strongly feasible for $\mathbb{IS}$, then $\overline{\mathbb{C}}$ must not be strongly feasible. In other words, it is never the case that both $\mathbb{C}$ and its complement are implementable.

COROLLARY 2.4. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^{\forall})$. *If* $\exists\, x \in \mathsf{I}, \forall\, y \in \mathsf{I}, y \neq x \Rightarrow P_y \not\subseteq P_x$. *Then for every fairness notion* $\mathbb{C}$, *it is never the case that both* $\mathbb{C}$ *and* $\overline{\mathbb{C}}$ *are implementable for* $\mathbb{IS}$.

Recall from Lemma 4.1 of Part I that in the presence of equivalence-robustness, strong feasibility is sufficient and necessary to determine fairness implementability. As this result will be referenced several times in the paper, for ease of reference, we restate the lemma below.

LEMMA 2.5. *If* $\mathbb{C}$ *is strongly feasible and equivalence-robust for* $\mathbb{IS}$, *then* $\mathbb{C}$ *is implementable for* $\mathbb{IS}$.

Moreover, since if $\mathbb{C}$ is stronger than $\mathbb{C}'$ then $\mathbb{C}(\mathbb{IS}) \subseteq \mathbb{C}'(\mathbb{IS})$, the fairness implementability criterion immediately implies the following lemma:

LEMMA 2.6. *Suppose* $\mathbb{C}$ *is stronger than* $\mathbb{C}'$. *If* $\mathbb{C}$ *is implementable for* $\mathbb{IS}$, *then so is* $\mathbb{C}'$; *and if* $\mathbb{C}'$ *is not implementable, then neither is* $\mathbb{C}$.

3. STRONG FEASIBILITY OF COMPLETIONS

In this section we consider the implementability of equivalence-robust fairness notions. In particular, we shall focus on *completions*—equivalence-robust fairness notions derived from a non-eqivalence-robust one. We shall show that the process of completion may not preserve strong feasibility. Since in the presence of equivalence-robustness strong feasibility suffices to determine fairness implementability, completions are not necessarily implementable.

Furthermore, we shall also show that if no interaction of $\mathbb{IS}$ contains an interaction, then there exists a fairness notion, denoted by SG$^+$, such that SG$^+$ is the strongest implementable and equivalence-robust fairness notion of $\mathbb{IS}$. On the other hand, if $\mathbb{IS}$ contains two interactions x, y such that $P_x \subseteq P_y$ then in general there does not exist a strongest implementable and equivalence-robust fairness notion for $\mathbb{IS}$.

3.1. Definitions

DEFINITION 3.1 [3]. 1. An equivalence class $eq(\pi)$ in run($\mathbb{IS}$) is *purely* $\mathbb{C}$-*fair* iff $eq(\pi) \subseteq \mathbb{C}(\mathbb{IS})$, it is *purely* $\mathbb{C}$-*unfair* iff $eq(\pi) \cap \mathbb{C}(\mathbb{IS}) = \emptyset$, and it is $\mathbb{C}$-*mixed* otherwise.

2. A fairness notion $\hat{\mathbb{C}}$ is a comletion of $\mathbb{C}$ iff the following three conditions are satisfied:
   (i) for every purely $\mathbb{C}$-fair class $eq(\pi)$, $eq(\pi) \subseteq \hat{\mathbb{C}}(\mathbb{IS})$
   (ii) for every purely $\mathbb{C}$-unfair class $eq(\pi)$, $eq(\pi) \cap \hat{\mathbb{C}}(\mathbb{IS}) = \emptyset$, and
   (iii) for every class $\mathbb{C}$-mixed class $eq(\pi)$, either $eq(\pi) \subseteq \hat{\mathbb{C}}(\mathbb{IS})$ or $eq(\pi) \cap \hat{\mathbb{C}}(\mathbb{IS}) = \emptyset$.

It follows directly that a completion $\hat{\mathbb{C}}$ must be equivalence-robust. Two extreme completions of $\mathbb{C}$ arise naturally: the *maximal* completion $\mathbb{C}^+$, which treats every $\mathbb{C}$-mixed class as fair, and the *minimal* completion $\mathbb{C}^-$, which treats every $\mathbb{C}$-mixed class as unfair [3]. Moreover, $\mathbb{C}^-(\mathbb{IS}) \subseteq \mathbb{C}(\mathbb{IS}) \subseteq \mathbb{C}^+(\mathbb{IS})$ for every $\mathbb{C}$. By Lemma 2.1, if $\mathbb{C}$ is strongly feasible then so is $\mathbb{C}^+$. Since $\mathbb{C}^+$ is equivalence-robust, by Lemma 2.5, if $\mathbb{C}$ is strongly feasible then $\mathbb{C}^+$ must be implementable.

LEMMA 3.1. *If $\mathbb{C}$ is strongly feasible for $\mathbb{IS}$, then $\mathbb{C}^+$ must be implementable for $\mathbb{IS}$.*

On the other hand, since the process of minimal completion may exclude some runs that are inevitable to $\mathbb{C}^-$, $\mathbb{C}^-$ may be unimplementable. In fact, if $\mathbb{C}$ is not implementable then strong feasibility cannot be preserved by $\mathbb{C}^-$. Therefore, unlike maximal completions, minimal completions do not help us obtain an implementable fairness notion from an unimplementable one while pursuing equivalence-robustness.

LEMMA 3.2. *If $\mathbb{C}$ is not implementable for $\mathbb{IS}$, then neither is $\mathbb{C}^-$. Moreover, if $\mathbb{C}$ is not implementable, then $\mathbb{C}^-$ must not be strongly feasible.*

Proof. Since $\mathbb{C}^-(\mathbb{IS}) \subseteq \mathbb{C}(\mathbb{IS})$, by Lemma 2.6, if $\mathbb{C}$ is not implementable then neither is $\mathbb{C}^-$. Since $\mathbb{C}^-$ is equivalence-robust, by Lemma 2.5, it must not be strongly feasible. ■

3.2. Strongly Feasible Completions

As it turns out, the weakest completion (i.e., the maximal completion) of a strongly feasible fairness notion is also strongly feasible, while the strongest completion (i.e., the minimal completion) may not be. Since strongly feasible completions are implementable, and since stronger completions induce more liveness properties, given a strongly feasible fairness notion $\mathbb{C}$, one would wish to know what is the strongest, strongly feasible completion of $\mathbb{C}$, or does it even exist. To answer this, we first establish a more important theorem showing the existence of a strongest fiarness notion that is both strongly feasible and equivalence-robust. For this, we need the following lemma.

LEMMA 3.3. *Let $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M})$ and assume that $\forall x, y \in \mathsf{I}, x \neq y \Rightarrow P_x \not\subseteq P_y$. The every $\pi \in$ run($\mathbb{IS}$) satisfying SIF is equivalent to a singular run.*

Proof. Let $\pi \in$ run($\mathbb{IS}$) be a run given by

$$\pi = p_{1,1}.I_{1,1} \ldots p_{1,n_1}.I_{1,n_1}\ x_1\ p_{2,1}.I_{2,1} \ldots p_{2,n_2}.I_{2,n_2}\ x_2 \ldots,$$

where $x_1, x_2, \ldots$ are the sequence of interactions executed in π. Assume that π satisfies SIF. Consider x_1. Suppose that we transform π into another run π_1 by the following procedure: For each $p_{1,j}.I_{1,j}$, $1 \leq j \leq n_1$, if (1) $p_{1,j} \notin P_{x_1}$ and (2) at some point in π (after the ready transition $p_{1,j}.I_{1,j}$) some interaction y involving $p_{1,j}$ is enabled, then move the action $p_{1,j}.I_{1,j}$ after x_1. Clearly, $\pi \equiv \pi_1$. Due to the restriction imposed on the structure of $\mathbb{IS}$, no subset of P_{x_1} is involved in any other interaction. So in π_1 at most one interaction is enabled at any point up to x_1.

Similarly, we can transform π_1 into another run π_2 by applying the above procedure to the ready transitions occurring between x_1 and x_2 so that $\pi_1 \equiv \pi_2$, and in π_2 at most one interaction is enabled at any point up to x_2. We claim that if we apply the procedure repeatedly for the rest of x_i's, then we will obtain a run π_∞ such that $\pi \equiv \pi_\infty$ and π_∞ is singular. To see this, observe that for any finite i, we have $\pi \equiv \pi_1 \equiv \cdots \equiv \pi_i$, and in π_i at most one interaction is enabled at any point up to x_i So it suffices to show that the equivalence relation between π and π_i is preserved when $i \to \infty$.[2]

[2] Note that, in general, equivalence relation may *not* be preserved through an infinite number of such transformations. For example, consider $\pi = p_1(p_2 p_3 x)^\omega$, and assume that $P_y = \{p_1, p_2\}$ and $P_x = \{p_2, p_3\}$. Let $\pi_i = (p_2 p_3 x)^i p_1 (p_2 p_3 x)^\omega$. Then, for each finite i, $\pi_{i-1} \equiv \pi_i$. However, $\pi_\infty = (p_2 p_3 x)^\omega$, which is not equivalent to π_i for any finite i.

Suppose otherwise that the equivalence relation does not hold. Then, it must be the case that some ready transition $p_{k,j}.I_{k,j}$ in between x_{k-1} and x_k of run π_k has to be moved during the transformation from π_{l-1} to π_l for every $l \geq k$, resulting in the extinction of $p_{k,j}.I_{k,j}$ when the transformation is performed an infinite number of times. However, recall that if $p_{k,j}.I_{k,j}$ has to be moved in the kth transformation (i.e., from π_{k-1}) to π_k, then (1) $p_{k,j} \notin P_{x_k}$ and (2) later at some point in π_{k-1} (and thus in π) some interaction y involving $p_{k,j}$ is enabled. Since in the remaining transformations $p_{k,j}.I_{k,j}$ is continually moved, none of the interactions $x_k, x_{k+1}, \ldots$ involves process $p_{k,j}$, and there exists an infinitely number of points in π such that at each point some interaction involving $p_{k,j}$ is enabled, but from x_k onward $p_{k,j}$ never participates in any interaction. So π does not satisfy SPF, and thus π does not satisfy SIF. This contradicts the assumption that π satisfies SIF. Therefore, the equivalence relation between π and π_∞ is preserved during the transformations from π to π_∞. ■

Recall from Lemma 2.3 that for every strongly feasible fairness notion $\mathbb{C}$, $\mathrm{SG}(\mathbb{IS}) \subseteq \mathbb{C}(\mathbb{IS})$, where $\mathrm{SG}(\mathbb{IS})$ is the set of singular runs of $\mathbb{IS}$. Define fairness notion SG^+ to be the maximal completion of SG, i.e.,

$$\mathrm{SG}^+(\mathbb{IS}) = \bigcup_{\pi \in \mathrm{SG}(\mathbb{IS})} eq(\pi).$$

Then, for every $\mathbb{C}$ that is both strongly feasible and equivalence-robust, $\mathrm{SG}^+(\mathbb{IS}) \subseteq \mathbb{C}(\mathbb{IS})$. Moreover, by Lemma 3.3, $\mathrm{SIF}(\mathbb{IS}) \subseteq \mathrm{SG}^+(\mathbb{IS})$. Since SIF is strongly feasible, by Lemma 2.1 SG^+ is also strongly feasible. Hence, by Lemma 2.5 SG^+ is implementable. The following theorem can thus be established.

THEOREM 3.4. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M})$ *and assume that* $\forall x, y \in \mathsf{I}, x \notin y \Rightarrow P_x \not\subseteq P_y$. *Then,* SG^+ *is strongly feasible and equivalence-robust for* $\mathbb{IS}$, *and for every other strongly feasible and equivalence-robust fairness notion* $\mathbb{C}$, $\mathrm{SG}^+(\mathbb{IS}) \subseteq \mathbb{C}(\mathbb{IS})$.

To illustrate SG^+, consider the following example taken from [1]:

$p_1 :: b_1 := \mathit{true};$

$\quad {}^*[b_1;\ p_2\,!\,0 \rightarrow b_1 := \mathit{false}]$

$p_2 :: b_2 := \mathit{true};$

$\quad {}^*[b_2;\ p_1\,?\,x \rightarrow b_2 := \mathit{false}$

$\quad \Box\ b_2;\ p_3\,?\,x \rightarrow \mathit{skip}];$

$\quad p_3\,!\,0$

$p_3 :: b_3 := \mathit{true};$

$\quad {}^*[b_3;\ p_2\,!\,0 \rightarrow \mathit{skip}$

$\quad \Box\ b_3;\ p_2\,?\,y \rightarrow b_3 := \mathit{false}]$

In this system, p_1 and p_2 may interact, and p_2 and p_3 may interact. In particular, p_2 and p_3 may establish two possible interactions, one to deliver a value from p_3 to p_2 and the other in the opposite direction. Although the two interactions contain each other, the program does not allow them to be enabled simultaneously. So Theorem 3.4 can be applied to the system so that SG^+ is the strongest implementable and equivalence-robust fairness notion for the system. From the program, it can be seen that the system may not terminate as p_2 and p_3 may repeatedly establish an interaction forever. However, any run of the system satisfying SG^+ must terminate.

From Theorem 3.4, we can derive the following corollary.

COROLLARY 3.5. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M})$ *and assume that* $\forall x, y \in \mathsf{I}, x \neq y \Rightarrow P_x \not\subseteq P_y$. *Then for every strongly feasible fairness notion* $\mathbb{C}$, *the completion* $\mathbb{C}^*$ *defined by*

$$\mathbb{C}^*(\mathbb{IS}) = \mathrm{SG}^+(\mathbb{IS}) \cup \{\pi \in E \mid E \textit{ is a purely } \mathbb{C}\textit{-fair equivalence class in } \mathrm{run}(\mathbb{IS})\}$$

is the strongest implementable completion of $\mathbb{C}$.

Proof. Since $\mathbb{C}$ is strongly feasible, by Lemma 2.3, SG($\mathbb{IS}$) $\subseteq \mathbb{C}(\mathbb{IS})$. So every equivalence class in SG$^+$($\mathbb{IS}$) must be purely $\mathbb{C}$-fair or $\mathbb{C}$-mixed. So by definition $\mathbb{C}^*$ is a completion of $\mathbb{C}$. Moreover, since SG$^+$($\mathbb{IS}$) $\subseteq \mathbb{C}^*(\mathbb{IS})$ and since SG$^+$ is strongly feasible, by Lemma 2.1, $\mathbb{C}^*$ is also strongly feasible. By Lemma 2.5, therefore, $\mathbb{C}^*$ is an implementable completion of $\mathbb{C}$.

To show that $\mathbb{C}^*$ is the strongest implementable completion of $\mathbb{C}$, let $\hat{\mathbb{C}}$ be any other implementable completion of $\mathbb{C}$. Then, by Theorem 3.4 and by the definition of completions, $\mathbb{C}^*(\mathbb{IS}) \subseteq \hat{\mathbb{C}}(\mathbb{IS})$. Hence, $\mathbb{C}^*$ is the strongest strongly feasible completion of $\mathbb{C}$. ■

Note that Theorem 3.4 does not depend on the arity of interactions, and so it holds as well if interactions are strictly bipartied. In particular, if every pair of processes share at most one interaction, then SG$^+$ is the strongest strongly feasible and equivalence-robust fairness notion one can get for biparty interaction systems.

COROLLARY 3.6. *For every* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M})$ *such that* $\forall x \in \mathsf{I}, |P_x| = 2$ *and* $\forall x, y \in \mathsf{I}, x \neq y \Rightarrow P_x \neq P_y$, SG$^+$ *is the strongest fairness notion for* $\mathbb{IS}$ *that is both strongly feasible and equivalence-robust.*

3.3. Non-Strongly-Feasible Completions

On the other hand, if some interaction contains an interaction, then the strongest strongly feasible and equivalence-robust fairness notion may not exist. Before proving this, we first show that if $\mathbb{IS}$ in addition contains at least two processes, then not all strongly feasible fairness notions of $\mathbb{IS}$ have a strongest implementable completion. Note that for this we shall consider interaction systems with programs of type $\mathsf{M}^\forall$. It can be seen that even if interactions may contain interactions, if the associated program M guarantees that at any time no enabled interaction x contains an interaction that is also enabled simultaneously, then SG$^+$ is still the strongest strongly feasible and equivalence-robust fairness notion for the system.

THEOREM 3.7. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^\forall)$ *be an interaction system satisfying the following conditions*:

1. $|\mathsf{P}| > 1$ *and*
2. $\exists x, y \in \mathsf{I}, x \neq y, P_x \subseteq P_y,$ *and* $\forall z \in \mathsf{I}, P_z \subseteq P_y \Rightarrow P_x \subseteq P_z$.

Then, there exists a fairness notion $\mathbb{C}$ *which is strongly feasible but does not have a strongest strongly feasible completion.*

Proof. Let $x, y \in \mathsf{I}$ be two interactions satisfying condition (2). Consider first that $|P_y| \geq 2$. let p_1 and p_2 be two arbitrary processes in P_y. Clearly, run($\mathbb{IS}$) contains runs of the form

$$p_1 p_2 (P_y - \{p_1, p_2\}) z_1 P_{z_1} z_2 P_{z_2} z_3 P_{z_3} \ldots . \tag{1}$$

Let E_x and $E_{\bar{x}}$ be two subsets of run($\mathbb{IS}$) defined by

$$E_x = \{\pi \mid \pi \text{ is equivalent to some run of form (1) where } \forall i, z_i = x\}$$
$$E_{\bar{x}} = \{\pi \mid \pi \text{is equivalent to some run of form (1) where } \exists i, z_i \neq x\}.$$

The two sets are obviously not empty and disjoint. Note that in the presence of p_1 and p_2, each run π in E_x and $E_{\bar{x}}$ has at least one equivalent run different from π, i.e., $|eq(\pi)| \geq 2$ (because the first ready transitions of the two processes can be arbitrarily permuted).

Let S be a nonblocking scheduler which selects an arbitrary enabled interaction for execution, except:

- If initially the adversary schedules the sequence of ready transitions $p_1 p_2 (P_y - \{p_1, p_2\})$, then x will always be chosen for execution whenever it is enabled.
- If initially the adversary schedules the sequence of ready transitions $p_2 p_1 (P_y - \{p_1, p_2\})$, then y will always be chosen for execution whenever it is enabled.

Define fairness notion $\mathbb{C}$ to be the following:

$$\mathbb{C}(\mathbb{IS}) = \{\pi \mid \text{there is an adversary } A \text{ of } \mathbb{IS} \text{ such that } r(S, A) = \pi\}.$$

Then $\mathbb{C}$ is strongly feasible because S generates only $\mathbb{C}$-fair runs. Observe that every run of the form $p_1 p_2(P_y - \{p_1, p_2\})(xP_x)^\omega$ belongs to $\mathbb{C}(\mathbb{IS})$ but its equivalent run $p_2 p_1(P_y - \{p_1, p_2\})(xP_x)^\omega$ does not. So every equivalence class $eq(\pi) \subseteq E_x$ is $\mathbb{C}$-mixed. (In fact, there is only one equivalence class in E_x.) On the other hand, for every π of form (1) such that for some i, $z_i \neq x$, π must not belong to $\mathbb{C}(\mathbb{IS})$ because no adversary versus S can generate π. So every equivalence class $eq(\pi) \subseteq E_{\bar{x}}$ is either $\mathbb{C}$-mixed or is purely $\mathbb{C}$-unfair.

Recall from Lemma 2.1 that if $\mathbb{C}_1$ is strongly feasible and $\mathbb{C}_1(\mathbb{IS}) \subseteq \mathbb{C}_2(\mathbb{IS})$, then $\mathbb{C}_2$ must also be strongly feasible. From the above description of $\mathbb{C}$, it can be seen that there exists a strongly feasible fairness notion $\mathbb{C}'$ for $\mathbb{IS}$ satisfying the following two conditions:

- $\forall \pi \in \text{run}(\mathbb{IS}) - (E_x \cup E_{\bar{x}})$, $\pi \in \mathbb{C}'(\mathbb{IS})$, and
- $\forall \pi \in E_x \cup E_{\bar{x}}$, $eq(\pi) \cap \mathbb{C}'(\mathbb{IS}) \neq \emptyset$ and $eq(\pi) \cap \overline{\mathbb{C}'}(\mathbb{IS}) \neq \emptyset$.

For example, $\mathbb{C}'$ can be obtained by extending the above $\mathbb{C}$ to include every run in $\text{run}(\mathbb{IS}) - (E_x \cup E_{\bar{x}})$ and including one run from every purely $\mathbb{C}$-unfair class of $E_{\bar{x}}$.

Now, since each equivalence class $eq(\pi) \subseteq E_x \cup E_{\bar{x}}$ is $\mathbb{C}'$-mixed, every completion $\hat{\mathbb{C}}$ of $\mathbb{C}'$ must decide the fairness of $eq(\pi)$. Moreover, if $\hat{\mathbb{C}}'$ is strongly feasible, then $\hat{\mathbb{C}}'$ cannot treat all the $eq(\pi)$'s in $E_x \cup E_{\bar{x}}$ as unfair; otherwise, no nonblocking scheduler can generate a fair run if it faces the following adversary. Initially, the adversary schedules the sequence of partial run $p_1 p_2(P_y - \{p_1, p_2\})$; subsequently, whichever interaction is chosen by the scheduler, the adversary in response simply schedules the processes of the interaction to be ready again.

We can define two completions $\hat{\mathbb{C}}'_x$ which treats all equivalence classes in E_x as fair and those in $E_{\bar{x}}$ as unfair, and $\hat{\mathbb{C}}'_{\bar{x}}$, which is defined in the other way. We argue that both are strongly feasible. To see this, consider first a nonblocking scheduler S_x which selects an arbitrary enabled interaction for executioin, except that x must be chosen whenever it is enabled. Then for every run π generated by S_x, if π is equivalent to some run of form (1), then all of the z_i's must be instances of x. This is because at any point in π only the processes in P_y can be ready. So for every z_i, $P_{z_i} \subseteq P_y$. By condition (2), x is enabled whenever z_i is. Since S_x prefers x to any other interaction containing x, $z_i = x$. So S_x cannot generate any run in $E_{\bar{x}}$. Therefore, $\hat{\mathbb{C}}'_x$ is strongly feasible because S_x generates only runs in $\hat{\mathbb{C}}'_x$.

For the strong feasibility of $\hat{\mathbb{C}}'_{\bar{x}}$ consider a nonblocking scheduler $S_{\bar{x}}$ which selects an arbitrary enabled interaction for execution, except that y must be chosen whenever y is enabled. Since every run in E_x contains a state in which y is enabled, no adversry versus $S_{\bar{x}}$ can generate a run in E_x. So every run generated by $S_{\bar{x}}$ satisfies $\hat{\mathbb{C}}'_{\bar{x}}$. Hence, $\hat{\mathbb{C}}'_{\bar{x}}$ is also strongly feasible.

Observe that the two completions $\hat{\mathbb{C}}'_x$ and $\hat{\mathbb{C}}'_{\bar{x}}$ are incomparable. So neither of them can be the strongest implementable completion of $\mathbb{C}'$. Moreover, recall that for every strongly feasible completion $\hat{\mathbb{C}}'$ of $\mathbb{C}'$, $\hat{\mathbb{C}}'(\mathbb{IS})$ must contain some equivalence classes in $E_x \cup E_{\bar{x}}$. So if $\hat{\mathbb{C}}'$ contains all equivalence classes in $E_x \cup E_{\bar{x}}$, then $\hat{\mathbb{C}}'$ must be weaker than $\hat{\mathbb{C}}'_x$ and $\hat{\mathbb{C}}'_{\bar{x}}$; and if $\hat{\mathbb{C}}'(\mathbb{IS})$ contains only part of them, then $\hat{\mathbb{C}}'$ must be incomparable with either $\hat{\mathbb{C}}'_x$ or $\hat{\mathbb{C}}'_{\bar{x}}$. Therefore, $\mathbb{C}'$ does not have a strongest strongly feasible completion.

In the above proof, we have assumed that $|P_y| \geq 2$. If $|P_y| = 1$, then $P_x = P_y$. Since $|P| > 1$, either there exists an interaction u involving more than one process, or there exists two interactions u_1, u_2 such that $|P_{u_1}| = |P_{u_2}| = 1$ and $P_{u_1} \neq P_{u_2}$. In the former case, we can modify form (1) to

$$p_1 p_2(P_u - \{p_1, p_2\}) v_1 \cdots v_k P_y z_1 P_{z_1} z_2 P_{z_2} z_3 P_{z_3} \ldots,$$

where p_1 and p_2 are two arbitrary process in P_u and $v_1, \ldots, v_k$ are instances of interactions such that no interaction is enabled immediately after v_k. In the latter case, we instead consider the form

$$p_1 u'_1 p_2 u'_2 P_y z_1 P_{z_1} z_2 P_{z_2} z_3 P_{z_3} \ldots,$$

where $P_{u_1} = \{p_1\}$ and $P_{u_2} = \{p_2\}$. In either case, we can define E_x and $E_{\bar{x}}$ analogously and show that there exists a strongly feasible fairness notion $\mathbb{C}'$ such that all its strongly feasible completions intersect (a) both E_x and $E_{\bar{x}}$, (b) only E_x, or (c) only $E_{\bar{x}}$ So $\mathbb{C}'$ does not have a strongest strongly feasible completion. ■

Note that in Theorem 3.7, if P consists of only one process, then for any $\pi \in \text{run}(\mathbb{IS})$, the equivalence class $eq(\pi)$ consists of π itself. So for any fairness notion $\mathbb{C}$, there is only one completion, i.e., $\mathbb{C}$ itself. Therefore, it holds trivially that if $\mathbb{C}$ is strongly feasible for $\mathbb{IS}$, then $\mathbb{C}$ is the strongest strongly feasible completion of $\mathbb{C}$.

We now show tht if some interaction contains an interaction, then there may not exist a strongest fairness notion for $\mathbb{IS}$ that is both strongly feasible and equivalence-robust.

THEOREM 3.8. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^{\forall})$ *be an interaction system satisfying the following condition*:

$$\exists x, y \in \mathsf{I}, x \neq y, P_x \subseteq P_y, \quad \text{and} \quad \forall z \in \mathsf{I}, P_z \subseteq P_y \Rightarrow P_x \subseteq P_z.$$

Then, there does not exist a fairness notion $\mathbb{C}$ *such that* (1) $\mathbb{C}$ *is strongly feasible and equivalence-robust and* (2) $\mathbb{C}$ *is the strongest fairness notion for* $\mathbb{IS}$ *that satisfies Condition* (1).

Proof. The proof is similar to that of Theorem 3.7, except that we do not need processes p_1 and p_2 to make some equivalence classes of run($\mathbb{IS}$) consist of more than one run. Let E_x and $E_{\bar{x}}$ be two subsets of run $\mathbb{IS}$ defined by

$$E_x = \{\pi \mid \pi \text{ is equivalent to some run of the form } P_y(xP_x)^{\omega}\},$$
$$E_{\bar{x}} = \{\pi \mid \pi \text{ is equivalent to some run of the form } P_y z_1 P_{z_1} z_2 P_{z_2} \ldots \text{where } \exists i, z_i \neq x\}.$$

Then, using an argument similar to Theorem 3.7, we can show that any strongly feasible and equivalence-robust fairness notion $\mathbb{C}$ for $\mathbb{IS}$ must intersect either E_x or $E_{\bar{x}}$. Furthermore, there are two incomparable fairness notions (which are strongly feasible and equivalence-robust) $\mathbb{C}_x$ and $\mathbb{C}_{\bar{x}}$ such that the following two conditions are satisfied: (1) $E_x \subseteq \mathbb{C}_x(\mathbb{IS})$ and $\mathbb{C}_x(\mathbb{IS}) \cap E_{\bar{x}} = \emptyset$ and (2) $E_{\bar{x}} \subseteq \mathbb{C}_{\bar{x}}(\mathbb{IS})$ and $\mathbb{C}_{\bar{x}}(\mathbb{IS}) \cap E_x = \emptyset$. All other strongly feasible and equivalence-robust fairness notions for $\mathbb{IS}$ must be either weaker than $\mathbb{C}_x$ and $\mathbb{C}_{\bar{x}}$ or incomparable with one of them. Therefore, the strongest strongly feasible and equivalence-robust fairness notion for $\mathbb{IS}$ does not exist. ■

Note that like Theorem 3.4, both Theorems 3.7 and 3.8 hold as well if interactions are strictly bipartied.

To illustrate Theorem 3.8, let $\mathbb{IS} = (\{p\}, \{x, y\}, \mathsf{M}^{\forall})$, where $P_x = P_y = \{p\}$. Let $F_x = \{(px)^{\omega}\}$ and $F_x = \{(py)^{\omega}\}$. Then both are strongly feasible and equivalence-robust (and so are implementable). However, the two fairness notions are incomparable. Note that each fairness notion has only one completion, i.e., itself. So each has a strongest implementable completion.

3.4. A Patch

The readers may have noticed that Theorem 3.8 alone is not enough to determine whether there exists a strongest implementable and equivalence-robust fairness notion for all $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^{\forall})$ where some interaction contains an interaction. This is because Theorem 3.8 concerns only the case where some interaction $y \in \mathsf{I}$ contains a minimal interaction x (where an interaction u is *minimal* if for every interaction v contained in u, $P_u = P_v$) such that for all other minimal interactions w contained in y, $P_w = P_x$. Clearly, y may contain two minimal interactions x and z such that neither of them contains the other. As we shall see, these systems do have a strongest implementable and equivalence-robust fairness notion, which is, by no surprise, SG$^+$.

LEMMA 3.9. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^{\forall})$ *and assume that*

$$\forall x, y \in \mathsf{I}, x \neq y, P_x \subseteq P_y \Rightarrow \exists z, w \in \mathsf{I}, P_z \subseteq P_y, P_w \subseteq P_y, P_z \not\subseteq P_w, \textit{ and } P_w \not\subseteq P_z.$$

Then a run $\pi \in \text{run}(\mathbb{IS})$ *is equivalent to a singular run if it satisfies the following conditions*:

1. *No interaction containing an interaction is ever executed in* π.

2. *For every* x *not containing any interaction, if* x *is enabled infinitely often, then it is executed infinitely often.*

Proof. Let $\pi \in \text{run}(\mathbb{IS})$ be a run given by

$$\pi = p_{1,1}.I_{1,1} \ldots p_{1,n_1}.I_{1,n_1}\, x_1\, p_{2,1}.I_{2,1} \ldots p_{2,n_2}.I_{2,n_2}\, x_2 \ldots,$$

where $x_1, x_2, \ldots$ are the sequence of interactions executed in π. Assume that π satisfies conditioins 1 and 2. We can use the method described in Lemma 3.3 to transform π into an equivalent run π_1 by moving each $p_{1,j}.I_{1,j}$, $1 \le j \le n_1$, to the end of x_1, where $p_{1,j}.I_{1,j}$ satisfies the following two conditions: (a) $p_{1,j} \notin P_{x_1}$ and (b) at some point in π (after the ready transition $p_{1,j}.I_{1,j}$) some interaction y involving $p_{1,j}$ is enabled. By condition 1 of the lemma π_1 is singular up to x_1.

Like Lemma 3.3, the transformation can be done ad infinitum. Let π_∞ denote the resulting run. Unlike Lemma 3.3, however, some ready transition $p_{k,j}.I_{k,j}$ in between x_{k-1} and x_k may be kept moving forever in the rest of the transformation from π_k to π_∞. (If no ready transition is moved indefinitely then $\pi \equiv \pi_\infty$ and π_∞ is singular; hence we are done.) If this happens, then by the transformation some interaction y involving process $p_{k,j}$ is enabled infinitely often but from x_k onward y is never executed. Note that by the conditions of the lemma, y must contain two interactions w and z such that $P_w - P_z \neq \emptyset$, $P_z - P_w \neq \emptyset$, and neither of them contains an interaction. Since both w and z are enabled whenever y is enabled, w and z are enabled infinitely often throughout π. Since they do not contain any interaction, by condition 2 they are executed infinitely often in π.

We can then modify the transformation such that starting from π_k the ready transition $p_{k,j}.I_{k,j}$ will not be moved in the rest of the transformation. Without loss of generality assume that no other ready transition is kept moving forever in the new transformation. (If there is one, then we can use the same method to freeze that transition too.) Let π'_∞ be the resulting run. It is clear that $\pi \equiv \pi'_\infty$ because no ready transition is kept moving forever.

We claim that π'_∞ is singular. By the transformation, it suffices to show that y will never be enabled even if we stop moving $p_{k,j}.I_{k,j}$ from x_k onward. For this, suppose otherwise y is enabled in some state s in π'_∞. So w and z are also enabled in s. Recall that w and z are executed infinitely often in π (and thus in π'_∞). Let u be the first interaction that is executed after s. By condition 1, P_u cannot contain P_w and P_z. So either $P_w - P_u \neq \emptyset$ (when $u \neq w$) or $P_z - P_u \neq \emptyset$ (when $u \neq z$). Since both w and z are executed infinitely often, by the transformation either the ready transitions by the processes in $P_w - P_u$ or the ready transitions by the processes in $P_z - P_u$ will be moved after s. Hence, y cannot be enabled in s; contradiction. The lemma is thus proven. ■

To illustrate Lemma 3.9, assume that $\mathbb{IS} = (\{p_1, p_2, p_3\}, \{x_2, x_3, x_{23}, x_{123}\}, \mathsf{M}^\forall)$, where $P_{x_2} = \{p_2\}$, $P_{x_3} = \{p_3\}$, $P_{x_{23}} = \{p_2, p_3\}$, and $P_{x_{123}} = \{p_1, p_2, p_3\}$ (see Fig. 3c). Then the run

$$\pi = p_1 p_2 p_3 (x_2 p_2 x_2 p_2 x_3 p_3)^\omega$$

can be transformed into

$$\rho = p_1 (p_2 x_2 p_2 x_2 p_3 x_3)^\omega$$

which is singular and is equivalent to π.

THEOREM 3.10. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^\forall)$ *and assume that*

$$\forall x, y \in \mathsf{I}, x \neq y, P_x \subseteq P_y \Rightarrow \exists z, w \in \mathsf{I}, P_z \subseteq P_y, P_w \subseteq P_y, P_z \not\subseteq P_w, \textit{ and } P_w \not\subseteq P_z$$

Then, SG$^+$ *is the strongest strongly feasible and equivalence-robust fairness notion for* $\mathbb{IS}$.

Proof. By Lemma 2.3, for every strongly feasible and equivalence-robust fairness notion $\mathbb{C}$, SG$^+(\mathbb{IS}) \subseteq \mathbb{C}(\mathbb{IS})$. As by definition SG$^+$ is equivalence-robust, to prove the theorem it suffices to show that SG$^+$ is strongly feasible for $\mathbb{IS}$. Moreover, by Lemma 3.9, it suffices to show that there exists a nonblocking scheduler for $\mathbb{IS}$ satisfying conditions 1 and 2 of the lemma. Such a scheduler can be easily obtained by modifying the SIF-scheduler presented in Part I (Fig. 6) so that no interaction containing an interaction is ever selected for execution. ■

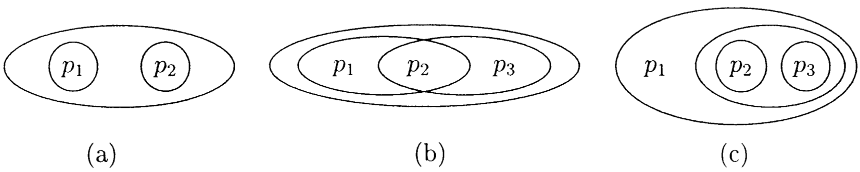


FIG. 3. Instances of interaction systems $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^{\forall})$ which permit some interactions to contain an interaction but have a strongest implementable and equivalence-robust fairness notion.

Figure 3 illustrates some instances of $\mathbb{IS}$ that have a strongest implementable and equivalence-robust fairness notion. Theorem 3.10 implies the following corollary (cf. Corollary 3.5).

COROLLARY 3.11. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^{\forall})$ *and assume that*

$$\forall x, y \in \mathsf{I}, x \neq y, P_x \subseteq P_y \Rightarrow \exists z, w \in \mathsf{I}, P_z \subseteq P_y, P_w \subseteq P_y, P_z \not\subseteq P_w, \text{ and } P_w \not\subseteq P_z.$$

Then for every strongly feasible fairness notion $\mathbb{C}$*, the completion* $\mathbb{C}^*$ *defined by*

$$\mathbb{C}^*(\mathbb{IS}) = \mathrm{SG}^+(\mathbb{IS}) \cup \{\pi \in E \mid E \text{ is a purely } \mathbb{C}\text{-fair equivalence class in } \mathrm{run}(\mathbb{IS})\}$$

is the strongest implementable completion of $\mathbb{C}$.

4. COMPARISONS OF SG$^+$ WITH OTHER FAIRNESS NOTIONS

In this section we compare SG$^+$ with the following well-known fairness notions and their completions:

Strong interaction fairness (SIF): An interaction that is infinitely often enabled is executed infinitely often.

Strong process fairness (SPF): A process that is infinitely often ready for an enabled interaction engages in an interaction infinitely often.

Weak process fairness (WPF): A process that is continuously ready for an enabled interaction (not necessary the same interaction) will eventually engage in an interaction.

Weak interaction fairness (WIF): An interaction that is continuously enabled will eventually be executed.

The comparison is intended to be comprehensive so that we know how these fairness notions differ for various systems. In particular, we shall divide the comparison into two subsections—one for systems involving strictly biparty interactions (à la CSP and Ada), and the other for those involving multiparty interactions of arbitrary arity. Recall that SG$^+$ is the strongest implementable and equivalence-robust fairness notion for systems where interactions cannot contain interactions. In the biparty case, an interaction x cannot be contained in another interaction y if $P_x \neq P_y$. This means that if interaction names only serve to identify the participants, then SG$^+$ is the strongest implementable and equivalence-robust fairness notion for biparty interaction systems.[3] We shall therefore use I^{B} to denote a set of biparty interactions such that $\forall x \in \mathsf{I}^{\mathsf{B}}, |P_x| = 2$ and $\forall y \in \mathsf{I}^{\mathsf{B}} - \{x\}, P_x \neq P_y$.

Recall from Lemma 2.2 that the above four fairness notions are all strongly feasible. Their equivalence-robustness is summarized in Table 1. It is clear that for every $\mathbb{IS}$ the following relation holds [1, 2]:

$$\mathrm{SIF}(\mathbb{IS}) \subseteq \mathrm{SPF}(\mathbb{IS}) \subseteq \mathrm{WPF}(\mathbb{IS}) \subseteq \mathrm{WIF}(\mathbb{IS}).$$

In particular, depending on the instances of $\mathbb{IS}$, the fairness notions may be identical or strictly different. To study the structure of interactions that distinguishes these fairness notions and their minimal and maximal completions, we shall associate $\mathbb{IS}$ with a program of type $\mathsf{M}^{\forall}$.

[3] In practice an interaction name usually identifies the set of participants, while the interaction body determines the content of communication, which can vary dynamically, and in some cases can even involve nondeterministic choices among guarded commands.

TABLE I

Equivalence-Robustness of Various Fairness Notions [1]

	Biparty interactions	Multiparty interactions
SIF	−	−
SPF	+	−
WPF	−	−
WIF	+	+

4.1. Biparty Interaction Systems

We will establish some lemmas that are useful in classifying the relationship between SG^+ and SIF, SPF, WPF, WIF, and their minimal and maximal completions. We begin with the comparison of SIF^-, SIF, and SIF^+. By definition of completions, for every $\mathbb{IS}$ we have $SIF^-(\mathbb{IS}) \subseteq SIF(\mathbb{IS}) \subseteq SIF^+(\mathbb{IS})$. Since SIF is not equivalence-robust, there exists some $\mathbb{IS}$ that distinguishes SIF^-, SIF, and SIF^+. The following lemma shows when they are distinct.

LEMMA 4.1. *For every* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^{\forall})$, $SIF^-(\mathbb{IS}) \subseteq SIF(\mathbb{IS}) \subseteq SIF^+(\mathbb{IS})$. *In particular,* $SIF^-(\mathbb{IS}) \subsetneq SIF(\mathbb{IS}) \subsetneq SIF^+(\mathbb{IS})$ *if* $\exists\, x, y, z \in \mathsf{I}$, $P_x \cap P_y \not\subseteq P_z$ *and* $P_x \cap P_z \not\subseteq P_y$.

Proof. To see the proper subset relation, consider the run

$$\pi = ((P_x - P_z) \cup P_y)(\, y\; P_z\; z\; P_y)^{\omega}.$$

If only y and z are enabled infinitely often in π, then $\pi \in SIF(\mathbb{IS})$. If some other interaction w is enabled infinitely often, then due to the restriction imposed on $\mathbb{IS}$, $w \neq x$ because x is never enabled in π. So either $P_w \subseteq (P_x - P_z) \cup P_y$ or $P_w \subseteq (P_x - P_y) \cup P_z$. If $P_w \subseteq (P_x - P_z) \cup P_y$, then let

$$\pi' = ((P_x - P_z) \cup P_y)(\, w\; P_w\; y\; P_z\; z\; P_y)^{\omega};$$

otherwise let

$$\pi' = ((P_x - P_z) \cup P_y)(\, y\; P_z\; w\; P_w\; z\; P_y)^{\omega}.$$

In either case, if π' is still not in $SIF(\mathbb{IS})$, then similarly there must be another interaction u such that either $P_u \subseteq (P_x - P_z) \cup P_y$ or $P_u \subseteq (P_x - P_y) \cup P_z$ and u is enabled infinitely often but is never executed. Then we can use the above method to obtain another run π'' such that u is executed infinitely often in π''. So without loss of generality assume that $\pi \in SIF(\mathbb{IS})$.

Consider the run

$$\rho = ((P_x - P_z) \cup P_y)((P_z - P_y)\, y\, (P_z \cap P_y)\, z\; P_y)^{\omega}.$$

It is easy to see that $\rho \equiv \pi$. However, ρ does not satisfy SIF because x is now enabled infinitely often in ρ but it is never executed. By the definition of minimal completions, $\pi \in SIF(\mathbb{IS}) - SIF^-(\mathbb{IS})$; and by the definition of maximal completions $\rho \in SIF^+(\mathbb{IS}) - SIF(\mathbb{IS})$. ■

Figures 4b and 4c illustrate some instances of $\mathbb{IS}$ for which $SIF^-(\mathbb{IS}) \subsetneq SIF(\mathbb{IS}) \subsetneq SIF^+(\mathbb{IS})$.[4]

LEMMA 4.2. *For every* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^{\forall})$, $SIF(\mathbb{IS}) \subseteq SPF(\mathbb{IS})$. *In particular,* $SIF(\mathbb{IS}) \subsetneq SPF(\mathbb{IS})$ *if* $\exists\, x, y_1, \ldots, y_n \in \mathsf{I}$ *such that* $\forall i \leq n$, $P_{y_i} \cap P_x \neq \emptyset$ *and* $\bigcup_i (P_{y_i} \cap P_x) = P_x$.

[4] It can also be shown that $SIF^-(\mathbb{IS}) \subsetneq SIF(\mathbb{IS}) \subsetneq SIF^+(\mathbb{IS})$ only if $\exists\, x, y, z \in \mathsf{I}$, $P_x \cap P_y \not\subseteq P_z$ and $P_x \cap P_z \not\subseteq P_y$. Since we do not need this result in the main theorems of this section, we omit the proof.

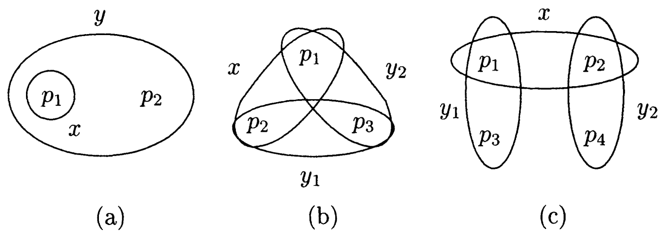


FIG. 4. Instances of interaction systems $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^{\forall})$ for which $\mathrm{SIF}(\mathbb{IS}) \subsetneq \mathrm{SPF}(\mathbb{IS})$. Moreover, SIF is not equivalence-robust for (b) and (c).

Proof. It is easy to see that $\forall\, \mathbb{IS}$, $\mathrm{SIF}(\mathbb{IS}) \subseteq \mathrm{SPF}(\mathbb{IS})$. To see the proper subset condition, consider the run

$$\pi = \left(P_{y_1} \cup \cdots \cup P_{y_n}\right)\left(y_1 P_{y_1} \cdots y_n P_{y_n}\right)^{\omega}.$$

Then $\pi \in \mathrm{SPF}(\mathbb{IS}) - \mathrm{SIF}(\mathbb{IS})$ because (1) x is enabled infinitely often (x is enabled because $P_x \subseteq P_{y_1} \cup \cdots \cup P_{y_n}$) but is never executed (so $\pi \notin \mathrm{SIF}(\mathbb{IS})$) and (2) every process in π executes some interaction y_i infinitely often (so $\pi \in \mathrm{SPF}(\mathbb{IS})$). ■

Figure 4 illustrates some instances of $\mathbb{IS}$ for which $\mathrm{SIF}(\mathbb{IS}) \subsetneq \mathrm{SPF}(\mathbb{IS})$.

We have shown the structure of $\mathbb{IS}$ that makes SIF non-equivalence-robust; that is, $\mathrm{SIF}^{-}(\mathbb{IS}) \subsetneq \mathrm{SIF}(\mathbb{IS}) \subsetneq \mathrm{SIF}^{+}(\mathbb{IS})$. Given that $\mathrm{SIF}(\mathbb{IS}) \subsetneq \mathrm{SPF}(\mathbb{IS})$, it is interesting to compare SPF with SIF^{+}. As we shall see shortly, SIF^{+}, SPF^{+}, and SG^{+} are all equivalent when interactions cannot contain interactions. Moreover, since in the biparty case SPF is equivalence-robust, it follows that SPF, SPF^{+}, SIF^{+}, and SG^{+} are all equivalent.

LEMMA 4.3. *For every* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M})$, *if* $\forall x, y \in \mathsf{I}$, $x \neq y \Rightarrow P_x \not\subseteq P_y$, *then* $\mathrm{SIF}(\mathbb{IS}) \subseteq \mathrm{SG}^{+}(\mathbb{IS})$, *and* $\mathrm{SPF}(\mathbb{IS}) \subseteq \mathrm{SG}^{+}(\mathbb{IS})$.

Proof. This follows directly from Lemma 3.3, and note that the proof of Lemma 3.3 can also be used to show that every run $\pi \in \mathrm{SPF}(\mathbb{IS})$ is equivalent to a singular run. ■

LEMMA 4.4. *For every* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M})$, *if* $\forall x, y \in \mathsf{I}$, $x \neq y \Rightarrow P_x \not\subseteq P_y$, *then* $\mathrm{SIF}^{+}(\mathbb{IS}) = \mathrm{SPF}^{+}(\mathbb{IS}) = \mathrm{SG}^{+}(\mathbb{IS})$.

Proof. By Lemma 4.3 and the fact that SG^{+} is equivalence-robust, $\mathrm{SIF}^{+}(\mathbb{IS}) \subseteq \mathrm{SG}^{+}(\mathbb{IS})$ and $\mathrm{SPF}^{+}(\mathbb{IS}) \subseteq \mathrm{SG}^{+}(\mathbb{IS})$. On the other hand, by Lemmas 2.2 and 2.1, SIF^{+} and SPF^{+} are strongly feasible and equivalence-robust. By Theorem 3.4, $\mathrm{SG}^{+}(\mathbb{IS}) \subseteq \mathrm{SIF}^{+}(\mathbb{IS})$ and $\mathrm{SG}^{+}(\mathbb{IS}) \subseteq \mathrm{SPF}^{+}(\mathbb{IS})$. Hence $\mathrm{SIF}^{+}(\mathbb{IS}) = \mathrm{SPF}^{+}(\mathbb{IS}) = \mathrm{SG}^{+}(\mathbb{IS})$. ■

We now consider the notion of WPF and its completions.

LEMMA 4.5. *For every* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^{\forall})$, $\mathrm{SPF}^{-}(\mathbb{IS}) \subseteq \mathrm{WPF}^{-}(\mathbb{IS})$. *In particular*, $\mathrm{SPF}^{-}(\mathbb{IS}) \subsetneq \mathrm{WPF}^{-}(\mathbb{IS})$ *if* $\exists\, x, y \in \mathsf{I}$ *such that* (1) $P_x \cap P_y \neq \emptyset$, *and* (2) $\exists\, p \in P_x - P_y$ *such that* $\forall z \in \mathsf{I}$, $P_z \subseteq P_x - P_y \Rightarrow p \notin P_z$.

Proof. Since $\mathrm{SPF}(\mathbb{IS}) \subseteq \mathrm{WPF}(\mathbb{IS})$, $\mathrm{SPF}^{-}(\mathbb{IS}) \subseteq \mathrm{WPF}^{-}(\mathbb{IS})$. To see the proper subset condition, without loss of generality assume that x and y are two interactions satisfying conditions (1) and (2) of the lemma such that for all other interactions x' and y' satisfying the same conditions, $|P_x \cup P_y| \leq |P_{x'} \cup P_{y'}|$.

Consider the run

$$\pi = (P_x \cup P_y)(y P_y)^{\omega}.$$

Let p be the process satisfying condition (2) of the lemma. Since $p \in P_x - P_y$ and since x is enabled infinitely often, π does not satisfy SPF. So π does not satisfy SPF^{-} either. Moreover, if some interaction

z_1 is continuously enabled in π, then $P_{z_1} \subseteq P_x - P_y$. By condition (2) of the lemma, $p \notin P_{z_1}$. Then in the run

$$\pi' = (P_x \cup P_y)\big(yz_1 P_y P_{z_1}\big)^\omega$$

z_1 is not continuously enabled. Still, π' does not satisfy SPF because p is ready for an enabled interaction (i.e., x) infinitely often but it never engages in any interaction. So π' does not satisfy SPF$^-$ either. If some other interaction z_2 is still continuously enabled in π', then $P_{z_2} \subseteq P_x - P_y - P_{z_1}$ and $p \notin P_{z_2}$. The above method can be used again to obtain a run π'' such that π'' does not satisfy SPF and z_2 is not continuously enabled. As there are only a finite number of interactions in I, without loss of generality let $\rho = (P_x \cup P_y)(yz_1z_2 \dots z_k P_y P_{z_1} P_{z_2} \dots P_{z_k})^\omega$ be a run that does not satisfy SPF and SPF$^-$ and that does not have an interaction that is continuously enabled. Note that by the construction, for every two different interactions a, b executed in ρ, $P_a \cap P_b = \emptyset$.

We argue that all runs in $eq(\rho)$ satisfy WPF. To see this, suppose otherwise some run ψ in $eq(\rho)$ violates WPF. Then, given that no interaction is continuously enabled in ρ (and thus in ψ), there must exist two interactions u_1 and v_1, where $P_{u_1} \cup P_{v_1} \subseteq P_x \cup P_y$, such that (a) $P_{u_1} \cap P_{v_1} \neq \emptyset$, (b) v_1 is executed infinitely often in ψ, and (c) some process $q \in P_{u_1} - P_{v_1}$ is continuously ready for an enabled interaction but it never executes any interaction. Moreover, immediately after v_1 is executed q must still be ready for an enabled interaction. Let u_2 be the smallest such interaction so that there is no interaction a such that $P_a \subsetneq P_{u_2}$. Because no interaction is continuously enabled in ρ, u_2 will subsequently be disabled due to the execution of some interaction v_2. So $P_{v_1} \cap P_{u_2} = \emptyset$ and $P_{v_2} \cap P_{u_2} \neq \emptyset$. Since any two different interactions executed in ρ are disjoint, $P_{v_1} \cap P_{v_2} = \emptyset$. Given that $P_{u_1}, P_{u_2}, P_{v_1}, P_{v_2} \subseteq P_x \cup P_y$, we have $P_{u_2} \cup P_{v_2} \subsetneq P_x \cup P_y$.

However, because q is not involved in an interaction a such that $P_a \subsetneq P_{u_2}$, it is not involved in any interaction b such that $P_b \subseteq P_{u_2} - P_{v_2}$. Then u_2 and v_2 satisfy the lemma conditions on $\mathbb{IS}$; but this then contradicts our earlier assumption that $|P_x \cup P_y| \leq |P_{u_2} \cup P_{v_2}|$. Therefore, all runs in $eq(\rho)$ satisfy WPF. Hence $\rho \in$ WPF$^-(\mathbb{IS}) -$ SPF$^-(\mathbb{IS})$. ■

Figure 5 depicts some instances of $\mathbb{IS}$ for which SPF$^-(\mathbb{IS}) \subsetneq$ WPF$^-(\mathbb{IS})$. The following lemma on the non-equivalence-robustness of WPF is somewhat complex.

LEMMA 4.6. *For every* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^\forall)$, WPF$^-(\mathbb{IS}) \subseteq$ WPF$(\mathbb{IS}) \subseteq$ WPF$^+(\mathbb{IS})$. *In particular,* WPF$^-(\mathbb{IS}) \subsetneq$ WPF$(\mathbb{IS}) \subsetneq$ WPF$^+(\mathbb{IS})$ *if* $\exists x_i, y_i \in \mathsf{I}$, *where* $0 \leq i \leq n-1$ *and* $n > 1$, *and* $\exists p \in \mathsf{P}$ *such that* (1) $p \in \bigcap P_{x_i}$, $p \notin \bigcup P_{y_i}$, (2) $\forall i$, $P_{y_i} \cap P_{x_i} \neq \emptyset$, $P_{y_i} \cap P_{x_{i+1}} = \emptyset$, *and* (3) $\forall u \in \mathsf{I}$, $P_u \subseteq \bigcup P_{x_i} - \bigcup P_{y_i} \Rightarrow \exists v \in \mathsf{I}$, $P_v \cap P_u \neq \emptyset$, $p \notin P_v$, *and* $\exists i$, $P_v \cap P_{x_i} = \emptyset$. *(Note that in the lemma additions and subtractions on indices of x and y are to be interpreted modulo n).*

Proof. To see the proper subset conditions, consider the run

$$\pi = \left(\bigcup P_{x_i} \cup \bigcup P_{y_i}\right)\left(y_0 P_{y_0} y_1 P_{y_1} \dots y_{n-1} P_{y_{n-1}}\right)^\omega.$$

Observe that before each instance of y_j process p is ready for all x_i's; and since $P_{y_j} \cap P_{x_{j+1}} = \emptyset$, after the instance p is ready for at least x_{j+1} (which exists because $n > 1$). So p is continuously ready for an enabled interaction (starting from the point the first interaction is to be executed). Since p never executes any interaction, $\pi \notin$ WPF$(\mathbb{IS})$. Now consider the run

$$\rho = \left(\bigcup P_{x_i} - \bigcup P_{y_i}\right)\left(P_{y_0} y_0 P_{y_1} y_1 \dots P_{y_{n-1}} y_{n-1}\right)^\omega$$

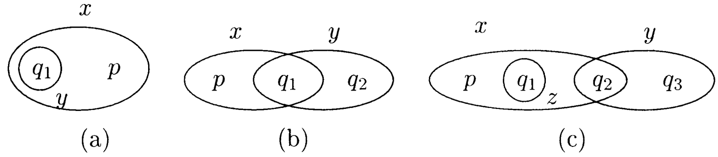


FIG. 5. Instances of interaction systems $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^\forall)$ for which SPF$^-(\mathbb{IS}) \subsetneq$ WPF$^-(\mathbb{IS})$.

which is obtained from π by deferring for each j the readiness of $\bigcup P_{y_i} - P_{y_j}$ before each instance of y_j until the instance is executed. Since the deferred actions are independent of the instance of y_j, and since no action is deferred indefinitely, ρ is equivalent to π. However, since $\forall i,\ P_{y_i} \cap P_{x_i} \neq \emptyset$, right after each instance of y_j none of the x_i's is enabled. So the x_i's can no longer cause p to be continuously ready for an enabled interaction.

If $\rho \in \mathrm{WPF}(\mathbb{IS})$ then we are done because $\rho \in \mathrm{WPF}(\mathbb{IS}) - \mathrm{WPF}^-(\mathbb{IS})$, while $\pi \in \mathrm{WPF}^+(\mathbb{IS}) - \mathrm{WPF}(\mathbb{IS})$. If $\rho \notin \mathrm{WPF}(\mathbb{IS})$, then some process is still continuously ready for an enabled interaction but it never engages in any interaction. Observe that after each instance of y_j only the set of processes $\bigcup P_{x_i} - \bigcup P_{y_i}$ are ready for interaction. Moreover, the processes are continuously ready for interaction throughout ρ. So if some process is continuously ready for an enabled interaction but never engages in any interaction, then the process must be continuously ready for the same interaction, say u, and $P_u \subseteq \bigcup P_{x_i} - \bigcup P_{y_i}$. By condition (3) there exists some v (where v could be u) and some k such that $P_v \cap P_u \neq \emptyset$, $p \notin P_v$, and $P_v \cap P_{x_k} = \emptyset$.

Let π' and ρ' be two equivalent runs given by

$$\pi' = \left(\bigcup P_{x_i} \cup \bigcup P_{y_i} \cup P_v\right) \left(y_0 P_{y_0} y_1 P_{y_1} \ldots y_{n-1} P_{y_{n-1}} v\, P_v\right)^\omega$$

$$\rho' = \left(\bigcup P_{x_i} - \bigcup P_{y_i} - P_v\right) \left(P_{y_0} y_0 P_{y_1} y_1 \ldots P_{y_{n-1}} y_{n-1} P_v\, v\right)^\omega .$$

Since $P_v \cap P_{x_k} = \emptyset$, p is still continuously ready for an enabled interaction in π' even if v is executed infinitely often. So $\pi' \notin \mathrm{WPF}(\mathbb{IS})$. Moreover, since $P_u \cap P_v \neq \emptyset$, u is not continuously enabled in ρ'. So either $\rho' \in \mathrm{WPF}(\mathbb{IS})$, in which case we are done, or there exists another u', $P_{u'} \subseteq \bigcup P_{x_i} - \bigcup P_{y_i} - P_v$, such that u' is continuously enabled in ρ'. In the latter case, we can apply the above method again to obtain two equivalent runs π'' and ρ'' such that $\pi'' \notin \mathrm{WPF}(\mathbb{IS})$ and u' (and u) are not continuously enabled in ρ''. Given that there are only a finite number of interactions and a finite number of processes in $\mathbb{IS}$, eventually we will establish the lemma. ■

Figure 6 depicts some instances of $\mathbb{IS}$ for which $\mathrm{WPF}^-(\mathbb{IS}) \subsetneq \mathrm{WPF}(\mathbb{IS}) \subsetneq \mathrm{WPF}^+(\mathbb{IS})$. Note that the non-equivalence-robustness must be intrigued by at least four interactions. In Fig. 6c, all the six interactions are needed in making p_1 be continuously ready for an enabled interaction but never engage in any interaction.

LEMMA 4.7. *For every* $\mathbb{IS}$, $\mathrm{WPF}^+(\mathbb{IS}) = \mathrm{WIF}(\mathbb{IS})$.

Proof. We shall show that for every $\pi \in \mathrm{run}(\mathbb{IS})$ it is never the case that $eq(\pi) \subseteq \mathrm{WIF}(\mathbb{IS}) - \mathrm{WPF}(\mathbb{IS})$. Since both WPF^+ and WIF are equivalence-robust, and since $\mathrm{WPF}(\mathbb{IS}) \subseteq \mathrm{WPF}^+(\mathbb{IS}) \subseteq \mathrm{WIF}(\mathbb{IS})$, we therefore have $\mathrm{WPF}^+(\mathbb{IS}) = \mathrm{WIF}(\mathbb{IS})$.

Let ψ be any run in $\mathrm{WIF}(\mathbb{IS}) - \mathrm{WPF}(\mathbb{IS})$. Then, there must exist a process p such that from some point onward (say t_0) p is continuously ready for an enabled interaction, but p never executes any interaction thereafter. Moreover, since ψ satisfies WIF, p cannot be ready for the same interaction continuously.

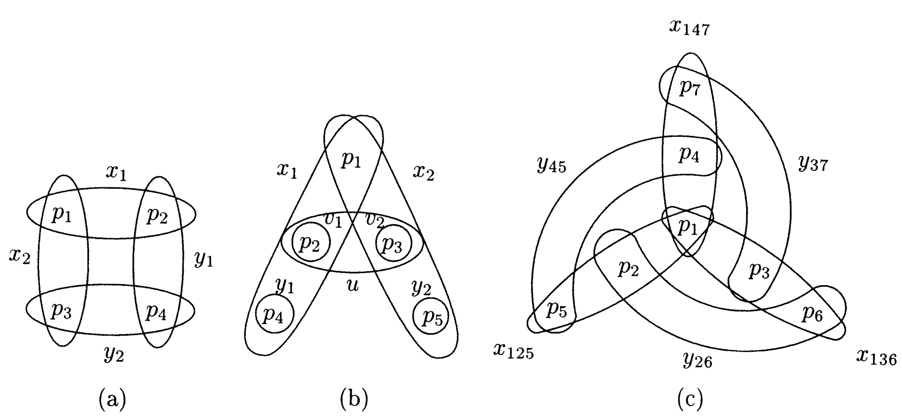


FIG. 6. Instances of $\mathbb{IS}$ for which $\mathrm{WPF}(\mathbb{IS})$ is not equivalence-robust.

So there exists an infinite number of points $t_1, t_2 \ldots$ such that (1) p is continuously ready for some interaction x_i at the time between t_{i-1} and t_i (inclusive), (2) x_i becomes enabled at some point between t_{i-2} and t_{i-1}, and is disabled at t_i, and (3) $x_i \neq x_{i+1}$ for each $i > 0$. Suppose that x_i is disabled due to the execution of some interaction y_i. Since x_{i+1} remains enabled while y_i is executed, $P_{x_{i+1}} \cap P_{y_i} = \emptyset$. That is, the enabledness of x_{i+1} (due to the readiness of some processes in $P_{x_{i+1}}$) is independent of the execution of y_i.

Consider the run ψ' obtained from ψ by deferring, for each i, the enabledness of x_{i+1} until y_i is executed. Then, $\psi \equiv \psi'$. Note that the transformation from ψ to ψ' does not cause any new interaction to be enabled, nor does it extend the duration of an interaction's enabledness. So the transformation cannot cause any new process to be continuously ready for an enabled interaction. However, for each i, right after y_i is executed in ψ', p is not ready for x_{i+1} (and not ready for x_i either). If there exist infinitely many i's such that p is not ready for any interaction immediately after each y_i is executed, then p is not continuously ready for an enabled interaction. Otherwise, there exists some i_0 such that for all $i \geq i_0$ there still exists another interaction x'_{i+1} which remains enabled right after y_i is executed. We can also use the above method again to break the overlap of the enabledness of x_i and x'_{i+1}. Since there is only a finite number of interactions, we can obtain a run equivalent to ψ such that p is not continuously ready for an enabled interaction.

Similarly, if there is some other process q in ψ' (and thus in ψ) that is continuously ready for an enabled interaction, then we can use the same method again to transform ψ' into ψ'' so that q is not continuously ready for an enabled interaction. Since there are only a finite number of processes, we can transform ψ into an equivalent run satisfying WPF. Therefore, for every run $\psi \in \mathrm{WIF}(\mathbb{IS}) - \mathrm{WPF}(\mathbb{IS})$, $eq(\psi) \cap \mathrm{WPF}(\mathbb{IS}) \neq \emptyset$. ■

We have finished the comparison of SG^+ with the four fairness notions SIF, SPF, WPF, and WIF, and their minimal/maximal completions. The following theorem summarizes the results.

THEOREM 4.8. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}^{\mathsf{B}}, \mathsf{M}^{\forall})$ *be a given biparty interaction system. Then the following relation holds*:

$$\mathrm{SIF}^-(\mathbb{IS}) \subseteq \mathrm{SIF}(\mathbb{IS}) \subseteq \mathrm{SIF}^+(\mathbb{IS}) = \mathrm{SG}^+(\mathbb{IS}) = \mathrm{SPF}^-(\mathbb{IS}) = \mathrm{SPF}(\mathbb{IS}) = \mathrm{SPF}^+(\mathbb{IS})$$
$$\subseteq \mathrm{WPF}^-(\mathbb{IS}) \subseteq \mathrm{WPF}(\mathbb{IS}) \subseteq \mathrm{WPF}^+(\mathbb{IS}) = \mathrm{WIF}(\mathbb{IS}).$$

In particular, there exists an $\mathbb{IS}$ *for which all the stronger-than relations "⊆" become strict.*

Figure 7a depicts the relationship between these fairness notions. In this figure $A \to B$ means A is stronger than B. The relation "→" is transitive. Note that since SG^+ is the strongest implementable and

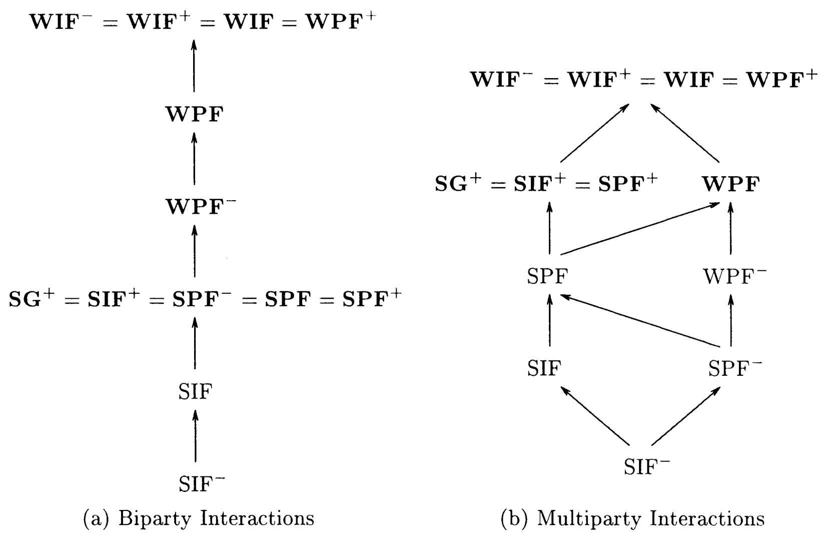


FIG. 7. The hierarchy of various fairness notions when interactions cannot contain interactions. The fairness notions in boldface are implementable, while the others are not.

equivalence-robust fairness notion, all fairness notions weaker than SG^+ are also implementable, and all equivalence-robust fairness notions stronger than SG^+, e.g., SIF^-, are unimplementable. Moreover, although SIF is not equivalence-robust, as we have seen in Part I, it is also unimplementable.

EXAMPLE. Consider the Producers–Consumers problem, in which there are two producers $producer_1$ and $producer_2$ and two consumers $consumer_1$ and $consumer_2$. The data produced by a producer can be consumed by either of the consumers. The following is a CSP program for the problem, where $i = 1, 2$:

$$
\begin{aligned}
producer_i &:: compute(data);\\
&\quad * [\ consumer_1\ !\ data \longrightarrow compute(data)\\
&\quad \Box\ \ consumer_2\ !\ data \longrightarrow compute(data)\ ]
\end{aligned}
$$

$$
\begin{aligned}
consumer_i &:: * [\ producer_1\ ?\ data \longrightarrow digest(data)\\
&\quad\ \Box\ \ producer_2\ ?\ data \longrightarrow digest(data)\ ]
\end{aligned}
$$

There are four biparty interactions in this program, each of which involves a producer and a consumer. They have the structure shown in Fig. 6a. So by our results in this section, the stronger-than relations (i.e., the arrow $\rightarrow$) in Fig. 7a are all strict; that is, for this problem

$$SIF^-(\mathbb{IS}) \subsetneq SIF(\mathbb{IS}) \subsetneq SG^+(\mathbb{IS}) \subsetneq WPF^-(\mathbb{IS}) \subsetneq WPF(\mathbb{IS}) \subsetneq WIF(\mathbb{IS}).$$

Therefore, any implementation of CSP's input/output guards which guarantees only WIF cannot prevent the following behavior, which continuously blocks $producer_1$ from sending its data to either consumer and so does not satisfy WPF (although it does satisfy WIF):

all processes are ready (for communication/interaction), and then the repeat of the following forever
$producer_2$ sends data to $consumer_1$
$producer_2$ and $consumer_1$ ready
$producer_2$ sends data to $consumer_2$
$producer_2$ and $consumer_2$ ready
...

Similarly, the following behavior which satisfies WIF and WPF but does not satisfy WPF^- and SPF is also possible:

all processes are ready, and then the repeat of the following forever
$producer_2$ sends data to $consumer_1$
$producer_2$ ready
$producer_2$ sends data to $consumer_2$
$producer_2$ ready
$consumer_1$ and $consumer_2$ ready
...

In the absence of $consumer_1$ (say, it terminates prematurely), the following scenario which satisfies WIF, WPF, and WPF^- but not SPF is also possible:

all processes are ready, and then the repeat of the following forever
$producer_2$ sends data to $consumer_2$
$producer_2$ and $consumer_2$ ready
...

Since in the biparty case SPF (which is equivalent to SG^+) is implementable, a good implementation should be able to avoid all the above unfair scenarios. On the other hand, since SPF is also the strongest implementable and equivalence-robust fairness notion, the strongest equivalence-robust property one can observe from the program executing in any asynchronous environment is that no process is forever blocked from communicating with its partners if it has infinitely many such opportunities.

4.2. Multiparty Interaction Systems

We now consider multiparty interactions of arbitrary arity. Recall that in Theorem 4.8 the following relations hold as well even if interactions are multipartied (but cannot contain interactions):

1. $\mathrm{SIF}^-(\mathbb{IS}) \subseteq \mathrm{SIF}(\mathbb{IS}) \subseteq \mathrm{SIF}^+(\mathbb{IS}) = \mathrm{SG}^+(\mathbb{IS}) = \mathrm{SPF}^+(\mathbb{IS})$
2. $\mathrm{SIF}(\mathbb{IS}) \subseteq \mathrm{SPF}(\mathbb{IS})$
3. $\mathrm{SPF}^-(\mathbb{IS}) \subseteq \mathrm{WPF}^-(\mathbb{IS}) \subseteq \mathrm{WPF}(\mathbb{IS}) \subseteq \mathrm{WPF}^+(\mathbb{IS}) = \mathrm{WIF}(\mathbb{IS})$
4. $\mathrm{SPF}^+(\mathbb{IS}) \subseteq \mathrm{WPF}^+(\mathbb{IS})$.

Since SPF becomes non-equivalence-robust in the multiparty case, we need to resolve the relationship between its completions and the other fairness notions.

LEMMA 4.9. *For every* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^\forall)$, $\mathrm{SPF}^-(\mathbb{IS}) \subseteq \mathrm{SPF}(\mathbb{IS}) \subseteq \mathrm{SPF}^+(\mathbb{IS})$. *In particular*, $\mathrm{SPF}^-(\mathbb{IS}) \subsetneq \mathrm{SPF}(\mathbb{IS}) \subsetneq \mathrm{SPF}^+(\mathbb{IS})$ *if* $\exists x, y, z \in \mathsf{I}, \exists p_1, p_2, p_3 \in P_x$ *such that* (1) $p_1 \in P_x - P_y - P_z$, $p_2 \in P_z - P_y$ *and* $p_3 \in P_y - P_z$ *and* (2) $\forall u \in \mathsf{I}$, $p_1 \in P_u \Rightarrow (P_u \not\subseteq (P_x - P_z) \cup P_y)$ *and* $(P_u \not\subseteq (P_x - P_y) \cup P_z)$.

Proof. To see the proper subset conditions, it suffices to find a run $\pi \in \mathrm{SPF}(\mathbb{IS})$ such that some run equivalent to π does not satisfy SPF. Consider the run

$$\pi = ((P_x - P_z) \cup P_y)(y P_z z P_y)^\omega.$$

Due to the two conditions $p_2 \in P_x \cap P_z - P_y$ and $p_3 \in P_x \cap P_y - P_z$, x is never enabled in π. So if no other interaction is enabled infinitely often but is never executed, then $\pi \in \mathrm{SPF}(\mathbb{IS})$. Moreover, the run

$$\rho = ((P_x - P_z) \cup P_y)((P_z - P_y)\, y\, (P_z \cap P_y)\, z P_y)^\omega$$

is equivalent to π but x is now enabled just before each instance of y is to be executed. So p_1, which belongs to P_x, is now ready for an enabled interaction infinitely often. Since p_1 never executes any interaction, $\rho \notin \mathrm{SPF}(\mathbb{IS})$.

If $\pi \notin \mathrm{SPF}(\mathbb{IS})$, then some interaction w is enabled infinitely often but some process q in P_w never executes any interaction. Then either $P_w \subseteq (P_x - P_y) \cup P_z$ or $P_w \subseteq (P_x - P_z) \cup P_y$. Due to condition (2) imposed on $\mathbb{IS}$, $p_1 \notin P_w$. Assume that $P_w \subseteq (P_x - P_y) \cup P_z$. (The other case can be proved similarly.) Then in the run

$$\pi' = ((P_x - P_z) \cup P_y)(y P_z w P_w z P_y)^\omega$$

q has executed w infinitely often. So either $\pi' \in \mathrm{SPF}(\mathbb{IS})$, or similarly there exists another interaction u, $u \neq w$, such that u is enabled infinitely often but some process in P_u never executes any interaction. In the former case, we can find a run ρ' similar to ρ such that $\rho' \equiv \pi'$ but ρ' does not satisfy SPF because p_1 is infinitely often ready for an enabled interaction (i.e., x) but it never engages in any interaction. In the latter case, given that I and P are finite, we can use the above method repeatedly to find two equivalent runs such that one satisfies SPF while the other does not. ■

Figure 8 illustrates some instances of $\mathbb{IS}$ for which SPF is not equivalence-robust. Note that all of them consists of a multiparty interaction involving more than two processes.

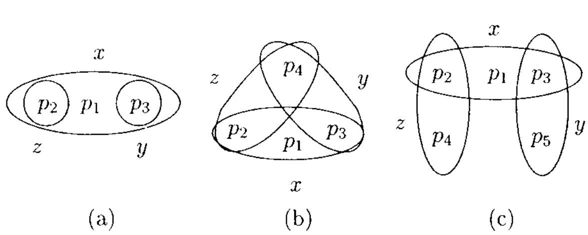


FIG. 8. Instances of interaction systems for which SPF is not equivalence-robust.

LEMMA 4.10. *For every* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^\forall)$, $\mathrm{SIF}^-(\mathbb{IS}) \subseteq \mathrm{SPF}^-(\mathbb{IS})$. *In particular,* $\mathrm{SIF}^-(\mathbb{IS}) \subsetneq \mathrm{SPF}^-(\mathbb{IS})$ *if* $\exists x, y_1, \ldots, y_n \in \mathsf{I}$ *such that* $\forall i \le n$, $P_{y_i} \cap P_x \neq \emptyset$ *and* $\bigcup_i (P_{y_i} \cap P_x) = P_x$.

Proof. Since $\mathrm{SIF}(\mathbb{IS}) \subseteq \mathrm{SPF}(\mathbb{IS})$, $\mathrm{SIF}^-(\mathbb{IS}) \subseteq \mathrm{SPF}^-(\mathbb{IS})$. To see the proper subset condition, recall the following run in Lemma 4.2,

$$\pi = \left(P_{y_1} \cup \cdots \cup P_{y_n}\right)\left(y_1 P_{y_1} \cdots y_n P_{y_n}\right)^\omega,$$

which is in $\mathrm{SPF}(\mathbb{IS}) - \mathrm{SIF}(\mathbb{IS})$. So $\pi \notin \mathrm{SIF}^-(\mathbb{IS})$ either. Since every process in π executes some interaction y_i infinitely often, $eq(\pi) \subseteq \mathrm{SPF}^-(\mathbb{IS})$. So $\pi \in \mathrm{SPF}^-(\mathbb{IS}) - \mathrm{SIF}^-(\mathbb{IS})$. ∎

Figure 4 also illustrates some instances of $\mathbb{IS}$ for which $\mathrm{SIF}^-(\mathbb{IS}) \subsetneq \mathrm{SPF}^-(\mathbb{IS})$.

LEMMA 4.11. *There exists a system* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^\forall)$, *where* $\forall x, y \in \mathsf{I}$, $x \neq y \Rightarrow P_x \not\subseteq P_y$, *such that*

$$\mathrm{SIF}(\mathbb{IS}) \not\subseteq \mathrm{SPF}^-(\mathbb{IS}) \qquad \textit{and} \qquad \mathrm{SPF}^-(\mathbb{IS}) \not\subseteq \mathrm{SIF}(\mathbb{IS}).$$

Proof. Let $\mathbb{IS}$ be the interaction system with P and I shown in Fig. 9. Consider the run

$$\pi_1 = p_5(p_1 p_3 x_{13} p_2 p_4 x_{24})^\omega.$$

Then $\pi_1 \in \mathrm{SIF}(\mathbb{IS})$. However, $\pi_1 \notin \mathrm{SPF}^-(\mathbb{IS})$ because its equivalent run

$$\pi_2 = p_5(p_1 p_3 p_2 p_4 x_{13} x_{24})^\omega$$

does not satisfy SPF due to the fact that p_5 is now ready for an enabled interaction (i.e., x_{345}) infinitely often but it never executes any interaction. So $\mathrm{SIF}(\mathbb{IS}) \not\subseteq \mathrm{SPF}^-(\mathbb{IS})$.

On the other hand, the run

$$\rho = (p_1 p_3 p_2 p_4 x_{13} x_{24})^\omega$$

and all of its equivalent runs satisfy SPF, and so they also satisfy SPF^-. However, $\rho \notin \mathrm{SIF}(\mathbb{IS})$ because x_{12} is enabled infinitely often but is never executed. So $\mathrm{SPF}^-(\mathbb{IS}) \not\subseteq \mathrm{SIF}(\mathbb{IS})$. ∎

LEMMA 4.12. *There exists a system* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^\forall)$, *where* $\forall x, y \in \mathsf{I}$, $x \neq y \Rightarrow P_x \not\subseteq P_y$, *such that*

1. $\mathrm{SG}^+(\mathbb{IS}) \not\subseteq \mathrm{WPF}(\mathbb{IS})$ *and* $\mathrm{WPF}(\mathbb{IS}) \not\subseteq \mathrm{SG}^+(\mathbb{IS})$ *and*
2. $\mathrm{SG}^+(\mathbb{IS}) \not\subseteq \mathrm{WPF}^-(\mathbb{IS})$ *and* $\mathrm{WPF}^-(\mathbb{IS}) \not\subseteq \mathrm{SG}^+(\mathbb{IS})$.

Proof. Let $\mathbb{IS}$ be the interaction system with P and I shown in Fig. 10. Consider the two runs

$$\pi_1 = p_5(p_1 p_3 x_{13} p_2 p_4 x_{24} p_6 p_8 x_{68} p_7 p_9 x_{79})^\omega$$
$$\pi_2 = p_5 p_1 p_3 p_2 p_4 p_6 p_8 p_7 p_9 (x_{13} p_1 p_3 x_{24} p_2 p_4 x_{68} p_6 p_8 x_{79} p_7 p_9)^\omega.$$

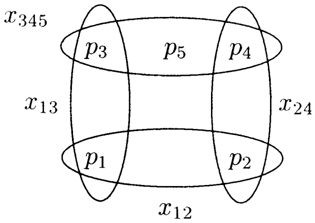


FIG. 9. An interaction system for which $\mathrm{SIF}(\mathbb{IS}) \not\subseteq \mathrm{SPF}^-(\mathbb{IS})$ and $\mathrm{SPF}^-(\mathbb{IS}) \not\subseteq \mathrm{SIF}(\mathbb{IS})$.

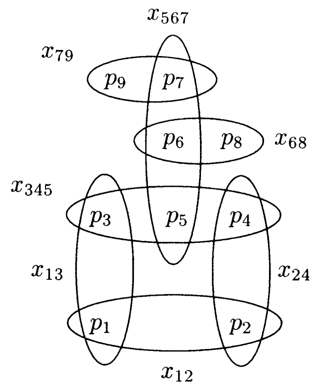


FIG. 10. An interaction system for which WPF($\mathbb{IS}$) $\not\subseteq$ SG$^+$($\mathbb{IS}$) and SG$^+$($\mathbb{IS}$) $\not\subseteq$ WPF($\mathbb{IS}$).

Observe that π_1 is singular and $\pi_1 \equiv \pi_2$. So $\pi_2 \in$ SG$^+$($\mathbb{IS}$). However, $\pi_2 \notin$ WPF($\mathbb{IS}$) because p_5 is continuously ready for an enabled interaction but p_5 never executes any interaction. So SG$^+$($\mathbb{IS}$) $\not\subseteq$ WPF($\mathbb{IS}$). Moreover, π_2 does not satisfy WPF$^-$ either. So SG$^+$($\mathbb{IS}$) $\not\subseteq$ WPF$^-$($\mathbb{IS}$).

On the other hand, the run

$$\rho = p_3(p_1 p_2 x_{12})^\omega$$

satisfies WPF. Moreover, all of its equivalent runs also satisfy WPF but none of them is singular. So $\rho \in$ WPF$^-$($\mathbb{IS}$) and $\rho \notin$ SG$^+$($\mathbb{IS}$). Hence WPF($\mathbb{IS}$) $\not\subseteq$ SG$^+$($\mathbb{IS}$), and WPF$^-$($\mathbb{IS}$) $\not\subseteq$ SG$^+$($\mathbb{IS}$). ■

It can be seen that for Lemma 4.12 to hold, I must contain interactions involving more than two processes. (Recall that in the biparty case SG$^+$($\mathbb{IS}$) $\subseteq$ WPF$^-$($\mathbb{IS}$) $\subseteq$ WPF($\mathbb{IS}$).)

LEMMA 4.13. *There exists a system* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^\forall)$, *where* $\forall x, y \in \mathsf{I}$, $x \neq y \Rightarrow P_x \not\subseteq P_y$, *such that*

1. SPF($\mathbb{IS}$) $\not\subseteq$ WPF$^-$($\mathbb{IS}$) *and* WPF$^-$($\mathbb{IS}$) $\not\subseteq$ SPF($\mathbb{IS}$) *and*
2. SIF($\mathbb{IS}$) $\not\subseteq$ WPF$^-$($\mathbb{IS}$) *and* WPF$^-$($\mathbb{IS}$) $\not\subseteq$ SIF($\mathbb{IS}$).

Proof. The example presented in Lemma 4.12 can be used to establish the lemma; we omit the details. ■

We now summarize the results in the following theorem. A pictorial representation of the comparison is given in Fig. 7b. Recall that the stronger-than relation "→" is transitive. So two fairness notions are incomparable if there is no path connecting them.

THEOREM 4.14. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^\forall)$ *be a given multiparty interaction system such that* $\forall x, y \in \mathsf{I}$, $x \neq y \Rightarrow P_x \not\subseteq P_y$. *Then the following relations hold*:

1. SIF$^-$($\mathbb{IS}$) $\subseteq$ SIF($\mathbb{IS}$) $\subseteq$ SPF($\mathbb{IS}$) $\subseteq$ SG$^+$($\mathbb{IS}$) $=$ SIF$^+$($\mathbb{IS}$) $=$ SPF$^+$($\mathbb{IS}$) $\subseteq$ WIF($\mathbb{IS}$) $=$ WPF$^+$($\mathbb{IS}$)
2. SIF$^-$($\mathbb{IS}$) $\subseteq$ SPF$^-$($\mathbb{IS}$) $\subseteq$ WPF$^-$($\mathbb{IS}$) $\subseteq$ WPF($\mathbb{IS}$) $\subseteq$ WIF($\mathbb{IS}$)
3. SPF$^-$($\mathbb{IS}$) $\subseteq$ SPF($\mathbb{IS}$) $\subseteq$ WPF($\mathbb{IS}$).

In particular, *there exists an* $\mathbb{IS}$ *for which all the above stronger-than relations "*$\subseteq$*" become strict. On the other hand*, *there exists some* $\mathbb{IS}$ *such that the following relations hold*:

1. SIF($\mathbb{IS}$) *is incomparable with* SPF$^-$($\mathbb{IS}$)
2. SG$^+$($\mathbb{IS}$) *is incomparable with* WPF($\mathbb{IS}$)
3. SIF($\mathbb{IS}$), SPF($\mathbb{IS}$), *and* SG$^+$($\mathbb{IS}$) *are incomparable with* WPF$^-$($\mathbb{IS}$).

Note that although WPF is incomparable with SG$^+$, it is implementable for all interaction systems (see Part I). This, however, does not contradict Theorem 3.4 (that SG$^+$ is the strongest implementable and equivalence-robust fairness notion) because WPF is not equivalence-robust. In fact, as we shall see in Section 5, there exists an implementable (but not equivalence-robust) fairness notion that is no weaker than SG^+, but there is no strongest implementable fairness notion for virtually all interaction systems.

Moreover, WPF^- is also incomparable with SG^+. Since WPF^- is equivalence-robust, by Theorem 3.4, WPF^- must not be strongly feasible. Indeed, WPF^- does exclude some singular runs from some systems; see Lemma 4.12. Also noteworthy is that SPF becomes unimplementable in the multiparty case. By Lemma 2.6, SPF^- is unimplementable too.

EXAMPLE. Consider the Dining Philosophers problem. We can define a multiparty interaction *eating_session*$_i$ involving the ith philosopher and its two neighboring chopsticks, where $0 \leq i \leq 4$.

The philosopher processes and the chopstick processes execute the following program:

philosopher$_i$ :: * [*hungry*; *eating_session*$_i$ $\longrightarrow$ *thinking*]

chopstick$_i$ :: * [*eating_session*$_i$ $\longrightarrow$ *clean_chopstick*
$\square$ *eating_session*$_{i-1 \bmod 5}$ $\longrightarrow$ *clean_chopstick*]

The interactions have the structure shown in Fig. 11, where p_i and c_j represent *philosopher*$_i$ and *chopstick*$_j$, respectively. The following is a possible scenario of the processes:

all processes are ready, and then the repeat of the following forever
philosopher$_3$, *chopstick*$_3$, *chopstick*$_4$ establish *eating_session*$_3$
philosopher$_3$, *chopstick*$_3$, *chopstick*$_4$ ready
philosopher$_1$, *chopstick*$_1$, *chopstick*$_2$ establish *eating_session*$_1$
philosopher$_1$, *chopstick*$_1$, *chopstick*$_2$ ready
...

This scenario satisfies WIF, and so is possible if the underlying implementation of the multiparty interactions guarantees only WIF. It can also be seen that the scenario does not satisfy SPF^+. Since by our results SPF^+ is implementable, we know that such a scenario can be avoided by an appropriate implementation.

On the other hand, since SPF^+ is the strongest implementable and equivalence-robust fairness notion, the following scenario which satisfies SPF^+ but not SPF cannot be excluded by any implementation ensuring equivalence-robust properties:

all processes are ready, and then the repeat of the following forever
philosopher$_3$, *chopstick*$_3$, *chopstick*$_4$ establish *eating_session*$_3$
philosopher$_3$, *chopstick*$_3$, *chopstick*$_4$ ready
philosopher$_1$, *chopstick*$_1$, *chopstick*$_2$ establish *eating_session*$_1$
philosopher$_1$, *chopstick*$_1$, *chopstick*$_2$ ready
philosopher$_0$, *chopstick*$_0$, *chopstick*$_1$ establish *eating_session*$_0$
philosopher$_0$, *chopstick*$_0$, *chopstick*$_1$ ready
...

It is interesting to note that in this scenario two non-neighboring philosophers (i.e., *philosopher*$_2$ and *philosopher*$_4$) are blocked from entering *eating_session*. So for this problem no implementation of the multiparty interactions can guarantee that at most one philosopher is starving.

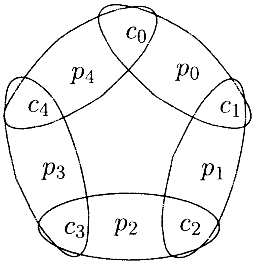


FIG. 11. The interaction structure of the Dining Philosophers problem.

4.3. When Interactions May Contain Interactions

If I contains two interactions x, y such that $P_x \subseteq P_y$, then in the biparty case the following relation (Lemma 4.4) no longer holds:

$$\mathrm{SG}^+(\mathbb{IS}) = \mathrm{SIF}^+(\mathbb{IS}) = \mathrm{SPF}^+(\mathbb{IS}).$$

Instead, the three fairness notions have the new relationship

$$\mathrm{SG}^+(\mathbb{IS}) \subsetneq \mathrm{SIF}^+(\mathbb{IS}) \subsetneq \mathrm{SPF}^+(\mathbb{IS}).$$

To see this, observe first that in general $\mathrm{SG}^+(\mathbb{IS}) \subseteq \mathrm{SIF}^+(\mathbb{IS}) \subseteq \mathrm{SPF}^+(\mathbb{IS})$ because every singular run satisfies SIF and SPF as well. For the proper subset relation, the run

$$\pi = P_y \cdot (y P_y x P_x)^\omega$$

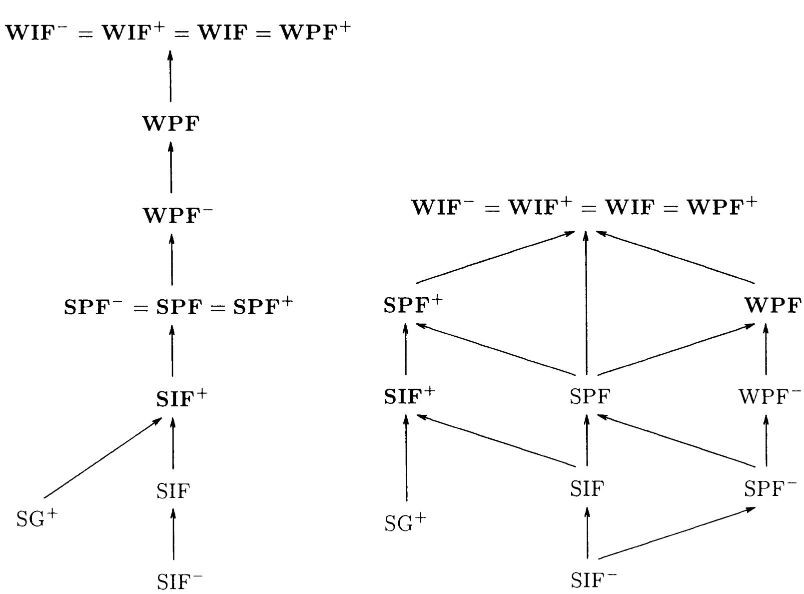
belongs to $\mathrm{SIF}^+(\mathbb{IS}) - \mathrm{SG}^+(\mathbb{IS})$ (assuming no interaction is contained in x), while

$$\rho = P_y \cdot (y P_y)^\omega$$

belongs to $\mathrm{SPF}^+(\mathbb{IS}) - \mathrm{SIF}^+(\mathbb{IS})$.

Moreover, in the above example π also belongs to $\mathrm{SIF}(\mathbb{IS}) - \mathrm{SG}^+(\mathbb{IS})$ and $\mathrm{SIF}^-(\mathbb{IS}) - \mathrm{SG}^+(\mathbb{IS})$. So SIF and SIF^- are no stronger than SG^+. But recall that when interactions cannot contain interactions, SIF and SIF^- are stronger than SG^+. Hence, in general, SG^+ is incomparable with SIF and SIF^-. Figure 12a summarizes the relationship of the fairness notions for biparty interactions.

Note that SG^+ becomes unimplementable. In fact, SG^+ is not even strongly feasible. (Recall by Lemmas 3.1 and 2.3 that if SG^+ were strongly feasible, then SG^+ would be the strongest implementable and equivalence-robust fairness notion, thus contradicting Theorem 3.8.) This is because if a system contains two interactions x and y such that $P_x \subseteq P_y$, then every time when y is enabled, x is enabled too (assuming a program of type $\mathsf{M}^{\forall}$). So all nonblocking schedulers for the system will inevitably generate



(a) Biparty Interactions (b) Multiparty Interactions

FIG. 12. The hierarchy of various fairness notions—the general case where interactions may contain interactions. The fairness notions in boldface are implementable, while the others are not.

some runs which cannot be equivalent to a singular run. Furthermore, although SIF$^+$ is the strongest implementable and equivalence-robust fairness notion shown in Fig. 12a, by Theorem 3.8 we know that there exists another implementable and equivalence-robust fairness notion which is no weaker than SIF$^+$.

When interactions are multipartied, the relationship of the fairness notions is shown in Fig. 12b. Recall that in this case SPF$^-$($\mathbb{IS}$) $\subseteq$ SPF($\mathbb{IS}$) $\subseteq$ SPF$^+$($\mathbb{IS}$). SG$^+$ is incomparable with SPF and SPF$^-$, but is stronger than SPF$^+$. The fact that SG$^+$ is incomparable with SPF and SPF$^-$ can be observed by the similar reason behind the incomparability between SG$^+$ versus SIF and SIF$^-$. The fact that SG$^+$ is stronger than SPF$^+$ is because SG$^+$ is stronger than SIF$^+$, which is stronger than SPF$^+$.

Moreover, SIF$^+$ is incomparable with SPF and SPF$^-$. To see this, let $x, y \in \mathsf{I}$ be two interactions such that $P_x \subsetneq P_y$. Then run $(P_y y)^\omega$ belongs to SPF($\mathbb{IS}$) $-$ SIF$^+$($\mathbb{IS}$) and SPF$^-$($\mathbb{IS}$) $-$ SIF$^+$($\mathbb{IS}$). So SPF and SPF$^-$ are no stronger than SIF$^+$. However, we have seen instances that SPF and SPF$^-$ are stronger than SIF$^+$ when interactions cannot contain interactions. So, in general, SPF and SPF$^-$ are incomparable with SIF$^+$.

Finally, although SG$^+$ in general is classified as unimplementable when some interaction y contains an interaction x, from Theorems 3.8 and 3.10 we know that the unimplementability holds only in the case where all other interactions contained in y, if any, contain x. Moreover, in the cases where SG$^+$ is implementable, although SG$^+$ is identical to SIF$^+$ (and SPF$^+$) when interactions can not contain interactions (see Lemma 4.4), SG$^+$ may be strictly stronger than SIF$^+$ when interactions can contain interactions. For example, let $\mathbb{IS} = (\mathsf{P} = \{p_1, p_2\}, \mathsf{I} = \{x_1, x_2, x_{12}\}, \mathsf{M}^\forall)$ be a system shown in Fig. 3a, where $P_{x_1} = \{p_1\}$, $P_{x_2} = \{p_2\}$, and $P_{x_{12}} = \{p_1, p_2\}$. Then the run $(p_1 x_1 p_2 x_2 p_1 p_2 x_{12})^\omega$ belongs to SIF$^+$($\mathbb{IS}$) $-$ SG$^+$($\mathbb{IS}$).

4.4. Further Remarks

In the comparison of SG$^+$ and existing fairness notions, we have divided the results into two categories: one for systems that support only biparty interactions and the other for systems that allow multiparty interactions of arbitrary arity. This classification is based on the fact that some popular languages/models (e.g., CSP, Ada, and CCS) facilitate only biparty interactions. For each category, we have further divided the results into two subcategories, depending on whether interactions may contain interactions. We have seen examples which do not need interactions to contain an interaction (see Sections 4.1 and 4.2). As shown in Theorem 3.4 and Lemma 4.4, disallowing interactions to contain interactions ensures a strongest implementable and equivalent-robust fairness notion for the system, namely, SG$^+$, which is identical to the maximal completions of SIF and SPF (regardless of biparty or multiparty interactions). Therefore, all other equivalent-robust fairness notions are either weaker than SG$^+$ or are unimplementable.

On some applications, however, one may find it useful to allow interactions to contain interactions. For example, a process p may choose to establish an interaction x with q and r, or an interaction y with only q. In the biparty case, a process p may choose to interact with q from a set of different interactions so as to perform different actions. If one will, one could eliminate the need for containing interactions within an interaction by defining only one interaction for the largest set of participants and let different subsets of participants establish different actions within the interaction. In the above biparty example, we may define a single interaction for p and q. Once the interaction is established, the two processes may negotiate with each other to decide which action to perform. Similarly, in the multiparty example we may replace x and y with an interaction w involving p, q, and r. However, r would become a "dummy" participant if only p and q interact within w. Since r must always be involved in w, the new setting is more restrictive as p and q may interact regardless of r in the original setting.

Perhaps the need for containing interactions within an interaction becomes more evident when one wishes to allow a choice between local actions and interactions. For example, consider a variant of the Producers–Consumers problem presented in Section 4.1, where each datum computed by a producer can be overwritten by a more up-to-date one if the target consumer is not yet ready for the data, and an old data can be "recycled" by a consumer if the producer cannot generate new data in time. Assuming only one producer and one consumer, then the following is a CSP program for the problem:

producer :: *compute*(*data*);
  * [*consumer* ! *data* ⟶ *skip*
  □ *compute*(*data*) ⟶ *skip*]
consumer :: * [*producer* ? *data* ⟶ *skip*
  □ *digest*(*data*) ⟶ *skip*]

In this example each process, when ready for interaction with the other process, has a choice to perform a local action. As noted in the definition of our abstract model (see Section 2.1 of Part I), such a local action can be modeled by an interaction involving solely the process. Therefore, the system has an interaction structure shown in Fig. 3a. Note that although some interaction contain interactions, by Theorem 3.10, SG^+ is still the strongest equivalent-robust fairness notion that can be implemented for the system. However, observe that the run

$$(\textit{producer} \quad \textit{compute} \quad \textit{consumer} \quad \textit{digest})^\omega$$

is a possible computation of the system (where *compute* and *digest* denote the local interactions by the processes) where the two processes repeatedly execute their local interactions forever. Since the run is singular, we see that no implementable equivalent-robust fairness notion can be enforced to ensure that the two processes will ever establish an interaction. It has been argued that nonuniform choice between local actions and interactions should be avoided to prevent *stuttering*—repetitions of a configuration in a computation [1]. From the above example we see that any attempt to impose an implementable and equivalent-robust fairness notion to prevent stuttering is doomed to fail.

Finally, although in this section we have only considered SG^+, SIF, SPF, WPF, and WIF, and their completions, based on our studies other fairness notions can be included in the fairness hierarchies as well. For example, U-fairness is incomparable with SIF^+ in the hierarchies of Fig. 12. The fact that $SIF^+(\mathbb{IS}) \not\subseteq U(\mathbb{IS})$ for some $\mathbb{IS}$ can be seen by the fact that SIF and U-fairness are incomparable, and that U-fairness is equivalent-robust [4]. The other direction can be illustrated by a system consists of two processes p_1 and p_2 and two interactions x and y, where $P_x = P_y = \{p_1, p_2\}$, such that the run $(p_1 p_2\, x\; p_1.\{x\}\; p_2.\{x\}\; x)^\omega$ belongs to $U(\mathbb{IS}) - SIF^+(\mathbb{IS})$. Moreover, it can also be shown that SIF^- is stronger than U-fairness, as any run that violates U-fairness must be equivalent to a run violating SIF.

5. THE IMPOSSIBILITY OF A STRONGEST IMPLEMENTABLE FAIRNESS NOTION

We now determine the possibility/impossibility of a strongest implementable (but not necessarily equivalence-robust) fairness notion for various interaction systems. For this, it is useful to recall the fairness implementability criterion and the definition of indistinguishableness relation introduced in Part I. We have shown in Theorem 3.4 that if no interaction contains an interaction, then SG^+ is the strongest fairness notion for $\mathbb{IS}$ that is both implementable and equivalence-robust. However, as we shall see shortly, SG^+ is *not* the strongest implementable fairness notion for $\mathbb{IS}$, unless I consists of only one interaction. Note that when interactions may contain interactions, SG^+, in general, is not strongly feasible and so is not implementable (see Section 4.3).

Since stronger fairness notions provide more liveness properties, one would wish to define a fairness notion as strong as possible while ensuring the implementability of the notion. Recall that every nonblocking scheduler S for $\mathbb{IS}$ must be able to generate all runs in $SG(\mathbb{IS})$. According to the fairness implementability criterion and Lemma 2.3, for every implementable fairness notion $\mathbb{C}$ and every $\pi \in SG(\mathbb{IS})$, $\mathbb{C}(\mathbb{IS})$ must contain *indistinct*(π). Thus, a potential candidate for the strongest implementable fairness notion is SG_INDISTINCT($\mathbb{IS}$), defined by

$$\text{SG\_INDISTINCT}(\mathbb{IS}) = \bigcup_{\pi \in SG(\mathbb{IS})} \textit{indistinct}(\pi).$$

However, it turns out that SG_INDISTINCT($\mathbb{IS}$) is not even strongly feasible. This holds even if interactions are bipartied and cannot contain interactions. In fact, as shown in the following two theorems, for every $\mathbb{IS}$ the strongest implementable fairness notion does not exist, unless $\mathbb{IS}$ consists of only one interaction.

THEOREM 5.1. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^{\forall})$ *and assume* $|\mathsf{I}| > 1$. *Then for every implementable fairness notion* $\mathbb{C}$, *there exists another implementable fairness notion* $\mathbb{C}'$ *such that* $\mathbb{C}(\mathbb{IS}) \not\subseteq \mathbb{C}'(\mathbb{IS})$.

Proof. Let x, y be any two distinct interactions in I. Let R_x and $R_{\bar{x}}$ be defined by

$$R_x = \{\pi \in \text{run}(\mathbb{IS}) \mid \pi \text{ begins with the form } (P_y - P_x) \cdot P_x\, x\}$$

$$R_{\bar{x}} = \{\pi \in \text{run}(\mathbb{IS}) \mid \pi \text{ begins with the form } (P_y \cup P_x - P_w) \cdot P_w\, w \text{ for some } w \neq x \text{ and } P_w \subseteq P_x \cup P_y\}$$

Clearly, both R_x and $R_{\bar{x}}$ are not empty.

Assume that $\mathbb{C}$ is an implementable fairness notion for $\mathbb{IS}$. By the fairness implementability criterion, there exists a nonblocking scheduler S such that for every run π generated by S, *indistinct*$(\pi) \subseteq \mathbb{C}(\mathbb{IS})$. We argue that that either $\mathbb{C}(\mathbb{IS}) \cap R_x \neq \emptyset$ or $\mathbb{C}(\mathbb{IS}) \cap R_{\bar{x}} \neq \emptyset$. To see this, consider an adversary which begins by letting the processes in $P_x \cup P_y$ be ready. If S in response schedules x for execution, then some run in *indistinct*$(r(S, A))$ begins with the form $(P_y - P_x) \cdot P_x x$, and so *indistinct*$(r(S, A)) \cap R_x \neq \emptyset$. Since *indistinct*$(r(S, A)) \subseteq \mathbb{C}(\mathbb{IS})$, $\mathbb{C}(\mathbb{IS}) \cap R_x \neq \emptyset$. Similarly, if S instead schedules a different interaction, then $\mathbb{C}(\mathbb{IS}) \cap R_{\bar{x}} \neq \emptyset$.

Suppose that $\mathbb{C}(\mathbb{IS})$ intersects both R_x and $R_{\bar{x}}$. Consider the following scheduler S_x:

> S_x behaves like the nonblocking scheduler for SIF presented in Part I. In particular, if x is enabled initially, then x is chosen for execution first.

We claim that for every run π generated by S_x, *indistinct*$(\pi) \cap R_{\bar{x}} = \emptyset$. This is because if the first interaction executed in π is x, then *indistinct*$(\pi) \cap R_{\bar{x}} = \emptyset$ (because all runs in *indistinct*(π) have x as their first interaction). If the first interaction is w for some $w \neq x$ and $P_w \subseteq P_x \cup P_y$, then x must not be enabled before the instance of w is to be executed, for otherwise S_x would instead choose x as the first interaction. So $P_x - P_w \neq \emptyset$. So π begins with $Q \cdot w$ for some Q such that some process (say p) in $P_x - P_w$ does not belong to Q. Then no run in *indistinct*(π) can begin with the form $(P_x \cup P_y - P_w) \cdot P_w w$ because the relation of indistinguishableness does not allow p's ready transition to be moved ahead of any ready transition in Q. So *indistinct*$(\pi) \cap R_{\bar{x}} = \emptyset$.

Define fairness notion $\mathbb{C}_x$ to be

$$\mathbb{C}_x(\mathbb{IS}) = \{\psi \in \textit{indistinct}(\pi) \mid \pi \text{ can be generated by } S_x\}.$$

Then, $\mathbb{C}_x$ is also implementable, and $\mathbb{C}_x(\mathbb{IS}) \cap R_{\bar{x}} = \emptyset$. So $\mathbb{C}(\mathbb{IS}) \not\subseteq \mathbb{C}_x(\mathbb{IS})$.

Note that since $\mathbb{C}_x$ does not intersect $R_{\bar{x}}$, if $\mathbb{C}(\mathbb{IS})$ intersects only $R_{\bar{x}}$, then clearly $\mathbb{C}(\mathbb{IS}) \not\subseteq \mathbb{C}_x(\mathbb{IS})$. Finally, if $\mathbb{C}(\mathbb{IS})$ intersects only R_x, then since the role of x and w is essentially symmetric, we can use a similar argument to show that there exists another implementable fairness notion $\mathbb{C}_{\bar{x}}(\mathbb{IS})$ such that $\mathbb{C}_{\bar{x}}(\mathbb{IS}) \cap R_x = \emptyset$. So $\mathbb{C}(\mathbb{IS}) \not\subseteq \mathbb{C}_{\bar{x}}(\mathbb{IS})$. ■

THEOREM 5.2. *Let* $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M}^{\forall})$. *If* $|\mathsf{I}| = 1$, *then there is only one implementable fairness notion, i.e.*, run$(\mathbb{IS})$ *which equals* $\mathrm{SG}^{+}(\mathbb{IS})$.

Proof. Straightforward. ■

Note that Theorem 5.1 does not depend on whether interactions are bipartied or multipartied, nor does it depend on whether interactions may contain interactions. It holds as long as I contains more than one interaction. Therefore, for every $\mathbb{IS}$ for which SG^{+} is implementable, there exists an implementable fairness notion which is no weaker than SG^{+}.

6. CONCLUSIONS

We have determined the system structure for which the strongest implementable completion of a given fairness notion exists. Moreover, for systems in which interactions do not contain interactions, we have obtained a fairness notion SG^{+} which is the strongest implementable and equivalence-robust fairness notion one can get for these systems. We have also presented a comprehensive comparison of

SG^+ with several commonly used fairness notions and their minimal and maximal completions. The results show that SG^+ is identical to the maximal completions of SPF and SIF and is stronger than WIF. Since WIF is generally accepted as the only fairness criterion by many multiparty interaction implementations, our results indicate that we could exclude more “unfair” computations from these implementations (see the examples in Sections 4.1 and 4.2). Moreover, when interactions are CSP-like bipartied, SG^+ is also equivalent to SPF. Therefore, SPF is the strongest equivalence-robust property one can observe from a CSP-like program executing in any asynchronous environment. Finally, we have shown that in the absence of equivalence-robustness, it is in general impossible to define a strongest implementable fairness notion, unless there is only one interaction in the system. This implies plenty of leeway in the design of fairness notions suitable for various applications.

In studying the relationships between various fairness notions and their minimal and maximal completions, we often assumed a program of type $\mathsf{M}^\forall$ when we need to distinguish two fairness notions, where $\mathsf{M}^\forall$ allows a process, whenever it is ready for interaction, to be ready for *all* interactions of which it is a participant. The choice of $\mathsf{M}^\forall$ also allows us to observe the structure of interactions that may distinguish two fairness notions. Based on this analysis, one may also analyze how the relationships are affected by the semantics of M for any given $\mathbb{IS} = (\mathsf{P}, \mathsf{I}, \mathsf{M})$, where M is not limited to type $\mathsf{M}^\forall$. In this case, the relationships are determined not only by the structure of I but also by the condition whether the semantics of M allows the interactions to be enabled as required so as to distinguish two fairness notions.

The notion of *liveness enhancement* is introduced in [1] as another fairness criterion. It requires a fairness notion to allow some system to gain some liveness property which the system would not have without the additional fairness requirement. Program termination is typically used to evaluate this criterion. By the example presented after Theorem 3.4, we see that SG^+ is also liveness enhancing.

Although we have not explicitly presented any scheduling algorithm to implement SG^+, we can easily obtain one by using the method proposed in Part I. The method transforms a nonblocking scheduler to a coordinator process running concurrently with the existing processes of the system. By communicating with the existing processes, the coordinator determines, for each ready process, when and which interaction to execute. Note that the scheduling is essentially centralized as all interactions are established by the coordinator. It is therefore worth exploring a *fully distributed* solution for the problem, meaning that nonconflicting interactions can be established by different coordinators. That would then indicate that SG^+ is the strongest equivalence-robust fairness notion that can be distributedly implemented (provided that no interaction contains an interaction).

ACKNOWLEDGMENTS

The author thanks Reino Kurki-Suonio for providing references on completions and some initial thoughts on the paper and the anonymous referees for their thorough reading of the manuscript and valuable comments and suggestions.

REFERENCES

1. Apt, K. R., Francez, N., and Katz, S. (1988), Appraising fairness in languages for distributed programming. *Distrib. Comput.* **2**(4), 226–241.
2. Back, R. J. R., and Kurki-Suonio, R. (1988), Serializability in distributed systems with handshaking, *in* “Proceedings of the 15th International Colloquium on Automata, Languages and Programming, Tampere, Finland,” Lecture Notes in Computer Science, Vol. 317, pp. 52–66, Springer-Verlag, Berlin.
3. Francez, N., Back, R. J. R., and Kurki-Suonio, R. (1992), On equivalence-completions of fairness assumptions. *Formal Aspects Comput.* **4**, 582–591.
4. Francez, N., and Forman, I. R. (1996). “Interacting Processes: A Multiparty Approach to Coordinated Distributed Programming,” Addison–Wesley, Reading, MA.
5. Joung, Y.-J. (2001), On fairness notions in distributed systems. I. A characterization of implementability, *Inform. Comput.* **166**, 35–60.
6. Reisig, W. (1984), Partial order semantics vs. interleaving semantics for CSP-like languages and its impact on fairness, *In* “Lecture Notes in Computer Science” (J. Paredaens, Ed.), Vol. 172, Springer-Verlag, Berlin.