URL: http://www.elsevier.nl/locate/entcs/volume52.html LastPage pages

On logical and concurrent equivalences

J. C. Bradfield and S. B. Fröschle

Laboratory for Foundations of Computer Science, Division of Informatics, King's Bldgs, University of Edinburgh, Edinburgh EH9 3JZ, UK.
{jcb,sib}@dcs.ed.ac.uk

Abstract

We consider modal analogues of Hintikka et al.'s 'independence-friendly first-order logic', and discuss their relationship to equivalences previously studied in concurrency theory.

Key words: Independence, concurrency, branching quantifiers, modal logic.

1 Introduction

In [1], Alur, Henzinger and Kupfermann introduced Alternating Temporal Logic, based on certain imperfect information games, in which independent 'teams' synchronize. In [3], the first author proposed the application of logics based on Henkin quantifiers to modal logic in computer science; such logics include ATL, but also allow more powerful forms of expression. In that paper, we argued that making sense of such logics required some notion of locality in processes. After establishing some basic facts about such logics, we left open the obvious question of how such logics relate to established notions of independence and concurrency in computer science.

In this paper, we first interpret Henkin modal logics in a setting without locality (at least, without explicit locality), and then relate them to some of the natural true concurrent notions in the literature. The results here are preliminary, but, we believe, go some way towards a satisfactory explanation, and open up many further questions.

2 Henkin quantifiers and independence-friendly logic

We give a brief summary of the notions of Henkin quantifier and independence-friendly logic.

A *branching quantifier* Q is a set $\{x_1, \ldots, x_m, y_1, \ldots, y_n\}$ of variables, carrying a partial order $\prec$; the x_i are universal, the y_i existential. The semantics of $Q\phi$ is defined to be that of $\exists f_1 \ldots f_n . \forall x_1 \ldots x_m . \phi[f_i(y_i\!\downarrow)/y_i]$, where $y_i\!\downarrow$ is the list of variables $\prec y_i$, and $[\cdot/\cdot]$ denotes syntactic substitution: thus f_i is a Skolem function for y_i, but it refers only to variables preceding y_i in the partial order.

In particular, the *Henkin quantifier* $\begin{smallmatrix}\forall \exists \\ \forall \exists\end{smallmatrix} = \{x_1, x_2, y_1, y_2\}$ with $x_i \prec y_i$ is written $\begin{smallmatrix}\forall x_1 \exists y_1 \\ \forall x_2 \exists y_2\end{smallmatrix}$; thus $\begin{smallmatrix}\forall x \exists y \\ \forall u \exists v\end{smallmatrix}\phi(x, y, u, v)$ is equivalent by definition to $\exists f, g. \forall x, u. \phi(x, f(x), u, g(u))$.

Henkin quantifiers turn out to have existential second-order power, and are thus a strong operator to add to one's logic.

An alternative way of giving semantics to branching quantifiers is via games. Recall the Hintikka model-checking game for first-order logic (in positive form): given a formula ψ and a structure M, a position is a subformula $\phi(\vec{x})$ of ψ together with a *deal* for ϕ, that is, an assignment of values $\vec{v}$ to its free variables $\vec{x}$. At a position $(\forall x. \phi_1, \vec{v})$, Abelard chooses a value v for x, and play moves to the position $(\phi_1, \vec{v} \cdot v)$; similarly Eloise moves at $\exists x. \phi$. At $(\phi_1 \wedge \phi_2, \vec{v})$, Abelard chooses a conjunct ϕ_i, and play moves to $(\phi_i(\vec{x}\,'), \vec{v}\,')$, where $\vec{x}\,', \vec{v}\,'$ are $\vec{x}, \vec{v}$ restricted to the free variables of ϕ_i; and at $(\phi_1 \vee \phi_2, \vec{v})$, Eloise similarly chooses a disjunct. A play of the game terminates at (negated) atoms $(P(\vec{x}), \vec{v})$ (resp. $(\neg P(\vec{x}), \vec{v})$), and is won by Eloise (resp. Abelard) iff $P(\vec{v})$ is true. Then it is standard that $M \vDash \phi$ exactly if Eloise has a winning strategy in this game, where a strategy is a function from sequences of legal positions to moves.

These games have *perfect information*; both players know everything that has happened, and in particular when one player makes a choice, they know the other player's previous choices. Game semantics for the Henkin quantifiers, following [8], use games of imperfect information: in the game for $\begin{smallmatrix}\forall x \exists y \\ \forall u \exists v\end{smallmatrix}\phi$, when Eloise chooses for v, she does not know what Abelard chose for x. To make this explicit, the logic is written with a more general syntax which is linear rather than two dimensional. A full account of the appropriate logic requires several new constructs, some of which raise subtle issues [9]; we shall work with a restricted version which is sufficient to express all Henkin quantifiers.

In addition to the usual first-order syntax, we also have *independent quantification*: If ϕ is a formula, x a variable, and W a finite set of variables, then $\forall x/W. \phi$ and $\exists x/W. \phi$ are formulae. The intention is that W is the set of independent variables, whose values the player is not allowed to know at this choice point: thus the Henkin quantifier $\begin{smallmatrix}\forall x \exists y \\ \forall u \exists v\end{smallmatrix}$ can be written as $\forall x/\varnothing. \exists y/\varnothing. \forall u/\{x, y\}. \exists v/\{x, y\}$. If one then plays the usual model-checking game with this additional condition, which can be formalized by requiring strategies to be uniform in the 'unknown' variables, one gets a game semantics which characterizes the Skolem function semantics in the sense that Eloise has a winning strategy iff the formula is true.

This logic is called by Hintikka 'independence-friendly' logic. Study of this particular formalism has been mostly carried out by Hintikka and colleagues; but there has been over the last thirty years a continued interest in branching quantification in natural language semantics, increased now by the current popularity of ‘Game Theoretical Semantics’. (The recent thesis [12] contains a most useful account of this area.) However, there has been little interest in the computer science temporal logic community.

3 Independence-friendly modal logic

One reason for this is that at first sight, independence-friendly modal logic makes little sense. Suppose that we extend the usual syntax of modal logic with the Hintikka slash; we will also need to assign a tag to each modality, so that we can refer back to it after a slash.

Definition 3.1 *The syntax of independence-friendly modal logic (IFML) is given as follows. Let $\alpha, \beta, \ldots$ range over a countable set of* tags*, $a, b, \ldots$ over a set of* labels*.* tt *and* ff *are IFML formulae. If Φ_1 and Φ_2 are IFML formulae, so are $\Phi_1 \vee \Phi_2$ and $\Phi_1 \wedge \Phi_2$; and so are $\langle a \rangle_{\alpha/\beta_1,\ldots,\beta_m} \Phi_1$ and $[a]_{\alpha/\beta_1,\ldots,\beta_m} \Phi_1$.*

Certain syntactic conditions may be imposed:

Definition 3.2 *An IFML formula Φ is* well-formed *if*

(a) *in every subformula $\langle a \rangle_{\alpha/\beta_1,\ldots,\beta_m} \Psi$, the* bound tag *$\alpha$ is uniquely bound in Φ;*

(b) *every* independent tag *β_i* of *α is bound in some higher modality in Φ.*
It is moreover good *if*

(c) *the* dependency relation *on tags given by $\alpha \prec \beta$ if β is not an independent tag of α, is transitive.*

We will for this paper restrict ourselves to good formulae.

Of the well-formedness requirements, (a) is a convenience to avoid renaming, but (b) is more controversial: it implies, for example, that a subformula of a well-formed formula is not in general well-formed. This is an issue related to questions of compositional semantics; see [9] for a discussion.

The ‘goodness’ requirement is a restriction largely for technical convenience. If the dependency relation is not transitive, one can have a phenomenon called ‘signalling’ [9], whereby intendedly independent choices can be made dependent. Although this is interesting in certain linguistic applications, in ‘normal’ mathematics, and arguably in logics for concurrency, it is undesirable.

Obviously, the intended semantics of an independence-friendly modal logic is that the existential choice in the $\langle a \rangle_{\alpha/\beta_1,\ldots,\beta_m}$ must be made independently of the choices made in the modalities tagged by β_i. However, in a standard transition system semantics for modal logic, the choices available at a modality are determined by the choices made in earlier modalities, and thus in general it makes no sense to ask for an independent choice.

This problem is removed if the events referred to in the modalities are ‘independent’ in some sense. For example, in a system comprising two parallel, non-communicating, components, two independent modalities can reasonably refer to choices made in different components. Moreover, the two independent local choices may result in only a single action at a global system level, as when in CCS two actions synchronize; it is this situation that gives the new expressive power in the ATL of [1], and in the 'Henkin modal logic' of [3]. This observation then naturally raises the question of the relationship between independence in the meaning of Hintikka, and independence in semantic models for concurrency.

To examine this question, we shall revert from models with independence implicitly given by locality, to a model with explicit independence. Of the many possibilities, let us choose *transition systems with independence*; these are perhaps the nearest model to ordinary labelled transition systems, and have been used by Nielsen and others to study branching-time logics of (concurrent) independence.

First, we banish a confusing clash of terminology. In 'transition systems with independence', the independence is concurrency, in the model; we wish to relate this to Hintikka-style logical 'independence'. Therefore, *henceforth, concurrent model independence will be called 'concurrency'; 'independence' will be used only to refer to logical independence.* We stress that 'concurrency' is here being used as an *ad hoc* term to distinguish model independence from logic independence. In the literature, 'concurrency' is a distinct concept from model 'independence'; because we will make restrictions on our classes of models, the distinction does not occur in our setting. (We welcome suggestions for better terminology.)

Definition 3.3 *A* coherent transition system with concurrency *(TSC) is a labelled transition system with states S, labels L, and transition relation $\rightarrow \subseteq S \times L \times S$, together with a relation $\mathcal{C} \subseteq \rightarrow \times \rightarrow$ and an initial state s_0. Two transitions $t_1 = (s_1 \xrightarrow{a_1} s_1')$ and $t_2 = (s_2 \xrightarrow{a_2} s_2')$ are* concurrent *if $(t_1, t_2) \in \mathcal{C}$. A relation $\prec$ between transitions with the same label is defined by*

$$s_1 \xrightarrow{a} s_1' \prec s_2 \xrightarrow{a} s_2' \Leftrightarrow \exists b.\, (s_1' \xrightarrow{b} s_2')\ \mathcal{C}\ (s_1 \xrightarrow{a} s_1')\ \mathcal{C}\ (s_1 \xrightarrow{b} s_2)\ \mathcal{C}\ (s_2 \xrightarrow{a} s_2')$$

(i.e., the two a transitions form a diamond with two b transitions independent of a; notionally, the two a transitions are the same a 'event', and the two b transitions are the same b 'event'); $\sim$ is the reflexive, symmetric and transitive closure of $\prec$, and it groups transitions into events. *In addition, the relation $\mathcal{C}$ is required to satisfy four natural axioms which ensure that an event has a unique outcome at a given state, that concurrent transitions may occur in either order, that concurrency respects events, and that two concurrent events*

can occur one after the other:

$$
\begin{array}{ll}
1. & s \xrightarrow{a} s_1 \sim s \xrightarrow{a} s_2 \Rightarrow s_1 = s_2 \\
2. & s \xrightarrow{a} s_1 \, \mathcal{C} \, s_1 \xrightarrow{b} u \Rightarrow \exists s_2.\, s \xrightarrow{a} s_1 \, \mathcal{C} \, s \xrightarrow{b} s_2 \, \mathcal{C} \, s_2 \xrightarrow{a} u \\
3. & s \xrightarrow{a} s_1 \prec s_2 \xrightarrow{a} u \, \mathcal{C} \, w \xrightarrow{b} w' \Rightarrow s \xrightarrow{a} s_1 \, \mathcal{C} \, w \xrightarrow{b} w' \\
\textit{and} & w \xrightarrow{b} w' \, \mathcal{C} \, s \xrightarrow{a} s_1 \prec s_2 \xrightarrow{a} u \Rightarrow w \xrightarrow{b} w' \, \mathcal{C} \, s_2 \xrightarrow{a} u \\
4. & s \xrightarrow{a} s_1 \, \mathcal{C} \, s \xrightarrow{b} s_2 \Rightarrow \exists u.\, s_1 \xrightarrow{b} u \, \mathcal{C} \, s \xrightarrow{a} s_1
\end{array}
$$

(a plain TSC need not satisfy axiom 4, the coherence axiom; however, most reasonable models and classes of models are coherent, and we need it for Theorem 6.10, so we adopt it as a standard requirement). Consequently, a firing sequence of transitions gives rise to a partial order of events, which can be linearized into several different transition sequences, in the usual way of partial order semantics. (Note: in the literature, I is used rather than $\mathcal{C}$, as TSCs are called TSIs.)

In graphical depictions of TSCs, concurrent transitions are denoted by putting the symbol $\mathcal{C}$ inside the commutative square, and the initial state is marked by a circle (when it is not obvious).

We can now define a semantics for IFML, given *à la* Hintikka, by defining its model-checking game as a game of imperfect information. A consequence of this is that the semantics is not defined on states, but requires some history to be kept.

Definition 3.4 *A* tagged run *of a TSC is a sequence $s_0 \xrightarrow[\alpha_0]{a_0} \ldots \xrightarrow[\alpha_{n-1}]{a_{n-1}} s_n$, where the α_i are distinct* tags*; we shall also use the tag α_i to refer to the transition $s_i \xrightarrow{a_i} s_{i+1}$. We let ρ, σ etc. range over tagged runs, and use obvious notations for extensions of runs.*

A position *of the model-checking game for an IFML formula Φ on a TSC is a pair of a tagged run and a subformula, written $\rho \vdash \Psi$.*

The initial position is $s_0 \vdash \Phi$.

The rules of the game are as follows:

- *At a position $\rho \vdash \mathrm{tt}$, Eloise wins; at $\rho \vdash \mathrm{ff}$, Abelard wins.*
- *At $\rho \vdash \Phi_1 \vee \Phi_2$ (resp. $\vdash \Phi_1 \wedge \Phi_2$), Eloise (resp. Abelard) chooses a new position $\rho \vdash \Phi_i$.*
- *At $\rho = s_0 \xrightarrow{a_0} \ldots \xrightarrow{a_{n-1}} s_n \vdash \langle b \rangle_{\beta/\alpha_{i_1},\alpha_{i_2},\ldots,\alpha_{i_m}} \Psi$ (resp. $\vdash [b]_{\beta/\alpha_{i_1},\alpha_{i_2},\ldots,\alpha_{i_m}} \Psi$), Eloise (resp. Abelard) chooses a transition $s_n \xrightarrow{b} t$ that is concurrent with all the transitions α_{i_j}, and the new position is $\rho \xrightarrow[\beta]{b} t \vdash \Psi$.*

Tags are, of course, merely syntactic sugar; it suffices to identify the ith transition by i. However, tags are convenient to match the definition of IFML.

As usual, a strategy for Eloise is a function from her positions to choices. Imperfect information games are handled by imposing additional conditions on strategies.

Definition 3.5 *An Eloise strategy σ is* uniform *if the choice at a $\langle\rangle$ position is uniform in the specified independent earlier choices, in the following sense:*

Let $\rho \vdash \langle b\rangle_{\beta/\alpha_{i_1},\alpha_{i_2},\ldots,\alpha_{i_m}}\Psi$ be as above. The strategy σ must choose $s_n \xrightarrow{b} t$ such that if $s_0 = s'_0 \xrightarrow{a_0} \ldots \xrightarrow{a_{n-1}} s'_n \vdash \langle b\rangle_{\beta/\alpha_{i_1},\alpha_{i_2},\ldots,\alpha_{i_m}}\Psi$ is any other position such that $j \notin \{i_1,\ldots,i_m\} \Rightarrow \alpha_j \sim \alpha'_j$, σ chooses a transition $s'_n \xrightarrow{b} t' \sim s_n \xrightarrow{b} t$. (In words, σ must choose the same event regardless of the events chosen in the independent modalities. If no such event can be chosen, there is no uniform strategy.) Abelard uniform strategies are defined similarly.

Definition 3.6 *An IFML formula Φ is* true *in a given TSC, written $s_0 \vDash \Phi$, iff Eloise has a uniform winning strategy for the model-checking game $s_0 \vdash \Phi$.*

Φ is false *iff Abelard has a uniform winning strategy.*

Φ is determined *iff it is either false or true.*

The non-determinacy in general of the model-checking game is a characteristic feature of independence-friendly logic. For a simple example, consider the TSC generated by the CCS process $((a.c + a.\bar{c}) \mid (b.c + b.\bar{c}))\backslash c$ (in which the a transitions are independent of the b transitions), and the formula $[a]_\alpha\langle b\rangle_{\beta/\alpha}\langle\tau\rangle\mathtt{tt}$. This formula is not true, since Eloise cannot choose a b transition so as to synchronize unless she knows which a transition was chosen; but it is also not false, since Abelard has no strategy for falsifying it. For practical purposes, we may consider untruth to be falsehood.

4 IFML equivalence

One of the first questions about any logic is, what is the induced equivalence? In the case of IFML (or indeed the simpler Henkin modal logic of [3]), the definition of equivalence itself is problematic, because of the non-determinacy. We take the weaker (practical) definition, and say

Definition 4.1 *Two TSCs S and T are IFML-equivalent, $S \sim_{\mathrm{IFML}} T$, if for every IFML formula Φ, $S \vDash \Phi \Leftrightarrow T \vDash \Phi$.*

Logically induced equivalences are typically characterized by a game naturally related to the satisfaction game: for modal logic, we have bisimulation games and model-checking games, for first-order logic we have Ehrenfeucht–Fraïssé games and Hintikka games. For IF logics, the outscoping nature of the / makes such a formulation harder, and to our knowledge none has been presented. We will consider E–F games for independence logics in a later article; here we study IFML equivalence by relation to known equivalences in true concurrency.

5 Restrictions on models

For the remainder of this paper, we will consider restricted classes of models. Analysing the effect of removing the restrictions is left to later work.

Firstly, *all TSCs will be image-finite*: that is, for any state s and label a, there are only finitely many a-successors of s. This is a standard restriction required to obtain an exact match between finitary modal logic and bisimulation.

Secondly, *all TSCs will be acyclic*: that is, no state is reachable from itself. This restriction avoids the necessity of distinguishing between models and their unfoldings, which in turn avoids the necessity to distinguish multiple occurrences of the 'same' event.

Finally, we require the dependency relations in the models to be transitive; this is formally, but not actually, a further restriction, since events that are formally concurrent but actually causally dependent can be made formally non-concurrent without change to the model.

6 Equivalences for concurrency

There are numerous equivalences for concurrency, but there is one spectrum of particularly natural equivalences that appears promising: the spectrum from bisimulation through to coherent hereditary history preserving bisimulation. These equivalences have several characterizations; we will define them in the style of classical bisimulation, and also give the game characterizations, which will be useful in our results.

The weakest equivalence is ordinary 'strong bisimulation'; this is well known to be too weak for true concurrent properties, but we define it just to help clarify the other definitions. In particular, we will define it on runs, rather than states.

Definition 6.1 *A relation R on pairs of runs of two TSCs S and T is a* (strong) bisimulation *if*

A $(s_0, t_0) \in R$

B *if $(\sigma, \tau) \in R$ and $\sigma' = \sigma \xrightarrow{a} s$ is a run, then there is t such that $\tau' = \tau \xrightarrow{a} t$ and $(\sigma', \tau') \in R$; and symmetrically.*

Systems S and T are (strongly) bisimilar), *$S \sim_{\mathrm{b}} T$, if there is a strong bisimulation between them.*

Bisimulation makes no use of the history of a run, and ignores the concurrency, and thus is definable on states of the TSCs, as is usually done. The definition can also be cast in game-theoretic terms:

Definition 6.2 *The* bisimulation game *played between Duplicator and Spoiler on two TSCs S and T is played as follows. Positions are pairs (σ, τ) of runs from S and T. The initial position is (s_0, t_0). The two players alternate, with*

Spoiler starting. The rules are:

I *Spoiler chooses one of S or T, say S, and chooses a transition $s_n \xrightarrow{a_n} s_{n+1}$. Duplicator must respond in the other system with a transition $t_n \xrightarrow{a_n} t_{n+1}$ extending τ, or else she loses.*

II *If either player cannot move, the other wins; if play continues for ever, Duplicator wins.*

S and T are bisimilar iff Duplicator has a winning strategy for the bisimulation game iff Duplicator has a history-free winning strategy.

Since modal logic characterizes bisimulation, and IFML includes modal logic, it is immediate that $\sim_{\mathrm{IFML}}$ implies $\sim_{\mathrm{b}}$.

A stronger notion of equivalence is obtained [7,13] by requiring the equivalence to preserve the concurrency relation between matching events. The following formulation is not the original definition, but is equivalent in our framework:

Definition 6.3 *R is a* history-preserving bisimulation (hpb) *if*

A $(s_0, t_0) \in R$

C *if $(\sigma, \tau) \in R$ and $\sigma' = \sigma \xrightarrow{a} s$ is a run, then there is t such that $\tau' = \tau \xrightarrow{a} t$, and transitions i and j in σ' are concurrent iff transitions i and j in τ' are concurrent, and $(\sigma', \tau') \in R$; and symmetrically.*

and we write $S \sim_{\mathrm{hpb}} T$ if there is an hbp between S and T.

and there is the obvious analogous game characterization.

Hpb detects at least some true concurrent features; for example, it distinguishes $a.b + b.a$ from $a|b$. However, it has been argued [6,5] that hpb and similar relations such as local/global cause equivalence are really about causality, not about concurrency, and that true concurrency is more correctly captured by the stronger equivalences. The development in this paper will provide further backing to such a view.

The first, initially discouraging, result is that hpb can make distinctions that IFML cannot.

Theorem 6.4 $\sim_{\mathrm{IFML}} \not\subseteq \sim_{\mathrm{hpb}}$

Proof. Consider the following systems:

$$\begin{tikzcd} \bullet \arrow[r, "b"] & \bullet \\ \circ \arrow[u, "a"] \arrow[r, "b"] \arrow[ur, phantom, "\mathcal{C}"] & \bullet \arrow[u, "a"] \end{tikzcd} \qquad \begin{tikzcd} \bullet & \bullet \arrow[l, "b"'] \arrow[r, "b"] & \bullet \\ & \circ \arrow[u, "a"] \arrow[r, "b"] \arrow[ur, phantom, "\mathcal{C}"] & \bullet \arrow[u, "a"] \end{tikzcd}$$

These systems are not hpb, but it may be verified by exhaustive checking that no IFML formula distinguishes them. □

This example will suggest later a possible modification to the definition of IFML; for the present, we continue with the investigation.

It would be surprising if hpb were finer than IFML-equivalence, and indeed it is not, although this is not quite so easy to demonstrate.

Theorem 6.5 $\sim_{\mathrm{hpb}} \not\subseteq \sim_{\mathrm{IFML}}$

Proof. The simplest counter-example we have at present is rather complex to draw in full, so we will give a combined graphical and syntactic description. Let A and C be the two systems

$$\begin{tikzcd}[row sep=small, column sep=small]
 & & 11 \\
 & 1 \arrow[ur, "b_1"] \arrow[r, "b_2"] & 12 \\
\circ \arrow[ur, "a_1"] \arrow[dr, "a_2"'] & & \\
 & 2 \arrow[r, "b_1"] \arrow[dr, "b_2"'] & 21 \\
 & & 22
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}[row sep=small, column sep=small]
 & & 11 \\
 & 1 \arrow[ur, "d_1"] \arrow[r, "d_2"] & 12 \\
\circ \arrow[ur, "c_1"] \arrow[dr, "c_2"'] & & \\
 & 2 \arrow[r, "d_1"] \arrow[dr, "d_2"'] & 21 \\
 & & 22
\end{tikzcd}$$

and let P be their concurrent composition, which is a pyramid with 16 distinct final states on the square face. The systems S and T are formed by adding an e transition to some of these final states, as indicated by the following matrix in which the columns are the A states $11, 12, 21, 22$, the rows are the C states $11, 12, 21, 22$, and the entries indicate the presence of an e transition in the given systems.

$$\begin{array}{cccc}
- & ST & - & S \\
ST & - & S & T \\
S & T & ST & - \\
- & S & - & ST
\end{array}$$

It may be verified (and has been checked with the Edinburgh Concurrency Workbench!) that S and T are strongly bisimilar, and since the concurrency relations are the same, they are also history-preserving bisimilar. However, the following IFML formula is true of S but not of T:

$$[a]_\alpha \langle b \rangle_\beta [c]_{\gamma/\alpha\beta} \langle d \rangle_{\delta/\alpha\beta} \langle e \rangle \mathrm{tt}.$$

(This is because in S, Eloise can choose b_1 after Abelard's a_1 and b_1 after Abelard's a_2; then she can choose d_2 after c_1 and d_1 after c_2, without depending on a, and she ends up in a state with an e transition. In T, on the other hand, no such uniform choice of d exists.) □

A stronger equivalence from concurrency theory is *hereditary (or strong) history-preserving bisimulation (hhpb)* [2,10]. Its relational characterization is

Definition 6.6 *R is a* hereditary history-preserving bisimulation (hhpb) *if*

A $(s_0, t_0) \in R$

B *if* $(\sigma,\tau)\in R$ *and* $\sigma'=\sigma\xrightarrow{a}s$ *is a run, then there is* t *such that* $\tau'=\tau\xrightarrow{a}t$ *and* $(\sigma',\tau')\in R$*; and symmetrically;*

D *if* $(\sigma = s_0 \xrightarrow[\alpha_0]{a_0} \ldots s_n, \tau = t_0 \xrightarrow[\beta_0]{a_0} \ldots t_n) \in R$*, and transition* α_i *is* backwards enabled *in* σ*, meaning that* α_i *is concurrent with every later* α_j*, then* β_i *is backwards enabled in* τ *and* $(\sigma',\tau')\in R$*, where* σ' *is obtained from* σ *by using the TSC diamond axioms to push* α_i *to the end, and then deleting* α_i*, and similarly* τ' *is obtained from* τ *by likewise 'backtracking'* β_i*; and symmetrically.*

The rather complex looking clause D is nothing more than undoing the latest action in some concurrent component; viewing a run as a partial order, rather than a sequence, it is simply the deletion of a maximal element.

It is easy to see that clauses B and D imply that hhpb also satisfies clause C of the hpb definition, and so hhpb is finer (and indeed strictly finer) than hpb. The natural game characterization [11] of hhpb is

Definition 6.7 *The* hhpb *game played between Duplicator and Spoiler on two TSCs* S *and* T *is played as follows. Positions are pairs* (σ,τ) *of runs from* S *and* T*. The initial position is* (s_0,t_0)*. The two players alternate, with Spoiler starting. Spoiler may move in two ways, to which Duplicator must respond.*

(i) *Spoiler chooses one of* S *or* T*, say* S*, and chooses a transition* $s_n\xrightarrow{a_n}s_{n+1}$*. Duplicator must respond in the other system with a transition* $t_n\xrightarrow{a_n}t_{n+1}$ *extending* τ*, or else she loses.*

(ii) *Alternatively, Spoiler chooses* S *or* T *(say* S*), and a transition* $s_i\xrightarrow{a_i}s_{i+1}$ *in* σ *which is backward-enabled. He then 'backtracks' along this transition, as in the relational definition. Duplicator must then respond by backtracking the ith transition in the other system; if this transition is not backwards enabled, she cannot move.*

(iii) *If either player cannot move, the other wins; if play continues for ever, Duplicator wins.*

Hhpb looks like a good candidate for comparison with IFML. For the same reasons as hpb, hhpb can distinguish systems that IFML cannot; but one might wonder whether hhpb is finer than IFML-equivalence (for our restricted models). We have a counter-example for infinite-branching models, but for image-finite models we have not so far constructed a counter-example (or proved the assertion). We make the

Conjecture 6.8 $\sim_{\mathrm{hhpb}} \not\subseteq \sim_{\mathrm{IFML}}$

(As an illustration of how hhpb is stronger than hpb, and how it is intuitively related to IFML, note that the two systems of Theorem 6.5 are distinguished by the formula

$$[a]\langle b\rangle[c]\langle d\rangle Ⓑ @ [a]\langle b\rangle\langle e\rangle\mathrm{tt}$$

of the characteristic logic [11] for hhpb (where @ is the modality of backtracking an a action). We shall discuss in a later article the nature of the relationship between this formula and the IFML formula.)

In order to find equivalences within concurrency that are stronger than IFML, it is necessary to introduce 'coherence' requirements, as studied in for example [4]. The requirement we need is in fact somewhat stronger than the requirement studied there, so the induced equivalence, which we call *strictly coherent hereditary history-preserving bisimulation (schhpb)*, is somewhat stronger than Cheng's *strong coherent history-preserving bisimulation*.

Definition 6.9 *R is a* strictly coherent hereditary history-preserving bisimulation (schhpb) *if*

The clauses of hhpb, together with

E if $(\sigma\alpha, \tau\alpha\prime) \in R$ and $(\sigma\beta, \tau\beta') \in R$ and $\alpha \; \mathcal{C} \; \beta$, then $\alpha' \; \mathcal{C} \; \beta'$ and $(\sigma\alpha\beta, \tau\alpha'\beta') \in R$, and symmetrically.

Theorem 6.10 *If S and T are schhpb, then they are IFML-equivalent.*

Proof. (Sketch) Let Φ be an IFML formula such that $S \models \Phi$. We shall use the schhpb relation and Eloise's winning uniform strategy for $S \vdash \Phi$ to allow her to win $T \vdash \Phi$.

Suppose that in the model-checking games we have reached positions $\sigma \vdash \Psi$ and $\tau \vdash \Psi$. If it is Abelard's turn to move in T, Eloise copies his move to S using the schhpb. If it is Eloise's turn to move, her move in T is given by taking her move in S and mapping it to T via the schhpb. This gives a winning strategy in T.

Using the hereditary and coherent properties of the schhpb, one can show inductively that when Eloise chooses a matching transition, she can do so uniformly in its concurrent events; and therefore that if her S strategy is uniform, she can construct her T strategy to be uniform. □

7 Alternatives to IFML?

The fact that all the concurrent equivalences (apart from bisimulation itself) distinguish systems that IFML does not, is unsatisfactory. Upon inspection of the counter-example of Theorem 6.5, one can see that this is due to a rather simple mismatch between the expressivity of the concurrent logics and IFML: the concurrent logics can express 'a followed by a concurrent b', 'a followed by a dependent b', and also 'a followed by choice of concurrent and dependent b'. IFML, on the other hand, can express 'a followed by a concurrent b', and 'a followed by a dependent b and no concurrent b', but cannot distinguish the case where there is a dependent b as well as a concurrent b.

It is possible to make a small change to the semantics of IFML which addresses this issue. Let us call the result IFMLd (IFML with explicit dependence), defined by the following change to the model-checking game of Defn 3.4:

Definition 7.1 *The IFMLd game is as for IFML except that:*

- *At* $\rho = s_0 \xrightarrow{a_0} \ldots \xrightarrow{a_{n-1}} s_n \vdash \langle b \rangle_{\beta/\alpha_{i_1},\alpha_{i_2},\ldots,\alpha_{i_m}} \Psi$ *(resp.* $\vdash [b]_{\beta/\alpha_{i_1},\alpha_{i_2},\ldots,\alpha_{i_m}} \Psi$*), Eloise (resp. Abelard) chooses a transition* $s_n \xrightarrow{b} t$ *that is concurrent with all the transitions* α_{i_j} and not concurrent with any other transition α_k, *and the new position is* $\rho \xrightarrow[\beta]{b} t \vdash \Psi$.

That is, choices in modalities are required to be concurrent with previous choices if and only if they are logically independent, rather than just if.

This is superficially attractive, and certainly deals with the example of Theorem 6.5, and we

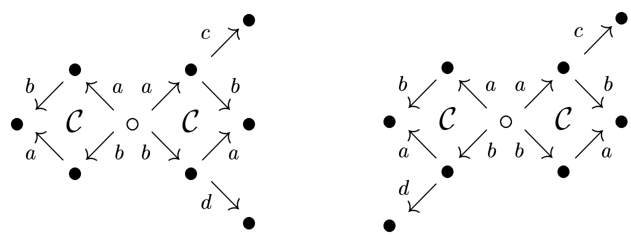
Conjecture 7.2 $\sim_{\text{IFMLd}} \subseteq \sim_{\text{hpb}}$

but have not established this conjecture.

It is also very tempting to conjecture that $\sim_{\text{IFMLd}} \subseteq \sim_{\text{hhpb}}$. Unfortunately, this conjecture fails.

Theorem 7.3 $\sim_{\text{IFMLd}} \not\subseteq \sim_{\text{hhpb}}$

Proof. The following is a notorious example [11] of two systems that are not hhbp (although they are hbp):



It may be verified by exhaustive (and in this case somewhat exhausting) checking that neither IFML nor IFMLd can distinguish them.

It should, however, be pointed out that despite the naturalness of IFMLd, there are some unpleasant consequences of adopting it. In particular, it becomes impossible to express the ordinary modal logic formula $[a]\langle b \rangle \Phi$, where the choice of b *may* depend on a, if a and b happen to be concurrent. (It is for this reason that Conjecture 7.2 is not the simple result one would like.)

8 Conclusion

We have shown that it is possible to define a modal version of the Hintikka–Sandu independence-friendly logic, and that such a logic naturally requires true concurrent models. We have looked at the relationship between the induced equivalence and the equivalences associated with true concurrent models. The results so far indicate that although there is a natural connection, it is not as clean as one would like; however, we are hopeful that further work will throw more light on this. We expect in the full version of this paper to settle all the issues explicitly labelled as conjectures; but we think it will take a more substantial effort to complete the analysis. There are intriguing questions about the exact relationship between backtracking (as used in hhbp), and uniformity (as used in schhpb and in IFML), and we suspect that these questions may provide a useful notion of Ehrenfeucht–Fraïssé game for independence logics. (To coin a slogan, the art of independence is in doing second-order things without appearing to do so.) In turn, independence logics may give new insight into the complexity of the concurrent equivalences.

9 Acknowledgements

The first author is supported by EPSRC Advanced Research Fellowship AF/100690; the second author is supported by EPSRC Research Grant GR/M84763 'Dimensions of Concurrency'. We thank the referees for helpful comments.

References

[1] R. Alur, T. Henzinger and O. Kupferman, Alternating-time temporal logic, in *Proc. 38th FOCS (1997)*, 100–109.

[2] M. A. Bednarczyk, Hereditary history preserving bisimulations, manuscript (1991).

[3] J. C. Bradfield, Independence: logics and concurrency, *Proc. CSL 2000*, LNCS **1862** 247–261 (2000).

[4] A. Cheng, *Reasoning About Concurrent Computational Systems*, Ph.D. thesis, BRICS DS–96–2, Univ. Aarhus (1996).

[5] S. Fröschle, *Decidability and coincidence of equivalences for concurrency.* Ph.D. thesis, Univ Edinburgh, forthcoming.

[6] S. Fröschle and T. Hildebrandt, On plain and hereditary history-preserving bisimulation. *Proc. MFCS '99*, LNCS **1672**, 354–365 (1999).

[7] R. van Glabbeek and U. Goltz, Equivalence notions for concurrent systems and refinement of actions, *Proc. MFCS '89*, LNCS **379** (1989).

[8] J. Hintikka and G. Sandu, A revolution in logic?, *Nordic J. Philos. Logic* **1**(2) 169–183 (1996).

[9] W. Hodges, Compositional semantics for a language of imperfect information, *Int. J. IGPL* **5**(4), 539–563.

[10] A. Joyal, M. Nielsen and G. Winskel, Bisimulation from open maps, *Inform. and Comput.*, **127**(2) 164–185 (1996).

[11] M. Nielsen and C. Clausen, Bisimulations, games and logic, in Karhumaki, Maurer and Rozenberg (eds), *Results and Trends in Theoretical Computer*

Science: Colloquium in Honour of Arto Salomaa, LNCS **812** 289-305 (1994). Also as BRICS report RS-94-6.

[12] A. Pietarinen, *Games logic plays. Informational independence in game-theoretic semantics.* D.Phil. thesis, Univ Sussex (2000).

[13] A. Rabinovitch and B. Trakhtenbrot, Behaviour structures and nets. *Fund. Inf.* **11**(4) (1988).