Logics and Bisimulation Games for Concurrency, Causality and Conflict

Julian Gutierrez

LFCS. School of Informatics. University of Edinburgh
Informatics Forum, 10 Crichton Street, Edinburgh, EH8 9AB, UK
J.E.Gutierrez@ed.ac.uk

Abstract. Based on a simple axiomatization of concurrent behaviour we define two ways of observing parallel computations and show that in each case they are dual to conflict and causality, respectively. We give a logical characterization to those dualities and show that natural fixpoint modal logics can be extracted from such a characterization. We also study the equivalences induced by such logics and prove that they are decidable and can be related with well-known bisimulations for interleaving and noninterleaving concurrency. Moreover, by giving a game-theoretical characterization to the equivalence induced by the main logic, which is called Separation Fixpoint Logic (SFL), we show that the equivalence SFL induces is strictly stronger than a history-preserving bisimulation (hpb) and strictly weaker than a hereditary history-preserving bisimulation (hhpb). Our study considers branching-time models of concurrency based on transition systems and petri net structures.

Key words: Modal and temporal logics; Bisimulation games; Behavioural equivalences; Concurrent and reactive systems; Petri nets.

1 Introduction

In [12] Milner and Hennessy studied an algebraic characterization of concurrent systems and defined a formal way of comparing different processes through the equivalence induced by such an axiomatization. They called it observational equivalence. They also gave a logical characterization to the same concepts and showed their correspondence with the axiomatization when interpreted using a simple modal logic with an interleaving model of concurrency. This work led to the definition of a well-known bisimulation equivalence for interleaving concurrency, namely the one induced by the Hennessy-Milner Logic (HML).

However, when studying concurrency at a more fundamental semantic level, partial order models should be considered, different axiomatizations must be defined, and finer bisimulation equivalences have to be taken into account. A natural question immediately arises, that is, what exactly is the new relationship between concurrency in the models (a model independence) and concurrency in the logical languages (a logical independence). The answer to this question is not unique since it depends on the models and axiomatizations that are being considered. Therefore, one would like them to be as general as possible.

In this paper an alternative answer for this question is put forward. Our results are relative to a surprisingly simple axiomatization of concurrent behaviour for a category of transition systems with independence introduced by Winskel and Nielsen (see [16] or Section 2). This axiomatization was pioneered by Mazurkiewicz for trace languages [14] and has been used ever since to understand the properties of several models of concurrency with partial order semantics, also called independence models [16]. Such axioms define local properties of the behaviour of a concurrent system and can be used to generate what is called the "concurrency diamonds" in some noninterleaving models of concurrency. Our results are also extended to a class of safe Petri net structures.

In particular, we study the relationships between model independence and logical independence purely based on observable *dualities* between concurrency and conflict, on the one hand, and concurrency and causality on the other. The logical characterization of these dualities relies on geometrical properties enforced by an algebraic axiomatization of concurrent events. At a semantic level, we develop a notion of locality called *support sets* which allows the recognition of independent events (Section 3). At a logical level we define both modal operators sensitive to causal information and a parallel conjunction that allows one to separate off concurrent events while avoiding those in conflict. This basic modal logic is then extended with fixpoint operators so as to express temporal properties of concurrent systems (Section 4). The final outcome is a fixpoint modal logic, which we call Separation Fixpoint Logic (SFL). A set of application examples for SFL is given at the end of this section.

We also study the equivalence induced by SFL and some sublogics of it (Section 5). These sublogics can be obtained from SFL by restricting *syntactically* the interplay between the dualities we have defined. This strongly eases the analysis of the relationships between concurrency, causality and conflict. Four SFL fragments are considered. The first one is Kozen's modal mu-calculus, $L\mu$. This SFL sublogic help us show that SFL can deal equally well with both interleaving and noninterleaving models of concurrency. Since the mu-calculus is a natural logical language for interleaving concurrent systems (and SFL includes it), nothing is lost with respect to the main interleaving approaches to concurrency.

However, the problem of observing concurrency and nondeterminism, as studied by Milner and Hennessy in [12] for interleaving systems, can be refined to a problem of observing concurrency, causality and conflict in a noninterleaving context. As a consequence, more specialized bisimulation equivalences have been introduced so as to analyse independence models. We study some of those equivalences and focus our attention on the two strongest equivalences presented in [7, 10], namely the hereditary history-preserving bisimulation (hhpb) [13] and the plain history-preserving bisimulation (hpb) [10].

The second sublogic we consider is a Separation modal mu-calculus, $SL\mu$. It extends the previous fragment by allowing the distinction between concurrency and conflict. We show that the equivalence induced by $SL\mu$ cannot be compared with any of the bisimulations we consider here. The third case is a Causal modal mu-calculus, $CL\mu$. It takes full account of causality and concurrency and induces

an hp bisimulation equivalence. Although hpb captures some features of concurrent computation, it has been argued that it is actually an equivalence of only causality [9]. As a consequence, stronger equivalences such as hhpb have been defined in order to capture aspects of true-concurrency rather than only causality. Therefore, in fourth place we consider the full SFL and compare the equivalence it induces with the stronger hhpb. Using game-theoretical arguments, we prove that the bisimulation equivalence for SFL is strictly stronger than hpb and strictly weaker than hhpb (Section 6). Since hhpb is undecidable, even on finite systems, the equivalence induced by SFL ranks at the top of the decidable equivalences for true-concurrency according to discriminating power (see [10] or completed hierarchy in [7]). This feature makes the equivalence induced by SFL an interesting candidate for an equivalence of true concurrency. Finally, some concluding remarks and related work are given in Section 7.

2 An Independence Model and Axioms of Concurrency

A *Transition System with Independence* (TSI) [16] is a simple extension of a Labelled Transition System (LTS), where independent transitions can be explicitly recognised. A TSI is a structural, branching-time and noninterleaving model of concurrency. Formally, a TSI $\mathfrak{T}$ is a structure (S, T, Σ, I) , where S is a finite set of states, $T \subseteq S \times \Sigma \times S$ is a transition relation, Σ is a set of action labels, and $I \subseteq T \times T$ is an irreflexive and symmetric relation on independent transitions, i.e., on concurrent transitions. The binary relation $\prec$ on transitions defined by

$$(s, a, s_1) \prec (s_2, a, q) \Leftrightarrow$$
$$\exists b.(s, a, s_1)I(s, b, s_2) \wedge (s, a, s_1)I(s_1, b, q) \wedge (s, b, s_2)I(s_2, a, q)$$

expresses when two transitions represent two occurrences of the same *event* (the same action in different interleavings). Such a relation can be used to generate a "concurrency diamond", as shown in Fig. 1.

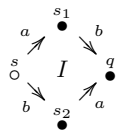

Fig. 1. A concurrency diamond for $a\ I\ b$. Concurrent transitions are recognized by the I symbol inside the square. The initial state of the TSI is marked by the circle $\circ$.

Moreover, $\sim$ is the least equivalence relation that includes $\prec$, i.e., the reflexive, symmetric and transitive closure of $\prec$. The equivalence relation $\sim$ is used to group all events that represent the same action in the TSI, and therefore, considers all occurrences of events of an action in all its possible interleavings. Additionally, the I relation is subject to the following axioms:

– **A1**. $(s, a, s_1) \sim (s, a, s_2) \Rightarrow s_1 = s_2$
– **A2**. $(s, a, s_1)I(s, b, s_2) \Rightarrow \exists q.(s, a, s_1)I(s_1, b, q) \wedge (s, b, s_2)I(s_2, a, q)$
– **A3**. $(s, a, s_1)I(s_1, b, q) \Rightarrow \exists s_2.(s, a, s_1)I(s, b, s_2) \wedge (s, b, s_2)I(s_2, a, q)$
– **A4**. $(s, a, s_1) \prec \cup \succ (s_2, a, q)I(w, b, w') \Rightarrow (s, a, s_1)I(w, b, w')$

Intuitively, **A1** states that from any state of the system, the execution of an action leads always to a unique future state. **A2** (resp. **A3**) says that if two independent actions can be executed in parallel (resp. one after the other), they can also be executed one after the other (resp. in parallel). Finally, **A4** ensures that the relation I between transitions is well defined. Roughly speaking **A4** says that if actions a and b are independent, then all the transitions that are occurrences of the action a are independent of all the transitions that are occurrences of the action b. Having said that, we can give an alternative, more intuitive, definition for axiom **A4**. Let $\mathcal{I}(t)$ be the set $\{t' \mid tIt'\}$. Then, axiom **A4** is equivalent to the following expression: **A4**. $t_1 \sim t_2 \Rightarrow \mathcal{I}(t_1) = \mathcal{I}(t_2)$.

Notation 2.1 *Given a transition $t = (s_1, a, s_2)$, also written as $s_1 \xrightarrow{a} s_2$, s_1 is the source, $src(t) = s_1$; s_2 the target, $trg(t) = s_2$; and a the label of t, $lbl(t) = a$.*

3 Local Dualities in Independence Models

We present two ways in which concurrency can be regarded as a dual concept to *conflict* and *causality*, respectively. These two ways of observing concurrency will be called *immediate concurrency* and *linearized concurrency*. Whereas immediate concurrency is dual to conflict, linearized concurrency is dual to causality.

The intuitions behind these two observations are the following. Consider a concurrent system, say, a TSI, and any two different transitions t_i and t_j with the same source node, i.e., $src(t_i) = src(t_j)$. These two transitions are either immediately concurrent, and therefore belong to I, or dependent, in which case they must be in conflict. Similarly, consider any two transitions t_i and t_j where $trg(t_i) = src(t_j)$. Again, these two transitions can either belong to I, in which case they are concurrent, yet have been linearized, or they do not belong to I, and therefore are causally dependent. In both cases, the two conditions are exclusive and *there are no other possibilities*.

Definition 3.1 Two transitions t_i and t_j, such that $trg(t_i) = src(t_j)$, are *linearly concurrent* iff $t_i I t_j$. We write $t_i \ominus t_j$ for such a relation.

Dually, causally dependent transitions can be defined.

Definition 3.2 Two transitions t_i and t_j, such that $trg(t_i) = src(t_j)$, are *causally dependent* iff $\neg(t_i I t_j)$. We write $t_i \leq t_j$ for such a relation.

This duality is defined locally with respect to a state s, such that $trg(t_i) = s = src(t_j)$, of the underlying independence model. The previous definitions will be used to give the semantics of the modal operators of SFL. Now, we turn our attention to the duality between immediate concurrency and conflict.

Definition 3.3 Two transitions t_i and t_j, such that $src(t_i) = src(t_j)$, are *immediately concurrent* iff $t_i I t_j$. We write $t_i \otimes t_j$ for such a relation.

Dually, transitions in conflict can be defined as follows.

Definition 3.4 Two transitions t_i and t_j, such that $src(t_i) = src(t_j)$, are *in conflict* iff $\neg(t_i I t_j)$. We write $t_i \# t_j$ for such a relation.

3.1 Separation and Support Sets for Local Reasoning

The definitions of immediate concurrency and conflict can be used to recognize sets where every transition is concurrent with each other and therefore can all be executed in parallel. We call these sets as *conflict-free sets of transitions*.

Definition 3.5 A *conflict-free set of transitions*, denoted by $\mathrm{cf}(E)$, is a set of transitions E such that $\forall t_i, t_j \in E.\ src(t_i) = src(t_j) \wedge (t_i \neq t_j \Rightarrow t_i \otimes t_j)$.

As we want to specify the existence of actual concurrent behaviour, we strengthen the definition of conflict-free sets of transitions given above. Notice that Definition 3.5 allows the existence of empty and singleton sets, which do not show any actual concurrent behaviour (or even any behaviour at all).

Definition 3.6 An *effective conflict-free set of transitions* is a conflict-free set of transitions E such that $|E| \geq 2$.

So, in order to do local reasoning on concurrent processes of an independence model, we want to recognize effective conflict-free sets given an arbitrary set of transitions at some state s of the independence model. In particular, the set of all transitions t such that $src(t) = s$, will be called the *maximal set* of transitions at s. Now, we introduce a notion of locality called *support sets* that is used to define the semantics of SFL formulae in the following section.

Definition 3.7 A *support set* is either a maximal set of transitions or a non-empty conflict-free set of transitions.

Given an independence model, the set of all its support sets is denoted by $\mathfrak{P}$. Notice that once one has non-empty conflict-free sets of transitions that are not singleton sets, i.e. effective conflict-free sets of transitions, to do local reasoning on sets of concurrent actions becomes easier since they can be *separated* or decomposed into smaller sets, where every transition is, as well, concurrent with each other. Finally, the following definitions are useful when defining the semantics of some SFL operators in the next section:

$E \sqsubseteq R \stackrel{\text{def}}{=} E \subseteq R$, provided that, $\mathrm{cf}(E)$ and $\neg\exists t \in (R \setminus E).\ \forall t_e \in E.\ t \otimes t_e$

$P_1 \uplus P_2 \stackrel{\text{def}}{=} P_1 \cup P_2$, provided that, $P_1 \cap P_2 = \emptyset \wedge P_1 \neq \emptyset \wedge P_2 \neq \emptyset$

The first definition, $E \sqsubseteq R$, characterizes support sets E that contain only concurrent transitions and cannot be made any bigger with respect to R; the second definition, $P_1 \uplus P_2$, formalizes the notion of *separation* for local reasoning used here. Such a definition resembles the notion of separation as used in Separation Logic [19], i.e., as disjointness of sets of independent resources.

4 Separation Fixpoint Logic

Definition 4.1 *Separation Fixpoint Logic* (SFL) has formulae ϕ built from a set Var of variables $\mathrm{Y}, \mathrm{Z}, ...$ and a set $\mathcal{L}$ of labels $a, b, ...$ by the following grammar:

$$\phi ::= Z \mid \neg\phi_1 \mid \phi_1 \wedge \phi_2 \mid \langle K\rangle_c\phi_1 \mid \langle K\rangle_{nc}\phi_1 \mid \phi_1 * \phi_2 \mid \mu Z.\phi_1$$

where $Z \in \mathrm{Var}$, $K \subseteq \mathcal{L}$, and $\mu Z.\phi_1$ has the restriction that any free occurrence of Z in ϕ_1 must be within the scope of an even number of negations.

Informally, the meaning of these operators is the following. "$\wedge$" and "$\neg$" are the usual boolean operators, "$\langle K\rangle_c$" (resp. "$\langle K\rangle_{nc}$") asserts at the fact that there is in the set of actions K a causally dependent (resp. a non-causally dependent or linearly concurrent) transition that can be performed; as defined in Section 3, such a transition is always either causally dependent or linearly concurrent w.r.t. the last transition that has been executed. $\phi_1 * \phi_2$ specifies that there exists a partition in the support set (i.e., a partition of the actions in the set to be considered) with which both formulae ϕ_1 and ϕ_2 can hold in parallel. This does not necessarily mean that both formulae hold in parallel everywhere because the operator "$*$" has a local meaning. Finally, "μ" is simply a least fixpoint operator.

Derived operators are defined in the familiar way: $\phi_1 \vee \phi_2 \stackrel{\text{def}}{=} \neg(\neg\phi_1 \wedge \neg\phi_2)$, $\phi_1 \bowtie \phi_2 \stackrel{\text{def}}{=} \neg(\neg\phi_1 * \neg\phi_2)$, $[K]_c\,\phi_1 \stackrel{\text{def}}{=} \neg\langle K\rangle_c\neg\phi_1$, $[K]_{nc}\,\phi_1 \stackrel{\text{def}}{=} \neg\langle K\rangle_{nc}\neg\phi_1$ and $\nu Z.\phi_1 \stackrel{\text{def}}{=} \neg\mu Z.\neg\phi_1\,[\neg Z/Z]$. The following abbreviations are also useful: $\mathrm{ff} \stackrel{\text{def}}{=} \mu Z.Z$, $\mathrm{tt} \stackrel{\text{def}}{=} \neg\mathrm{ff}$, $\langle K\rangle\phi_1 \stackrel{\text{def}}{=} \langle K\rangle_c\phi_1 \vee \langle K\rangle_{nc}\phi_1$, $[K]\,\phi_1 \stackrel{\text{def}}{=} [K]_c\,\phi_1 \wedge [K]_{nc}\,\phi_1$. Also, sometimes we write $[-]\,\phi$ for $[\mathcal{L}]\,\phi$ and $[-K]\,\phi$ for $[\mathcal{L} - K]\,\phi$, and similarly for the other box ("[ ]") and diamond ("⟨ ⟩") operators.

4.1 Denotation of SFL Formulae

Definition 4.2 A *TSI-based SFL model* $\mathfrak{M}$ is a Transition system with independence $\mathfrak{T} = (S, T, \Sigma, I)$ together with a valuation $\mathcal{V} : \mathrm{Var} \to 2^{\mathfrak{S}}$, where $\mathfrak{S} = S \times \mathfrak{P} \times \mathfrak{A}$ is the set of tuples (s, P, t_a) of states $s \in S$, support sets $P \in \mathfrak{P}$ in the TSI $\mathfrak{T}$, and transitions $t_a \in \mathfrak{A} = T \cup \{t_\epsilon\}$, where a is an action label in $\Sigma \cup \{\epsilon\}$. The denotation $\|\phi\|^{\mathfrak{T}}_{\mathcal{V}}$ of an SFL formula ϕ in the model $\mathfrak{M} = (\mathfrak{T}, \mathcal{V})$ is a subset of $\mathfrak{S}$, given by the following rules (omitting the superscript $\mathfrak{T}$):

$$\begin{aligned}
\|Z\|_{\mathcal{V}} &= \mathcal{V}(Z)\\
\|\neg\phi\|_{\mathcal{V}} &= \mathfrak{S} - \|\phi\|_{\mathcal{V}}\\
\|\phi_1 \wedge \phi_2\|_{\mathcal{V}} &= \|\phi_1\|_{\mathcal{V}} \cap \|\phi_2\|_{\mathcal{V}}\\
\|\langle K\rangle_c\phi\|_{\mathcal{V}} &= \{(s, P, t_a) \mid \exists b \in K.\ \exists s' \in S.\ t_a \leq s \xrightarrow{b} s' \in P \wedge (s', P', t_b) \in \|\phi\|_{\mathcal{V}}\}\\
\|\langle K\rangle_{nc}\phi\|_{\mathcal{V}} &= \{(s, P, t_a) \mid \exists b \in K.\ \exists s' \in S.\ t_a \ominus s \xrightarrow{b} s' \in P \wedge (s', P', t_b) \in \|\phi\|_{\mathcal{V}}\}\\
\|\phi_1 * \phi_2\|_{\mathcal{V}} &= \{(s, P, t_a) \mid \exists P_1, P_2.\ P_1 \uplus P_2 \sqsubseteq P \wedge (s, P_1, t_a) \in \|\phi_1\|_{\mathcal{V}} \wedge (s, P_2, t_a) \in \|\phi_2\|_{\mathcal{V}}\}
\end{aligned}$$

where P' is the maximal set at s'. A tuple (s, P, t_a) of a model $\mathfrak{M}$ is called a *process*. An initial process is a tuple (s, P, t_ϵ), where s is the initial state of the TSI $\mathfrak{T}$, P is the maximal set at s, and t_ϵ is the empty transition, such that

for all t whose source node is the initial state, $t_\epsilon \leq t$. Notice that since the third component of every process denotes the last action that has been made, an initial process requires the definition of the empty transition t_ϵ. Also, since an initial state defines a unique initial process, and vice-versa, the two terms and notations are interchangeable.

Given the usual restriction on free occurrences of variables, imposed in order to obtain monotone operators in the complete lattice $\mathcal{P}(\mathfrak{S}) = 2^{\mathfrak{S}}$, it is possible to define the denotation of the fixpoint operator $\mu Z.\phi(Z)$ in the standard way, according to the Knaster-Tarski fixpoint theorem:

$$\|\mu Z.\phi(Z)\|_{\mathcal{V}} = \bigcap\{Q \in \mathcal{P}(\mathfrak{S}) \mid \|\phi\|_{\mathcal{V}[Z:=Q]} \subseteq Q\}$$

where $\mathcal{V}[Z := Q]$ is the valuation $\mathcal{V}'$ which agrees with $\mathcal{V}$ save that $\mathcal{V}'(Z) = Q$. Since positive normal form is assumed henceforth, the semantics of the dual boolean, modal, structural and fixpoint operators can be given in the usual way.

A Petri Net Semantics. The logical characterization defined previously using a TSI model can also be given using other independence models. As an example of this, it is shown how to give a Petri net semantics for SFL, but analogous procedures can be followed with other independence models.

A labelled net $\mathfrak{N}$ is a tuple $(P, T, A, \mathcal{F}, \Sigma)$, where P is a set of places, T is a set of transitions and A is a relation between places and transitions such that $A \subseteq (P \times T) \cup (T \times P)$, and $\mathcal{F}$ is a labeling function, $\mathcal{F} : T \to \Sigma$, from transitions to a finite set of action labels Σ. Places and transitions are called nodes. Given a node n, ${}^\bullet n = \{x \mid (x, n) \in A\}$ is the preset of n and $n^\bullet = \{y \mid (n, y) \in A\}$ is the postset of n. A marking M of $\mathfrak{N}$ is a mapping such that $M \subseteq P$. A marking M enables a transition t iff ${}^\bullet t \subseteq M$. If t is enabled at M, then t can occur, and its occurrence leads to a successor marking M', where $M' = (M \setminus {}^\bullet t) \cup t^\bullet$. The nets considered here are safe Petri nets [16]. All safe nets are finite systems. Given a 1-safe labelled net with an initial marking, the set of all reachable markings is fixed and can be constructed with the occurrence net.

We call a reachable marking M as a state s of the system $\mathfrak{N}$, and S as the set of all possible reachable markings or states of the system. Since in this model transitions represent actions rather than events (as in the TSI case), every transition is annotated with a particular marking M to recognize which event they refer to. Therefore we can write t_a^M for the event of the action or transition t with label a at state $s = M$, i.e., whenever ${}^\bullet t \subseteq M$ and $\mathcal{F}(t) = a$. Support sets for nets are defined as for TSIs considering that $src(t_a^M) = M$ (to define conflict-free sets of transitions) and the maximal set at a state $s = M$ is the set $\{t \mid {}^\bullet t \subseteq M\}$. Finally, the dualities used to give the denotation to SFL formulae in a model $\mathfrak{M} = (\mathfrak{N}, \mathcal{V})$ are defined as for TSIs provided that the I relation on events in the net model is defined for any two events t_a^M and t_b^N of the transitions t_a and t_b, respectively, whenever there is a marking such that t_a and t_b are enabled at such marking and ${}^\bullet t_a \cap {}^\bullet t_b = \emptyset$. With this information the set of processes $\mathfrak{S}$ and the valuation $\mathcal{V}$ can be defined, and the net model $\mathfrak{M}$ for SFL is straightforward.

4.2 Applications

A Temporal Logic. SFL can express all usual temporal properties, such as, liveness, safety, fairness, and so on. These properties are equally handled in both interleaving and noninterleaving models of concurrency.

A Process Logic. SFL can differentiate concurrency from nondeterminism using two different local dualities. Consider these concurrent systems: $P = a.0 \parallel b.0$ and $Q = a.b.0 + b.a.0$. P and Q are behaviourally equivalent in an interleaving context (since they are strongly bisimilar [12]), but different from a true-concurrency point of view (since they are not history-preserving bisimilar [10]). Such a difference can be captured in these two ways: with the formulae $\phi = \langle a\rangle \mathrm{tt} * \langle b\rangle \mathrm{tt}$ or $\psi = \langle a\rangle_c \langle b\rangle_{nc} \mathrm{tt}$, which are both true in P but not in Q.

A Petri net Logic. SFL can express properties of net systems by defining properties of the localities and synchronization mechanisms. In order to do so, it is useful to define a derived operator that allows one to perform a causal step that may require resource from *separate localities* so as to proceed (a strongly causal step). Consider the derived operator: $\langle a\rangle_{sc}\phi \stackrel{\text{def}}{=} \langle a\rangle_c \mathrm{tt} \wedge \mu Z.\langle a\rangle\phi \vee (\langle a\rangle \mathrm{tt} * \langle -a\rangle Z)$. Expressing that a property p eventually holds in a causal line can be easily defined as $\phi = \mu Z.p \vee \langle -\rangle_{sc} Z$. The result of evaluating this formula corresponds to finding a "line" in the net to some state where p holds, even though separate resource may be needed. These kinds of properties can be further studied with the structural operators of SFL. For instance, the formula $\phi = \mu Z.p \vee \langle -\rangle_{sc} Z * \mu Y.q \vee \langle -\rangle_{sc} Y$ not only states that properties p and q hold eventually in two causal lines, but also that such causal lines have an independent source.

A resource-sensitive Logic. The parallel conjunction of SFL can also be used to define data structures, such as, lists and trees, *à la* Reynolds [19], but using a temporal specification approach. In SFL independent transitions can be treated as resources, allowing indirectly the specification of independent localities in the model. For instance, let $\phi = \mu Z.\langle \mathrm{nil}\rangle_c \mathrm{tt} \vee (\langle \mathrm{data}\rangle_c \mathrm{tt} * \langle -\rangle_c Z)$. Formula ϕ represents an abstraction of a finite list structure where all nodes in the list causally depend on the previous one (a sequential structure), whereas its data and pointers are spatially separated and thus can be regarded as independent. Since our approach is not proof-theoretic, there is not a "frame rule" similar to the one defined for Separation Logic and concurrent extensions of it [6, 11, 19].

A Logic for Multi-Agent Systems (MAS). MAS, such as those specified with logics like ATL [1] can be studied using SFL. In order to express properties of MAS, define a set Γ of agents, a labeling function $\mathcal{B}$ on sets of transitions, and a mapping $\mathcal{A}$ that assigns transitions to agents, subject to the restriction that if $t_1 \sim t_2$ then $\mathcal{A}(t_1) = \mathcal{A}(t_2)$. For simplicity, we write $\langle K\rangle^\gamma$ for $\langle \mathcal{B}(\mathcal{A}^{-1}(\gamma))\rangle$, and similarly for the other modal operators. The formula $\psi = [-]^\beta \langle -\rangle^\alpha_{nc} \mu Z.\phi \vee \langle -\rangle^\alpha_c Z$ expresses that there is an agent α (the system) that can satisfy ϕ regardless the behaviour of an adversarial agent β (the environment).[1] Informally, ψ says

[1] I thank Bradfield for the IFML version of this example. $\mathcal{A}^{-1}$ is the inv. func. of $\mathcal{A}$.

"whatever you (the environment) do, I (the system) can get to ϕ, though I may first have to do some things that do not depend on what you did."

5 Logical and Concurrent Equivalences

We now turn our attention to the study of the relationships between model independence and logical independence. We do so by relating well-known equivalences for concurrency, namely hpb [10] and hhpb [13], with the equivalences induced by different SFL sublogics where the interplay between concurrency and conflict, and concurrency and causality is restricted *syntactically*. The reader is referred to [10, 13] or [5] for a detailed description of the history-preserving bisimulations not proposed in this paper (hpb and hhpb), but studied in the following sections. Due to lack of space, in most cases, proofs are omitted or simply sketched.

Definition 5.1 ***(SFL equivalence*** $\sim_{SFL}$***).*** Two processes P and Q of two independence models $\mathfrak{T}_1$ and $\mathfrak{T}_2$, respectively, are SFL-equivalent, $P \sim_{SFL} Q$, iff for every SFL formula ϕ in $\mathcal{F}_{SFL}$, $P \models^{\mathfrak{T}_1} \phi \Leftrightarrow Q \models^{\mathfrak{T}_2} \phi$, where $\mathcal{F}_{SFL}$ is the set of all fixpoint-free closed formulae of SFL.

Remark 5.1. Similar definitions can be made for the equivalences of the SFL sublogics we present here.

Remark 5.2. In order to obtain an exact match between finitary modal logic and bisimulation, all models considered in this paper are image-finite [12].

5.1 SFL, HML and the Modal Mu-Calculus

The first SFL sublogic is obtained from SFL by disabling its sensitivity to both dualities. On the one hand, insensitivity to the duality between concurrency and causality can be captured by considering only modalities without subscript, using the abbreviations for modalities given previously in Section 4. On the other hand, insensitivity to the duality between concurrency and conflict can be captured by considering the $*$-free SFL sublanguage. The resulting logic has the following syntax: $\phi ::= Z \mid \neg\phi_1 \mid \phi_1 \wedge \phi_2 \mid \langle K \rangle \phi_1 \mid \mu Z.\phi_1$. This SFL syntactic fragment is the modal $*$-free fragment of SFL.

Proposition 1 *The modal $*$-free fragment of SFL is Kozen's mu-calculus, $L\mu$.*

Proof. By computing the semantics of the derived operators of this sublogic. □

Remark 5.3. This SFL sublogic cannot recognize elements in I and therefore sees TSIs as plain LTSs, or what is equivalent, TSIs with an empty I relation. As a consequence, although using an independence model, it is possible to retain in SFL all the joys of a logic with an interleaving model, and so, *nothing is lost with respect to the main interleaving approaches to concurrency.*

Regarding logical and concurrent equivalences, it is now easy to see that Milner's *strong bisimulation* [12], $\sim_b$, the equivalence induced by modal logic is captured by the fixpoint-free fragment of this SFL sublogic, which we can denote by $\sim_{L\mu}$. Hence, the relation $\sim_{L\mu} \equiv \sim_b$ follows from Proposition 1 and the fact that modal logic characterises bisimulation on finite models.

5.2 A Separation Modal Mu-Calculus

The second sublogic is the Separation modal mu-calculus, $SL\mu$. This logic is obtained from SFL by allowing only the recognition of the duality between concurrency and conflict by using its structural operator. The syntax of $SL\mu$ is $\phi ::= Z \mid \neg\phi_1 \mid \phi_1 \wedge \phi_2 \mid \langle K\rangle\phi_1 \mid \phi_1 * \phi_2 \mid \mu Z.\phi_1$. We write $\sim_{SL\mu}$ for the equivalence induced by this SFL sublogic. It is easy to see that $SL\mu$ is more expressive than $L\mu$ in independence models simply because $SL\mu$ includes $L\mu$ and can differentiate concurrency from nondeterminism. However, the following counter-examples show that $\sim_{SL\mu}$ and $\sim_{hpb}$, in general, do not coincide.

Proposition 2 *Neither* $\sim_{hpb} \subseteq \sim_{SL\mu}$ *nor* $\sim_{SL\mu} \subseteq \sim_{hpb}$.

Proof. The two systems on the right (in Fig. 2) are hpb and yet can be distinguished by the formula $\phi = \langle a\rangle\langle c\rangle\mathrm{tt} * \langle b\rangle\langle d\rangle\mathrm{tt}$. On the other hand, the systems on the left are not hpb and cannot be differentiated by any $SL\mu$ formula. □

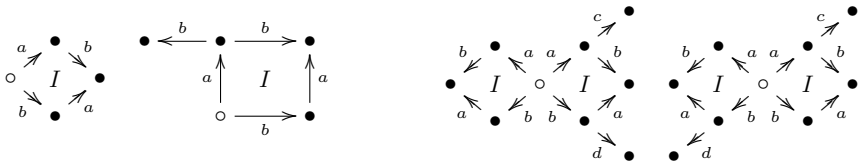

Fig. 2. Systems used for defining the equivalence induced by Separation $L\mu$

There are two reasons for the mismatch between $\sim_{hpb}$ and $\sim_{SL\mu}$. The first one (related with the two systems on the left in Fig. 2) is that $\sim_{hpb}$, unlike $\sim_{SL\mu}$, recognizes the pattern "an action a followed by both a concurrent action b and a causally dependent action b". The second reason, which is related with the systems on the right, is that $\sim_{hpb}$ cannot recognize some forms of conflict that $SL\mu$ can capture. For instance, the fact that even though two actions can be performed in parallel, it is also possible that the execution of one of them affects (prevents) the execution of transitions that depends on the other.

5.3 A Causal Modal Mu-Calculus

The third fragment to be considered is the Causal modal mu-calculus, $CL\mu$. This sublogic is obtained from SFL by allowing only the recognition of the duality between concurrency and causality throughout the modal operators of the logic. Its syntax is $\phi ::= Z \mid \neg\phi_1 \mid \phi_1 \wedge \phi_2 \mid \langle K\rangle_c\phi_1 \mid \langle K\rangle_{nc}\phi_1 \mid \mu Z.\phi_1$. Clearly, $CL\mu$ is also more expressive than $L\mu$ in independence models because of the same reasons given for $SL\mu$. The naturality of $CL\mu$ for expressing causal properties is demonstrated by the equivalence it induces in any model, written as $\sim_{CL\mu}$, which coincides with a *history-preserving bisimulation*, $\sim_{hpb}$, [10].

Theorem 1 $\sim_{CL\mu} \equiv \sim_{hpb}$

Proof. The proof goes by showing the two inclusions separately, $\sim_{hpb} \subseteq \sim_{CL\mu}$ and $\sim_{CL\mu} \subseteq \sim_{hpb}$. The first inclusion, $\sim_{hpb} \subseteq \sim_{CL\mu}$, can be proved by induction on $CL\mu$ formulae, called $\mathcal{F}_{CL\mu}$. For any two processes P and Q that belong to TSIs $\mathfrak{T}_1$ and $\mathfrak{T}_0$, respectively, if $P \sim_{hpb} Q$ then for all ϕ in $\mathcal{F}_{CL\mu}$, $P \models^{\mathfrak{T}_1} \phi \Leftrightarrow Q \models^{\mathfrak{T}_0} \phi$. The induction only requires (simplified) processes $P = (p, t_a)$ and $Q = (q, t_a)$, which are binary tuples in $S \times \mathfrak{A}$ of a TSI-model $\mathfrak{M}$, because $CL\mu$ only considers maximal sets and therefore support sets can be disregarded.

The second inclusion, $\sim_{CL\mu} \subseteq \sim_{hpb}$, is shown by contradiction. Suppose that for all ϕ in $\mathcal{F}_{CL\mu}$ we have that $P \models^{\mathfrak{T}_1} \phi \Leftrightarrow Q \models^{\mathfrak{T}_0} \phi$ and $P \not\sim_{hpb} Q$. The contradiction comes from the fact that even though processes P and Q satisfy the same $CL\mu$ formulae, there would be a transition in one of the processes that cannot be simulated by the other process in an hpb way, i.e., concurrent actions matched only with concurrent ones so as to keep the bisimulation synchronous, which is impossible. More precisely, synchrony in an hp bisimulation means that the last transition chosen in $\mathfrak{T}_1$ (resp. in $\mathfrak{T}_2$) is concurrent with the former transition also chosen in $\mathfrak{T}_1$ (resp. in $\mathfrak{T}_2$) iff the same pattern holds in the last two transitions chosen in $\mathfrak{T}_2$ (resp. in $\mathfrak{T}_1$). □

Corollary 1 $\sim_{CL\mu}$ *is decidable.*

Proof. Follows from Theorem 1 and the fact that $\sim_{hpb}$ is decidable. □

5.4 The Full Separation Fixpoint Logic

Although the equivalence induced by SFL is analysed in the following section using game-theoretical arguments, we first present a simple result that relates $\sim_{SFL}$ with $\sim_{hhpb}$, without using any game-theoretical machinery. Consider the counter-example presented in [8] (or the TSI representation of it), which is used there to disprove the coincidence between hpb and hhpb in free-choice systems. Due to lack of space such a counter-example is not reported here. Although the systems presented in [8] (in Figure 1) are not hhp bisimilar, they cannot be distinguished by any SFL formula. This result shows that in general $\sim_{hhpb}$ does not coincide with $\sim_{SFL}$. However, the precise relation between $\sim_{hhpb}$ and $\sim_{SFL}$ is to be defined in the following section. For now, we have the following result:

Proposition 3 $\sim_{SFL} \not\subseteq \sim_{hhpb}$

6 Bisimulation Games

Based on some of the results of the previous section, we give a game-theoretical characterization of the equivalence SFL induces by defining a Bisimulation Game for it. A knowledge on basic concepts about bisimulation games is assumed. An introduction to Bisimulation games can be found in [21].

The games presented here conservatively extend the hp bisimulation game, and therefore usual games for modal logics, i.e., classical bisimulation. We prove that these bisimulation games, which we call independence hp bisimulation (ihpb) games, characterize the equivalence induced by SFL. Most importantly, it is shown that the ihp bisimulation games induce an equivalence relation for concurrent systems that is strictly stronger than hpb and strictly weaker than hhpb. Remarkably, whereas ihpb games are decidable, hhpb games are undecidable in finite models. These features make ihpb games, and consequently the equivalence induced by SFL, an interesting candidate for an equivalence of true-concurrency. A hierarchy of true concurrent equivalences can be found in [7].

Definition 6.1 A *linearized concurrent run* is any sequence of transitions $r = t_0, ..., t_n$ of an independence model $\mathfrak{T}$ such that $s \xrightarrow{r} s'$ for some states s and s' in $\mathfrak{T}$. The set of linearized concurrent runs of a structure $\mathfrak{T}$ with respect to an initial state s can be written as $cRuns_s(\mathfrak{T})$. An empty run is written as $r = \epsilon$. The last transition of the sequence $r = t_0, ..., t_n$ is denoted by $last(r) = t_n$.

Remark 6.1. Any play of an (h)hp bisimulation game corresponds exactly to a pair of linearized concurrent runs in the structures at hand.

Before presenting the games for SFL, let us introduce a final definition that is related with the role of support sets as locally identifiable sets of concurrent transitions (conflict-free sets of transitions).

Definition 6.2 Two sets P_1 and P_2 are said to be *history-preserving isomorphic* with respect to a transition t iff there exists a bijection $\mathcal{B}$ between them such that for every $(t_i, t_j) \in \mathcal{B}$, if $t \leq t_i$ (resp. $t \ominus t_i$) then $t \leq t_j$ (resp. $t \ominus t_j$).

Definition 6.3 *(Independence history-preserving bisimulation Games).* Consider the standard bisimulation game for modal logics. There are two players, Spoiler and Duplicator, and a pair of structures $\mathfrak{T}_1$ and $\mathfrak{T}_2$ with initial states/processes P and Q, respectively. A configuration of a play in the game is a pair (r_1, r_2), where $r_1 \in cRuns_P(\mathfrak{T}_1)$ and $r_2 \in cRuns_Q(\mathfrak{T}_2)$. R is an independence history-preserving bisimulation between $\mathfrak{T}_1$ and $\mathfrak{T}_2$ if:

1. The initial configuration is (ϵ, ϵ). Therefore $(\epsilon, \epsilon) \in R$
2. (Rule for hp bisimulation). Let (r_1, r_2) be the current configuration, thus $(r_1, r_2) \in R$. Spoiler chooses one of the two systems, say $\mathfrak{T}_1$ ($\mathfrak{T}_2$), and picks a transition t_1 (t_2) that is enabled at r_1 (r_2). Duplicator has to respond by executing a transition t_2 (t_1) in the opposite structure $\mathfrak{T}_2$ ($\mathfrak{T}_1$) such that the two extended linearized concurrent runs stay synchronous. Synchrony means that whenever $last(r_1) \leq t_1$ (resp. $last(r_1) \ominus t_1$) then $last(r_2) \leq t_2$ (resp. $last(r_2) \ominus t_2$). The new configuration of the play is $(r_1.t_1, r_2.t_2)$, and hence $(r_1.t_1, r_2.t_2) \in R$.
3. (Additional rule for ihp bisimulation). Before Spoiler chooses a transition t_1 (t_2) from the enabled ones at r_1 (r_2), he can also choose a non-empty conflict-free subset of them to be the new set of enabled transitions P_1 (P_2) of size n. Duplicator must choose a history-preserving isomorphic set P_2 (P_1) with respect to $last(r_2)$ ($last(r_1)$) in the opposite structure $\mathfrak{T}_2$ ($\mathfrak{T}_1$).

If the play continues forever or Spoiler cannot make a move, then Duplicator wins. Otherwise Spoiler wins. The two structures are ihp bisimilar iff Duplicator has a winning strategy for every play in the game.

Lemma 1 *If $P \sim_{SFL} Q$, then Duplicator has a winning strategy for every play in the independence history-preserving bisimulation game $\mathcal{G}(\mathfrak{T}_1, \mathfrak{T}_2, P, Q)$.*

Proof. By constructing a winning strategy for Duplicator based on the fact that $P \sim_{SFL} Q$. Since $CL\mu$ induces an hp bisimulation and the ihpb game conservatively extends the hpb game, w.l.o.g. we can consider only the case when Spoiler plays Rule 3 of the ihpb game. □

Lemma 2 *If Duplicator has a winning strategy for every play in the independence history-preserving bisimulation game $\mathcal{G}(\mathfrak{T}_1, \mathfrak{T}_2, P, Q)$, then $P \sim_{SFL} Q$.*

Proof. By contradiction suppose that Duplicator has a winning strategy and $P \not\sim_{SFL} Q$. There are two cases. Suppose that Spoiler cannot make a move. This means that both $P \models^{\mathfrak{T}_1} [-]\,\mathrm{ff}$ and $Q \models^{\mathfrak{T}_2} [-]\,\mathrm{ff}$ only, which is a contradiction. The other case is when Duplicator wins in an infinite play. For the same reasons given previously, w.l.o.g., it is possible to consider only the case when Spoiler uses the Rule 3 of the ihpb game. But, this case also leads to a contradiction. □

The previous two Lemmas give a full game-theoretical characterization to the equivalence induced by SFL.

Theorem 2 *$P \sim_{SFL} Q$ iff Duplicator has a winning strategy for every play in the independence history-preserving bisimulation game $\mathcal{G}(\mathfrak{T}_1, \mathfrak{T}_2, P, Q)$.*

Theorem 3 *$\sim_{ihpb} \equiv \sim_{SFL}$ is decidable on finite systems.*

Proof. An independence history-preserving bisimulation game is a two-player zero-sum perfect-information (infinite) game, thus it is determined. Moreover, the length of the game is bounded. Duplicator wins when Spoiler cannot make a move or when a finite set of repeated configurations is visited infinitely often. In either case all possible winning strategies can be computed and therefore decidability follows. □

The previous results let us relate $\sim_{hhpb}$ with $\sim_{SFL}$ using game-theoretical arguments. Since both bisimulation games, namely the one for hhpb one and the one for ihpb, are conservative extensions of the hp bisimulation game, they can be compared just by looking at their additional rules with respect to the hpb game. So, consider the game-theoretical definition of hhpb as presented in [15]. We will describe the hhpb rule that extends the hpb game in the style used here.

Definition 6.4 *(Hereditary history-preserving bisimulation Games).* An hhp bisimulation game is just as an hp bisimulation game, as presented in Definition 6.3 (only Rules one and two), adding the following rule:

– (Additional rule for hhp bisimulation). Alternatively to a forwards move, having chosen one of the two systems, say $\mathfrak{T}_1$, Spoiler can choose a transition t_i that is backwards enabled at r_1 with respect to a previous configuration (w_1, w_2), i.e., a transition t_i that is concurrent with every transition t after w_1, where w_1 is obtained from r_1 by using the "diamond axioms" to push t_i to the end, and then deleting t_i, and symmetrically for w_2. Duplicator must respond by choosing (deleting) the corresponding t_j in w_2. The new configuration of the game is that obtained by deleting both transitions t_i and t_j from the history of the game.

Now, only by showing that the additional rule for the hhpb game is at least as powerful as the additional rule for the ihpb game, and taking into account that $\sim_{hhpb}$ and $\sim_{SFL}$ do not coincide, by Proposition 3, we have:

Theorem 4 $\sim_{hhpb} \subset \sim_{SFL}$

7 Concluding Remarks and Related Work

We have given a logical characterization to the *dualities* that can be found when analysing *locally* the relationships between concurrency and conflict as well as concurrency and causality. This characterization aims at defining relationships between *model independence* and *logical independence*. Our study led to several positive results with respect to the equivalences induced by a number of modal logics whose denotations are based on these dualities.

Related work: At a philosophical level, this study is similar to that in [3, 5], a work primarily on mathematical logic using game logics for concurrency where model independence is captured explicitly with the use of Henkin quantifiers. At a more practical level, this work can be related to logics with partial order semantics at large. These logics, in the simplest cases, are given denotations that consider the one-step interleaving semantics of a particular independence model. Therefore no new logical constructions have to be introduced. The problem is that the explicit notion of independence in the models is completely lost. Thus, the usual approach is to introduce logical operators that somehow capture the independence in the models. In most cases that kind of logical independence is actually a *sequential* interpretation of concurrency based on the introduction of past operators sensitive to concurrent actions and a mixture of forwards and backwards reasoning (which usually leads to several undecidable results). A survey of logics of this kind can be found in [17]. Other logics with partial order semantics can be found in [2, 15], but the literature has many more references.

Also, the work here presented can be related to logics for local reasoning. In particular, the use of separation properties to do local reasoning has been investigated elsewhere and applied in other settings [4, 6, 11, 18–20]. However, as said before, apart from [3, 5] the main motivation for this paper is different from all the examples given above, since we actually want to distill the relationships between model and logical independence so as to understand the semantic foundations of concurrent computations.

Acknowledgements. I am grateful to Julian Bradfield for helpful discussions. I thank Ian Stark and Colin Stirling for comments on a preliminary version of this paper. I also thank the referees for their comments.

References

1. R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.
2. R. Alur, D. Peled, and W. Penczek. Model-Checking of Causality Properties. In *LICS*, pages 90–100. IEEE Computer Society, 1995.
3. J. Bradfield. Truth and Games: Essays in Honour of Gabriel Sandu. In T. Aho and A. Pietarinen, editors, *Acta Philosophica Fennica*, volume 78, chapter Independence: Logics and Concurrency, pages 47–70. Phil. Soc. of Finland, 2006.
4. J. Bradfield, J. Esparza, and A. Mader. A Causal Fixpoint Logic. *Unpub.*, 1997.
5. J. Bradfield and S. Fröschle. Independence-Friendly Modal Logic and True Concurrency. *Nord. J. Comput.*, 9(1):102–117, 2002.
6. S. Brookes. A Semantics for Concurrent Separation Logic. In P. Gardner and N. Yoshida, editors, *CONCUR*, volume 3170 of *LNCS*, pages 16–34. Springer, 2004.
7. H. Fecher. A Completed Hierarchy of True Concurrent Equivalences. *Inf. Process. Lett.*, 89(5):261–265, 2004.
8. S. Fröschle. The Decidability Border of Hereditary History Preserving Bisimilarity. *Inf. Process. Lett.*, 93(6):289–293, 2005.
9. S. Fröschle and T. Hildebrandt. On Plain and Hereditary History-Preserving Bisimulation. In *MFCS*, volume 1672 of *LNCS*, pages 354–365. Springer, 1999.
10. R. Glabbeek and U. Goltz. Refinement of Actions and Equivalence Notions for Concurrent Systems. *Acta Inf.*, 37(4/5):229–327, 2001.
11. J. Hayman and G. Winskel. Independence and Concurrent Separation Logic. In *LICS*, pages 147–156. IEEE Computer Society, 2006.
12. M. Hennessy and R. Milner. Algebraic Laws for Nondeterminism and Concurrency. *J. ACM*, 32(1):137–161, 1985.
13. A. Joyal, M. Nielsen, and G. Winskel. Bisimulation from Open Maps. *Inf. Comput.*, 127(2):164–185, 1996.
14. A. Mazurkiewicz. Trace Theory. In Brauer, Reisig, and Rozenberg, editors, *Advances in Petri Nets*, volume 255 of *LNCS*, pages 279–324. Springer, 1986.
15. M. Nielsen and C. Clausen. Games and Logics for a Noninterleaving Bisimulation. *Nord. J. Comput.*, 2(2):221–249, 1995.
16. M. Nielsen and G. Winskel. Models for Concurrency. In *Handbook of Logic in Computer Science*, volume 4, pages 1–148. Oxford University Press, 1995.
17. W. Penczek. Branching Time and Partial Order in Temporal Logics. In *Time and Logic: A Computational Approach*, pages 179–228. UCL Press, 1995.
18. D. Pym and C. Tofts. A Calculus and Logic of Resources and Processes. *Formal Asp. Comput.*, 18(4):495–517, 2006.
19. J. Reynolds. Separation Logic: A Logic for Shared Mutable Data Structures. In *LICS*, pages 55–74. IEEE Computer Society, 2002.
20. É.-J. Sims. Extending Separation Logic with Fixpoints and Postponed Substitution. *Theor. Comput. Sci.*, 351(2):258–275, 2006.
21. C. Stirling. *Modal and Temporal Properties of Processes*. Springer, 2001.