Independence: logics and concurrency

J. C. Bradfield

Laboratory for Foundations of Computer Science, School of Informatics, King's Bldgs, University of Edinburgh, Edinburgh EH9 3JZ, UK. Email jcb@inf.ed.ac.uk

Abstract: We consider a variety of ways in which independence-friendly logic as introduced by Hintikka and Sandu can be applied to modal logics for concurrent and distributed systems.

Key words: Independence, branching quantifiers, modal logic, fixpoints

1 Introduction

Modal and temporal logics have a long history as system specification languages in computer science, and computer scientists' study of temporal logic has generated many interesting theoretical developments, as well as many important practical advances. The logics in question typically describe properties of execution paths of systems, either explicitly, as in LTL and CTL, or via a 'next-step' operator and fixpoints, as in the modal mu-calculus.

Another major issue in computer science is concurrency. The theoretical and practical analysis of concurrent and distributed systems also has a long history. Probably the most successful approach to understanding concurrency at a fundamental semantic level is the use of 'independence models', in which certain events are stated to be independent of other events, and the associated partial order semantics, in which a partial order of causality is established between events. There have also been several major practical advances in exploiting partial order structure, such as stubborn sets and sleep sets [Val91] and unfolding techniques [McM92].

Naturally, one wishes to have modal and temporal logics for concurrent systems. There are several ways to apply the paradigm of normal temporal logic, typically by working on event structures, or similar models, and having explicit operators representing concurrency or independence: see [PK95] for a survey of such logics. In some models, and therefore in the associated logics, the notion of independence is abstract; in others, it is more concrete, perhaps derived from a notion of location; but in all cases it inheres in the model rather than the logic.

However, the notion of logical independence, as in Henkin quantifiers and independence-friendly logic, also seems quite natural when specifying concurrent systems, but which has not, with one notable exception, received much attention in concurrency theory.

In 1961, Leon Henkin [Hen61] introduced the quantifier $\begin{smallmatrix}\forall x \, \exists y\\ \forall u \, \exists v\end{smallmatrix}$, which is intended to mean that the choice of y depends only on x, *not* on u; and similarly the choice of v depends only on u; a formal semantics is given by using suitable Skolem functions.

The Henkin quantifier and its generalizations received some attention: the main papers were at the beginning of the 70s, by William Walkoe [Wal70] and Herbert Enderton [End70]; more recently Krynicki, Mostowski and others have also investigated Henkin quantifiers, and a survey of the work up to the mid 90s can be found in [KrM95]. More recently, Georg Gottlob has looked at generalized quantifiers, including $\begin{smallmatrix}\forall \exists\\ \forall \exists\end{smallmatrix}$, for capturing complexity classes.

This work is primarily mathematical logic; but there is also a lot of work on branching quantifiers in philosophical and linguistic logic; for much of this work the stimulant (in some cases 'irritant' might be a more appropriate term) is Jaakko Hintikka's programme of game semantics for logic, in which branching quantification is naturally represented by imperfect information in the satisfaction game. In one well known paper, Hintikka and Gabriel Sandu [HiS96] suggested that the absence of branching quantifiers from Frege's logic was a bad decision, and suggested that their reintroduction might herald 'a revolution in logic' (albeit with a question mark in the title). Whether one is a revolutionary or a conservative, 'independence-friendly first-order logic' (IF-FOL) is an interesting, and natural [Bar79,San93] logic. For example, when considering the development of systems by several designers working independently, the notion of independent choice seems natural, and may reasonably form part of any language for discussing properties of such designs. At a more concrete level, different components of a distributed system cannot be assumed to have full knowledge of other components, and so cannot make fully informed choices.

In a preliminary paper [Bra00] presented at CSL 2000, I suggested that the use of modalities based on Henkin quantifiers gives an approach to independence in concurrency which complements the model-based independence usually used, and in particular that a temporal fixpoint logic based on Henkin quantifiers could be defined. In subsequent articles, I studied modal logics using a Hintikka–Sandu style independence, but mentioned fixpoint logics only in passing. In this article, I revise and extend the ideas in [Bra00] on Henkin modal logic, and bring together the Henkin and IF style approaches, considering in particular some of the more subtle design issues involved in producing an independence-friendly modal mu-calculus.

2 Henkin quantifiers

By FOL^+ we mean first-order logic *in positive form*: that is, negation is applied only to atomic formulae, and both operators of the dual pairs $\vee/\wedge$ and $\exists/\forall$ are considered primitive; FOL denotes the usual logic where negation is not so restricted. $\mathrm{FOL}^+{+}Q$ denotes the logic where the additional operator Q appears only positively. We use similar notation for other logics.

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 $\exists f_1 \dots f_n. \forall x_1 \dots x_m. \phi[f_i(y_i{\downarrow})/y_i]$, 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 $\exists f, g. \forall x, u. \phi(x, f(x), u, g(u))$.

The following are some basic properties of branching quantifiers.

(1) In the presence of a pairing function, any formula $Q\phi$ where Q is a branching quantifier and ϕ is FOL$^+$ is equivalent to a formula of FOL$^+$+$\begin{smallmatrix}\forall\exists\\\forall\exists\end{smallmatrix}$. [Wal70,End70]. (There is a looser normal form without pairing functions – see [KrM95].)

(2) By definition, any FOL$^+$+$\begin{smallmatrix}\forall\exists\\\forall\exists\end{smallmatrix}$ formula is equivalent to an existential second-order formula.

(3) Moreover, any existential second-order formula is equivalent to a formula of FOL$^+$+$\begin{smallmatrix}\forall\exists\\\forall\exists\end{smallmatrix}$ (in the presence of a pairing function). This is because the formula can be 'unskolemized' using $\begin{smallmatrix}\forall\exists\\\forall\exists\end{smallmatrix}$: for example, the assertion that there exists an injective function which never takes a particular value c, which being true only on infinite structures is not first-order, can be written as $\exists f. \forall x, u. (f(x) = f(u) \Rightarrow x = u) \wedge (f(x) \neq c)$; and this can be written using $\begin{smallmatrix}\forall\exists\\\forall\exists\end{smallmatrix}$ as $\begin{smallmatrix}\forall x \exists y\\\forall u \exists v\end{smallmatrix}. (x = u \Rightarrow y = v) \wedge (y = v \Rightarrow x = u) \wedge (y \neq c)$.

(4) By similar arguments and Fagin's Theorem, on finite structures FOL$^+$+$\begin{smallmatrix}\forall\exists\\\forall\exists\end{smallmatrix}$ expresses NP-hard properties [BlG86]; in fact, as one would expect, FOL+$\begin{smallmatrix}\forall\exists\\\forall\exists\end{smallmatrix}$ captures L$^{\mathrm{NP}}$ [Got97].

2.1 Henkin quantifiers and imperfect information games

An alternative way of giving semantics to branching quantifiers is via games. Recall the Hintikka model-checking game for FOL$^+$: 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 [HiS96], 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 more transparent, Hintikka et al. use a linear syntax in which independence is written explicitly: for example, the Henkin quantifier $\begin{smallmatrix}\forall x \exists y\\\forall u \exists v\end{smallmatrix}$ is written as $\forall x/\emptyset. \exists y/\emptyset. \forall u/\{x, y\}. \exists v/\{x, y\}.$, where the variables following a / are the variables with respect to which the bound variable must be independent. 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. However, these games are not determined, so it is *not* true that Abelard has a winning strategy iff the formula is false. For example, $\begin{smallmatrix}\forall x\\ \exists y\end{smallmatrix} x = y$ (or $\forall x.\, \exists y/x.\, x = y$) is false in any structure with more than one element, but Abelard has no winning strategy.

We note here that this presentation does not include the independent $\vee/\ldots$ and $\wedge/\ldots$ that are part of full IF logic. From the point of view of a concurrency theorist, these operators obscure the more interesting relationship between modalities and concurrency. However, as shown in [Tul04], in the non-concurrent interpretation of IFML, they have a significant effect.

3 Henkin modal logic

For a concurrency theorist, it is natural to see the model-checking game for $\begin{smallmatrix}\forall x & \exists y\\ \forall u & \exists v\end{smallmatrix}\phi$ not as a game of imperfect information, but as a *concurrent* game: at $\begin{smallmatrix}\forall x & \exists y\\ \forall u & \exists v\end{smallmatrix}\phi$ the game splits into two independent concurrent components: in one component play proceeds with $\forall x.\, \exists y.$, in the other with $\forall u.\, \exists v.$, and then the components join to proceed with ϕ. This also seems a natural statement to want to make about concurrent systems: a choice in one component should not even be able to depend on a concurrent choice in another! However, until recently all modal or temporal logics enforced logical dependence of choices.

Example 1 In the children's game Scissors–Paper–Stone, the two players (say Abe and Elly) each put one hand behind their back, and make it either open (Paper), a fist (Stone), or a V-sign (Scissors). The two players then *simultaneously* bring forward their hands. The round is won according to the rules: Scissors cut Paper, Paper wraps Stone, Stone blunts Scissors (if both players choose the same object, the round is drawn). Can we ask the question 'can Elly always win?'?

If we formalize the game by seeing Abe as choosing between the three actions $A = \{sc_A, pa_A, st_A\}$, and similarly for Elly, and viewing the game as their independent concurrent composition with final states that satisfy *Ewins* when Elly wins, we can approximate the question by $[A]\langle E\rangle \mathit{Ewins} \wedge \langle E\rangle[A]\mathit{Ewins}$. This expresses that Elly can win on all interleavings; it is (correctly) false, but it is false for the wrong reason: that $\langle E\rangle[A]\mathit{Ewins}$ fails, means that after Elly chooses, Abe can, with knowledge of that choice, make her lose, which mis-models the situation.

If we formalize the simultaneity by using a synchronous concurrent composition, there is, in normal modal or temporal logics, no way to ask the question at all, since only one action (the simultaneous choice) happens.

Of course, with an eye on Henkin quantifiers, the obvious answer is to define a modality $\begin{smallmatrix}[\,]\\ \langle\,\rangle\end{smallmatrix}$ so one can write $\begin{smallmatrix}[A]\\ \langle E\rangle\end{smallmatrix} \mathit{Ewins}$, with the intended meaning.

A few years ago, Alur, Henzinger and Kupferman [AHK97] implicitly took this approach in their Alternating Temporal Logic, without mentioning Henkin quantifiers: they express, rather, their logic directly in terms of games and strategies. We shall now define a simple logic of Henkin modalities; a fragment of this provides a generalization of ATL, and also includes several other distributed logics.

3.1 A distributed system model

One problem when considering modal logics for concurrency is deciding which of the many system models to adopt. Our purpose in this part is to investigate logical independence rather than model independence, and so we shall, for now, consider only normal distributed models without explicit 'true concurrency'; the relationship between Henkin-style independence and true-concurrency-style independence is not quite as simple as one might hope, and will be reviewed in a later section. For usual modal logic, the most basic model is the labelled transition system. However, our logic is inherently a logic of distributed systems, and we shall therefore take a model with distribution built in. The major question now is, whether the structure of systems is static, as in Petri nets, products of automata, etc.; or dynamic, as in general process algebras. There is no principled reason to banish dynamic structure, but it raises several problems and is technically more complex to deal with even at the basic level. Therefore we shall initially adopt a simple model in which systems have a fixed number of communicating components, with no hierarchical structure. The communication will be synchronous, but not otherwise restricted in form.

Definition 2 Let Act be a set of action labels, and let $Act_\perp = Act \cup \{\perp\}$ where $\perp$ (the idle action, or non-action) is a label not in Act. An *n-ary synchronization algebra on Act* is a function $\mathscr{S} : Act_\perp^n \to Act_\perp$. Let $\mathscr{T}$ be the set of Act-labelled transition systems; given $T = (S, \longrightarrow) \in \mathscr{T}$, let $\xrightarrow{\perp}$ be the identity relation on the states of T. We will let T denote both the transition system and its state set.

An *n-component system (over $\mathscr{S}$)* $T = \|_{1 \le i \le n} T_i$ is a product of n Act-labelled transition systems $T_1, \ldots, T_n$, the product transition relation being given by

$$(s_1, \ldots, s_n) \xrightarrow{a} (s'_1, \ldots, s'_n) \text{ for } a \in Act \text{ iff}$$
$$\exists \alpha_1, \ldots, \alpha_n \in Act_\perp . (\forall 1 \le i \le n.\, s_i \xrightarrow{\alpha_i} s'_i) \wedge \mathscr{S}(\alpha_1, \ldots, \alpha_n) = a.$$

Where $\mathscr{S}$ is understood, we may write $\bigotimes_i \alpha_i$ for $\mathscr{S}(\alpha_1, \ldots, \alpha_n)$.

Note that the $\perp$ symbol is slightly overloaded: within components (i.e. in the domain of $\mathscr{S}$), it refers to the non-action or idle action; in the global system (i.e. as a result of $\mathscr{S}$) it indicates the absence of a global action, and it is not to be interpreted as an idle action.

In order to present examples, we shall use a CCS style notation for the components, defining processes P by means of action prefix $a.P$, non-deterministic choice $P + Q$, and recursive definition $P = \ldots P \ldots$. Given n components P_i, the product system will be written $\|_i^{\mathscr{S}} P_i$, dropping $\mathscr{S}$ where understood.

3.2 Henkin modalities

We now define Henkin modal logic ($^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}$) on such a system thus: in addition to ML^+, that is, basic modal logic in positive form, we have the *concurrent modalities* of the form $\bigotimes_{i=1,\ldots,n} \mathsf{Q}_i^1(A_i^1) \ldots \mathsf{Q}_i^m(A_i^m)$, where $\mathsf{Q}_i^j(A_i^j)$ is either $[A_i^j]$ or $\langle A_i^j \rangle$, and $A_i^j \subseteq Act_\perp$; the *length* of the modality is m. We shall sometimes use '$-$' to mean 'Act' in modalities. These modalities are given a semantics in terms of the corresponding first-order Henkin quantifier. This is notationally intricate to write out, and there are one or two wrinkles to consider, so before giving the formal definition we shall look at the prototypical examples.

Note that by definition ${}^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}$ does not include the duals of the Henkin modalities.

3.3 Handling deadlock

Firstly, consider the modality $\substack{[A]\\ \langle E\rangle}\phi$, or formally $[A] \otimes \langle E\rangle\phi$, used in our first motivating example of Scissors–Paper–Stone. The intention is that for every A move, there is a uniformly chosen E move, such that the resulting global move results in a state satisfying ϕ. What if there is no A move, and thus no resulting global move? Is the formula vacuously true (as $[]$), or vacuously false (as $\langle\rangle$)? What if there is no E move? These are easy questions, since the natural answer goes with the $[]$ or $\langle\rangle$ respectively; but what if there are possible A and E moves, but they do not synchronize to produce a resulting global move? For example, we might choose to implement the notion of 'draw' in Scissors–Paper–Stone by saying that if the two players choose the same action, there is no resultant move, and so the game is effectively not played. Resolution of this issue appears to require breaking the symmetry, and as suggested by the example, it should be broken by requiring a global move to exist, giving the following semantics: $\substack{\forall s_1'\\ \exists s_2'}.s_1 \xrightarrow{A} s_1' \Rightarrow (s_2 \xrightarrow{E} s_2' \wedge (s_1, s_2) \xrightarrow{A\otimes E} (s_1', s_2') \wedge (s_1', s_2') \vDash \phi)$. However, this is not really symmetry breaking, for the symmetry has already been broken by the decision to use Henkin quantifiers, which privilege the existential player compared to the pure game-theoretic interpretation of independence-friendly logic. (A Henkin quantified sentence is classically true iff Eloise wins its IF equivalent; and classically false if either Abelard wins the IF equivalent or the IF equivalent is undetermined.)

For a second, and more important, example, consider the semantics of the length 2 Henkin modality $\substack{[]\\ []}\substack{\langle\rangle\\ \langle\rangle}$ on a 2-component system $T = T_1 \|_{\mathscr{S}} T_2$, where again $\substack{[A]\\ [C]}\substack{\langle B\rangle\\ \langle D\rangle}$ is Henkin notation for our formal $([A]\langle B\rangle \otimes [C]\langle D\rangle)$. At first sight, the obvious semantics to give is

$$(s_1, s_2) \vDash \begin{matrix}[A_1]\\ [A_2]\end{matrix}\begin{matrix}\langle B_1\rangle\\ \langle B_2\rangle\end{matrix}\phi$$

holds iff

$$\begin{matrix}\forall(\alpha_1, s_1') \in A_1 \times T_1\, \exists(\beta_1, s_1'') \in B_1 \times T_1\\ \forall(\alpha_2, s_2' \in A_2 \times T_2\;\; \exists(\beta_2, s_2'') \in B_2 \times T_2\end{matrix}(s_1, s_2) \xrightarrow{\alpha_1\otimes\alpha_2} (s_1', s_2') \Rightarrow ((s_1', s_2') \xrightarrow{\beta_1\otimes\beta_2} (s_1'', s_2'') \wedge (s_1'', s_2'') \vDash \phi)$$

for there is no such problem as before: in the first step, all moves are Abelard's, and in the second all are Eloise's. However, this definition is not entirely consistent with our argument for the previous example: it permits the absence of a global move in the first step in the case where moves exist in each component. This matches the usual definition of the modal box, but for the reasons given above, in the case of Henkin modalities it is preferable to abandon that intuition, and require a global move to exist. In the majority of cases (in fact, all that we will consider), a global move always exists, and so the above definition suffices for intuitive understanding and ease of proof. However, formally we use the general definition given below, which does require the existence of a global move.

3.4 Semantics of Henkin modalities

The discussion above suffices, we hope, to motivate the following general definition:

Definition 3 The semantics of $\phi' = \bigotimes_{i=1,\ldots,n} \mathsf{Q}_i^1(A_i^1)\ldots\mathsf{Q}_i^m(A_i^m).\phi$ is as follows. Let $\mathcal{H}$ be the Henkin quantifier

$$\begin{array}{ccc} \hat{\mathsf{Q}}_1^1(s_1^1,\alpha_1^1) \in T_1 \times A_1^1 & \ldots & \hat{\mathsf{Q}}_1^m(s_1^m,\alpha_1^m) \in T_1 \times A_1^m \\ \vdots & & \vdots \\ \hat{\mathsf{Q}}_n^m(s_n^1,\alpha_n^1) \in T_n \times A_n^1 & \ldots & \hat{\mathsf{Q}}_n^m(s_n^m,\alpha_n^m) \in T_n \times A_n^m \end{array}$$

where $\hat{\mathsf{Q}}_i^j$ is $\forall$ or $\exists$ as Q_i^j is $[]$ or $\langle\rangle$.

For each j, let $\psi_\forall^j$ be $\bigwedge_{\{\,i|\mathsf{Q}_i^j=[]\,\}}(s_i^{j-1} \xrightarrow{\alpha_i^j} s_i^j)$, and let $\psi_\exists^j$ be $\bigwedge_{\{\,i|\mathsf{Q}_i^j=\langle\rangle\,\}}(s_i^{j-1} \xrightarrow{\alpha_i^j} s_i^j)$.

Let $\boldsymbol{s}^j$ be $(s_1^j,\ldots,s_n^j)$. Let ψ^j be $\psi_\forall^j \Rightarrow (\psi_\exists^j \wedge \boldsymbol{s}^{j-1} \xrightarrow{\bigotimes_{i=1,\ldots,n}\alpha_i^j} \boldsymbol{s}^j \wedge \psi^{j+1})$ for $j = 1,\ldots,n$, and ψ^{n+1} be $\boldsymbol{s}_n \vDash \phi$.

Then we define $\boldsymbol{s}_0 \vDash \phi'$ iff $\mathcal{H}\psi^1$.

3.5 Examples of length 1 modalities

Let ${}^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}_n$ denote the sublogic of ${}^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}$ where the modalities have length at most n. ${}^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}_1$ with the addition of fixpoints (see later) includes many existing distributed modal logics.

Example 4 The indexed transition systems of Andersen [And93] are systems $\|^{\mathscr{S}}_{1\le i\le n} T_i$, where $\mathscr{S}(\alpha_1,\ldots,\alpha_n) = a$ iff there is some k such that $\alpha_k = a$ and all other α_i are $\bot$, and $=\bot$ otherwise. (The systems move one component at a time.) Then in Andersen's 'polyadic mu-calculus', the indexed modality $[a]_k$ is in our notation $\bigotimes_i[\alpha_i]$ where $\alpha_k = a$ and the other α_i are all $\bot$.

Example 5 The distributed net systems used by Huhn, Niebert and Wallner [HNW97] are systems $\|^{\mathscr{S}}_{1\le i\le n} P_i$ where each P_i is a sequential transition-labelled Petri net, and $\mathscr{S}$ defines the synchronization of common transitions. In our setting, we need the label of an arrow to represent the identity of the net transition t; so to represent a net transition label a we use instead the set A of transitions labelled with a. The logic of [HNW97] is defined locally via an event structure style semantics, so that the basic modality is $\langle a\rangle_J\phi$, where $J \subseteq \{1,\ldots,n\}$; this is true at a *local state* of P_i, for $i \in J$, if some a-labelled transition (which must share some location with J) can fire as an immediate action of the J processes, possibly with non-J actions happening first, and then ϕ holds.

Such a modality is not expressible in ${}^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}_1$ without fixpoints, since we have an interleaving, global semantics, but with fixpoints it is just $\mu Z.\,\langle A\rangle\phi \vee (\bigotimes_i\langle A_i\rangle)Z$, where A is the set of a-labelled transitions, and A_i is $\bot$ for $i \in J$ and $Act_\bot$ otherwise.

The previous examples are merely using the idea of location; to use the power of independent quantification, we need:

Example 6 The alternating transition systems of [AHK97] are an unlabelled setting. An ATS is a global state space Q, shared by n agents P_i. Each agent has a transition function

$\delta_i : Q \to 2^{2^Q}$. At a state q, each agent chooses $Q_i \in \delta_i(q)$. It is required that for all such choices, $\bigcap_i Q_i$ is a singleton set $\{q'\}$, and this determines the next state of the system. Thus the intuition is that each agent chooses its desired successors, independently of the others, and the system moves to the one state they all desire.

To encode this in our setting, it is easiest to make the choice an explicit local transition happening before the move to the next state. So we take each agent P_i to be a transition system with state space $Q \cup 2^Q$, with transitions $q \xrightarrow{\tau} Q_i$ (where τ is a dummy label) for every $Q_i \in \delta_i(q)$, and $Q_i \xrightarrow{q'} q'$ for every $q' \in Q_i$ (note that q' is also being used as a label). The synchronization algebra $\mathscr{S}$ is then the diagonal on $Q \cup \{\tau\}$, and one move of the original ATS corresponds to two moves of our system $\|_i^{\mathscr{S}} P_i$. Thus the reachable states of our system have the form $(q, q, \ldots, q)$ or $(Q_1, \ldots, Q_n)$, and the requirements on the ATS transition function mean that there is exactly one transition from any reachable $(Q_1, \ldots, Q_n)$, to the unique $(q', q', \ldots, q')$ with $q' \in \bigcap_i Q_i$.

The Alternating Temporal Logic of [AHK97] is like CTL, except that the path quantifiers are not just $\forall$ and $\exists$, but have the form $\langle\langle E \rangle\rangle$, where $E \subseteq \{1, \ldots, n\}$. The interpretation is that, given a path formula ϕ (for example, $\mathbf{F}\psi$, 'eventually ψ'), $\langle\langle E \rangle\rangle\phi$ is true if the agents E can choose their successors to make ϕ hold on all paths, regardless of how the other agents $\overline{E}$ choose their successors. In other words, the E agents can win the path game defined by ϕ.

This logic is then in turn defined as a fragment of a suitable mu-calculus, just as CTL is defined in the usual modal mu-calculus. The [AHK97] mu-calculus has the basic modality $\langle\langle E \rangle\rangle \bigcirc \phi$, where $\bigcirc$ is the 'next' operator, interpreted as just described.

Now in our encoding, this modality is just $(\bigotimes_i \mathsf{Q}_i)\langle - \rangle\phi$, where Q_i is $\langle \tau \rangle$ for $i \in E$, and $[\tau]$ otherwise.

3.6 Adding fixpoints

We have already used fixpoint notation above; to justify this, it suffices to note the following fact:

Proposition 7 A Henkin modality defines a monotone operator on sets of global states. Therefore least and greatest fixpoint operators can be added to ${}^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}$ as in the normal modal mu-calculus: call this $\mu{}^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}$. $\square$

The addition of fixpoints adds the usual ability to express temporal properties, such as 'eventually', 'always', 'infinitely often' and so on. Because we are defining interpretations as sets of global states, and the Henkin modalities are simply operators on sets of global states, $\mu{}^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}$ is still relatively easy to model-check:

Proposition 8 The usual simple complexity upper bound applies: given a $\mu{}^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}$ formula ϕ of length m and fixpoint alternation depth d, and a system of size n (meaning here the number of global states plus the number of transitions), the complexity of determing whether $s \vDash \phi$ is $O(m \cdot H \cdot n^d)$, where H is the cost of evaluating a basic modality or boolean. $\square$

In normal modal mu-calculus, H is $O(n)$; we consider now what H is for Henkin modalities.

3.7 Cost of Henkin modalities

The cost of model-checking depends on the cost of checking single modalities. For length 1 modalities, as in ATL, this is not too bad, although somewhat worse than ordinary modalities.

Proposition 9 The cost of evaluating a Henkin modality of length 1 is $O(n^3)$.

Proof. To check $s_1 \| s_2 \in [\![([A]\otimes\langle B\rangle)\phi]\!]$, it suffices to try in turn each of the $O(n)$ possible B-successors of s_2 and check whether all the so chosen $A \otimes B$-successors are included in $[\![\phi]\!]$. $\square$

This means that $\mu{}^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}_1$ is not significantly harder (in theory rather than practice terms) than than normal modal mu-calculus. However, when we move to real Henkin modalities, things become more expensive. As we know that first-order Henkin modalities are NP-complete, we might expect this; but we might also hope that the restricted quantification involved in modal logic reduces the complexity. Unfortunately, this is not the case:

Proposition 10 Model-checking ${}^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}_2$ is NP-complete.

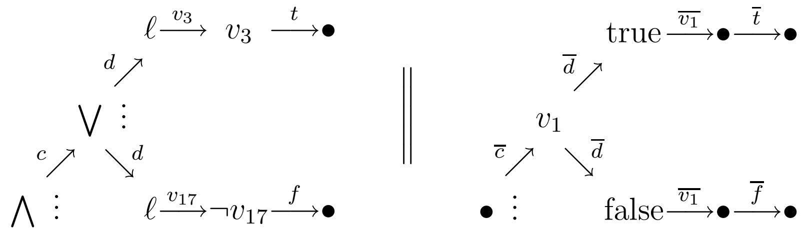
Proof. The problem is in NP, since the usual argument for modal mu-calculus being NP extends by guessing the independent choice functions in the Henkin modalities. (Note that this relies crucially on the absence of negation.)

For the hardness, there is a direct reduction (obtained in discussion with Perdita Stevens) from CNF-SAT, which illustrates the use of the Henkin modalities quite nicely, and so is worth giving in full.

Consider an instance Φ of CNF-SAT: it has the form $\bigwedge_{1\le i\le m} C_i$, where $C_i = \bigvee_{1\le j\le n_i} \ell_{ij}$, and each ℓ_{ij} is v_k or $\neg v_k$ for one of the r variables v_k.

We now define a system with two components: the first represents the formula, the second a choice of variable assignment. Let $D = \sum_{1\le i\le m} c.C_i$, and $C_i = \sum_{1\le j\le n_i} d.L_{ij}$, and $L_{ij} = v_k.t$ if $\ell_{ij} = v_k$, or $L_{ij} = v_k.f$ if $\ell_{ij} = \neg v_k$

Let $A = \sum_{1\le k\le r} \overline{c}.V_k$, and $V_k = \overline{d}.\overline{v_k}.\overline{t} + \overline{d}.\overline{v_k}.\overline{f}$.



Now let $\mathscr{S}$ be the synchronization algebra given by $\alpha \otimes \overline{\alpha} = \tau$, and symmetrically, for every $\alpha \in \{c, d, v_k, t, f\}$, and let $P = D\|^{\mathscr{S}} A$. (In standard CCS notation, this would be $(D \mid A)\backslash\{c, d, v_k, t, f\}$.) Note that the size of this system is at most quadratic in the size of Φ.

Now we claim that $P \vDash [{}^{-}_{-}]\langle{}^{-}_{-}\rangle[\tau]\langle\tau\rangle\mathtt{tt}$ iff Φ is satisfiable.

Firstly, suppose Φ has a satisfying assignment S. In the game for $\left[\begin{smallmatrix}-\\-\end{smallmatrix}\right]\left\langle\begin{smallmatrix}-\\-\end{smallmatrix}\right\rangle$, Eloise plays the following strategy: in the top half, in response to Abelard's choice of C_i, she chooses any L_{ij} made true by S – such a literal exists, by definition of satisfying assignment. In the bottom half, in response to Abelard's choice of V_k, she chooses $\overline{v_k}.\overline{t}$ or $\overline{v_k}.\overline{f}$ according as v_k is true or false in S. The resulting process is then $v_{k'}.(t/f)\|^{\mathscr{S}}\overline{v_k}.(\overline{t}/\overline{f})$ according as ℓ_{ij} is $v_{k'}$ or $\neg v_{k'}$, and as v_k is true or false. If $k' \neq k$, then the process is deadlocked, and so satisfies $[\tau]\langle\tau\rangle\mathtt{tt}$. If $k' = k$, then since Eloise chose L_{ij} to be true in S, the left hand process has t iff the right hand process has $\overline{t}$, and so again the process satisfies $[\tau]\langle\tau\rangle\mathtt{tt}$.

Conversely, if $P \vDash \left[\begin{smallmatrix}-\\-\end{smallmatrix}\right]\left\langle\begin{smallmatrix}-\\-\end{smallmatrix}\right\rangle[\tau]\langle\tau\rangle\mathtt{tt}$, then Eloise's strategy for $\left[\begin{smallmatrix}-\\-\end{smallmatrix}\right]\left\langle\begin{smallmatrix}-\\-\end{smallmatrix}\right\rangle$ defines, in the bottom half, an assignment S, and in the top a selection of one literal L_{ij_i} for each conjunct C_i, such that the literal is made true by the assignment; so S is a satisfying assignment. □

So it appears that the Henkin modality is exponential to check. This has two apparent consequences: it means that one might reasonably argue it is useless, even if there are natural properties to be expressed with it; and it means that the fixpoint alternation depth is no longer the dominant factor in the combined complexity of model-checking.

The first of these may be worrying: although worst-case complexity is not always of any relevance to real life, the structure of this problem suggests that exponential costs might easily be achieved. The second is theoretically interesting, although as usual one can say that fixpoint alternation is irrelevant to practice anyway. However, once we remember that our measure of system size is the global state/transition space, the exponential cost of model-checking the Henkin modality is not quite as bad as it looks, if systems have some certain structure.

Given a system in our framework, define the *local size* to be the maximum of the sizes of the individual components.

Proposition 11 In a k-component system with local size d, evaluating a Henkin modality of length 2 costs $O(d^{d+2k}) = O(2^{(d+2k)\log d})$.

Proof. Brute force exploration of all the possible 'Skolem functions': in each component there are at most d^d possible local strategies in the local $[]\langle\rangle$ game; to check that a candidate k-tuple of local strategies satisfies the formula costs at most $d^k \cdot d^k$. Thus the total cost is d^{d+2k} □

In a loosely coupled system, the global state space is of size d^k; thus as the number of components increases, the exponential of the Henkin modality is absorbed by the exponential of the state space explosion, rather than adding to it.

The fact that the Henkin modality in the worst case dominates fixpoint complexity raises the question of the fixpoint alternation hierarchy for finite models. In the *infinite* case, this question can be solved by extending previous techniques.

Proposition 12 For $\mu^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}_2$ on infinite models (and even on 2-component systems), the fixpoint alternation hierarchy is strict.

Proof. If one considers arithmetic with fixpoints and the Henkin quantifier, the fixpoint alternation hierarchy is strict – see the next section. Then a natural extension of the techniques of [Bra97], in which a Henkin modality is used to encode a first-order Henkin quantifier, transfers this hierarchy to $\mu^{\mathrm{H}}_{\mathrm{M}}\mathrm{L}_2$ on 2-component systems.

However, current techniques do not apply to the case of finite models.

4 IF-FOL and fixpoints

4.1 Fixpoints and the Henkin quantifier

We return now to the general setting of first-order logic. As we remarked in the last proposition, there is no great difficulty in combining a simple Henkin quantifier with fixpoints: ${}^{\forall\,\exists}_{\forall\,\exists}$ is just another monotone operator. Getting a handle on the expressive power is less trivial. Consider LFP^{+}+${}^{\forall\,\exists}_{\forall\,\exists}$, that is, first-order logic with fixpoints in positive form, plus the Henkin quantifier occurring only positively, and consider the structure of arithmetic. The Henkin quantifier itself has Σ^1_1 power, so a formula of the form $\mu(z,Z).\,{}^{\forall x\,\exists y}_{\forall u\,\exists v}\phi$, for first-order ϕ, is at worst a fixpoint over Σ^1_1, which is at worst Σ^{μ}_2 in the normal fixpoint hierarchy (because $\Sigma^1_1 = \Pi^{\mu}_1$: Kleene's theorem). However, it is not immediately obvious that, say, ${}^{\forall x\,\exists y}_{\forall u\,\exists v}\mu(z,Z).\,\phi$ is well-behaved; naively, it might be Σ^1_2. Of course, it is not, because the fixpoint is parametrized on the variables x, y, u, v, and cannot actually refer to the defined Skolem functions. The key to analysing the expressive power is to extend the normal form results of [Lub93,Bra97] with the

Lemma 13 In arithmetic (or other structures with suitable coding power), a Henkin quantifier can be pushed inside a fixpoint operator.

Proof. We use the notation of [Bra97] in the next couple of lemmas, so that the case here slots directly into the proof of [Bra97, Lemma 2], the normal form theorem for mu-arithmetic. In particular, $\langle,\rangle$ is a tupling function, and $(w)_i$ denotes the ith component of the coded tuple w. Consider a formula of the form

$$\begin{matrix}\forall x\,\exists y\\ \forall u\,\exists v\end{matrix}.\tau \in \mu(t,T).\,\phi(x,y,u,v,t).$$

This formula is true iff

$$\begin{aligned}\langle 0,0\rangle \in \mu(w,W).\qquad & (w = \langle 0,0\rangle \wedge {}^{\forall x\,\exists y}_{\forall u\,\exists v}.\langle 1,\langle x,y,u,v,\tau\rangle\rangle \in W)\\ \vee\;\; & ((w)_0 = 1 \wedge \phi'),\end{aligned}$$

where ϕ' is obtained from ϕ by replacing every '$\rho \in T$' by '$\langle 1,\langle x,y,u,v,\rho\rangle\rangle \in W$', and then every x by $((w)_1)_0$, every y by $((w)_1)_1$, ... and every t by $((w)_1)_4$. (Note that although the Henkin quantifier is now syntactically inside the fixpoint, it is only evaluated once in the computation of W.)

The case of a maximal fixpoint is similar.

From this lemma, one obtains a normal form for LFP^{+}+${}^{\forall\,\exists}_{\forall\,\exists}$, and hence

Theorem 14 *The fixpoint alternation hierarchy for LFP^{+}+${}^{\forall\,\exists}_{\forall\,\exists}$ (on arithmetic) is strict.*

Proof.

As corollaries of the normal form and existing theorems [Bra99a,Bra99b] on the power of fixpoints, one obtains

Corollary 15 $\mathrm{LFP}^{+} + {}^{\forall\exists}_{\forall\exists}$ is no more expressive than LFP^{+} (on arithmetic). $\square$

and in fact both these apply to $\mathrm{LFP} + {}^{\forall\exists}_{\forall\exists}$ as well; adding the dual is not problematic. The latter result contrasts sharply with the position on finite models, where a single ${}^{\forall\exists}_{\forall\exists}$ is stronger than fixpoints, unless P = NP.

4.2 Fixpoints and full IF-FOL

Now consider the full independence-friendly logic, with the linear syntax allowing arbitrary specification of independence in quantifiers. We described the Skolem function semantics, and the game semantics of Hintikka. These semantics have the problem that they are not compositional, and in particular they give no meaning to a formula such as $\exists y/x.\,\phi$, which occurs as a subformula of $\forall x.\,\exists y/x.\,\phi$ (alias ${}^{\forall x}_{\exists y}\phi$). Wilfrid Hodges [Hod97], and others, have also given a compositional semantics, which allows us to add fixpoints freely in the usual way (call the result IF-LFP, 'independence-friendly least fixpoint logic'), but at a price: the interpretation of a formula with free variables is no longer just a set of value tuples, but a set of sets of tuples.

We have studied IF-LFP elsewhere [Bra04,BrK05], and shown that it is a highly expressive logic, with a correspondingly high complexity for model-checking. On infinite models, it is at least Σ^1_2 [Bra04], and on finite models it is EXPSPACE-hard to verify, and even includes all of second-order logic [BrK05]. For this article, we wish to use IF-LFP as a meta-language for IF modal mu-calculus. The definition of IF-LFP is unfortunately a little complex, but we give it here for completeness.

Recall the definition of the syntax in section 2.1 – which is, for simplicity, a fragment of the full logic. We will give, using Hodges' terminology, the semantics for this fragment.

Definition 16 Let a structure A be given, with constants, propositions and relations interpreted in the usual way. A *deal* $\vec{a}$ for $\phi[\vec{x}]$ or $\vec{t}[\vec{x}]$ is an assignment of an element of A to each variable in $\vec{x}$. Given a deal $\vec{a}$ for a tuple of terms $\vec{t}[\vec{x}]$, let $\vec{t}(\vec{a})$ denote the tuple of elements obtained by evaluating the terms under the deal $\vec{a}$.

If $\phi[\vec{x}]$ is a formula and W is a subset of the variables in $\vec{x}$, two deals $\vec{a}$ and $\vec{b}$ for ϕ are $\simeq_W$*-equivalent* ($\vec{a} \simeq_W \vec{b}$) iff they agree on the variables not in W. A $\simeq_W$*-set* is a non-empty set of pairwise $\simeq_W$-equivalent deals.

The denotation $[\![\phi]\!]$ of a formula is a set T of *trumps*. If ϕ has n free variables, then $T \in \wp(\wp(A^n))$ – that is, a trump is a set of deals.

- If $(R(\vec{t}))[\vec{x}]$ is atomic, then a non-empty set D of deals is a trump iff $\vec{t}(\vec{a}) \in R$ for every $\vec{a} \in D$.
- D is a trump for $(\phi \wedge \psi)[\vec{x}]$ iff D is a trump for $\phi[\vec{x}]$ and D is a trump for $\psi[\vec{x}]$.
- D is a trump for $(\phi \vee \psi)[\vec{x}]$ iff it is non-empty and there are trumps E of ϕ and F of ψ such that every deal in D belongs either to E or F.
- D is a trump for $(\forall y/W.\,\psi)[\vec{x}]$ iff the set $\{\, \vec{a}b \mid \vec{a} \in D, b \in A \,\}$ is a trump for $\psi[\vec{x}, y]$.

- D is a trump for $(\exists y/W.\,\psi)[\vec{x}]$ iff there is a trump E for $\psi[\vec{x}, y]$ such that for every $\simeq_W$-set $F \subseteq D$ there is a b such that $\{\, \vec{a}b \mid \vec{a} \in F \,\} \subseteq E$.
- D is a trump for $(\neg R(\vec{t}))[\vec{x}]$ iff $\vec{t}(\vec{a}) \notin R$ for every $\vec{a} \in D$.

A trump for ϕ is essentially a set of winning positions for the model-checking game for ϕ, for a given *uniform* strategy, that is, a strategy where choices are uniform in the 'hidden' variables. The most intricate part of the above definition is the clause for $\exists y/W.\,\psi$: it says that a trump for $\exists y/W.\,\psi$ is got by adding a witness for y, uniform in the W-variables, to trumps for ψ.

We now describe the addition of fixpoint operators to IF-FOL. This is slightly intricate, although the normal intuitions for understanding fixpoint logics still apply.

Definition 17 IF-LFP extends the syntax of IF-FOL as follows:
- There is a set $\mathrm{Var} = \{X, Y, \ldots\}$ of fixpoint variables. Each variable X has an arity $\mathrm{ar}(X)$.
- If X is a fixpoint variable, and $\vec{t}$ an $\mathrm{ar}(X)$-vector of terms then $X(\vec{t})$ is a formula.
- Let ϕ be a formula with free fixpoint variable X. ϕ has free individual variables $\vec{x} = \langle x_1, \ldots, x_{\mathrm{ar}(X)} \rangle$ for the elements of X, together with other free individual variables $\vec{z}$; let $\mathrm{fv}_\phi(X)$ be the length of $\vec{z}$. Now if $\vec{t}$ is a sequence of $\mathrm{ar}(X)$ terms with free variables $\vec{y}$, then $(\mu X(\vec{x}).\phi)(\vec{t})[\vec{z}, \vec{y}]$ is a formula; **provided that** ϕ is IF-FOL$^+$. In this context, we write just $\mathrm{fv}(X)$ for $\mathrm{fv}_\phi(X)$.
- similarly for $\nu X(\vec{x}).\phi$.

To give the semantics of IF-LFP, we first define valuations for free fixpoint variables, in the context of some IF-LFP formula.

Definition 18 A fixpoint valuation $\mathscr{V}$ maps each fixpoint variable X to a set $\mathscr{V}(X) \in \wp(\wp(A^{\mathrm{ar}(X)+\mathrm{fv}(X)}))$.

Let D be a non-empty set of deals for $X(\vec{t})[\vec{x}, \vec{z}, \vec{y}]$, where $\vec{y}$ are the free variables of $\vec{t}$ not already among $\vec{x}, \vec{z}$. A deal $d = \vec{a}\vec{c}\vec{b} \in D$, where $\vec{a}, \vec{c}, \vec{b}$ are the deals for $\vec{x}, \vec{z}, \vec{y}$ respectively, determines a deal $d' = \vec{t}(d)\vec{c}$ for $X[\vec{x}, \vec{z}]$. Let $D' = \{\, d' \mid d \in D \,\}$. D is a trump for $X(\vec{t})$ iff $D' \in \mathscr{V}(X)$.

The intuition here is that a fixpoint variable needs to carry the trumps both for the elements of the fixpoint and for any free variables, as we shall see below. Then we define a suitable complete partial order on the range of valuations, which will also be the range of denotations for formulae; it is simply the inclusion order on trump sets.

Definition 19 If T_1 and T_2 are elements of $\wp(\wp(A^n))$, define $T_1 \preceq T_2$ iff $T_1 \subseteq T_2$.

This order gives the standard basic lemma for fixpoint logics:

Lemma 20 If $\phi(X)[\vec{x}, \vec{z}]$ is an IF-FOL$^+$ formula and $\mathscr{V}$ is a fixpoint valuation, the map on $\wp(\wp(A^{\mathrm{ar}(X)+\mathrm{fv}(X)})$ given by

$$T \mapsto [\![\phi]\!]_{\mathscr{V}[X:=T]}$$

is monotone with respect to $\preceq$; hence it has least and greatest fixpoints, constructible by iteration from the bottom and top elements of the set of denotations.

Thus we have the familiar definition of the μ operator:

Definition 21 $[\![\mu X(x).\phi(X)[\vec{x}, \vec{z}]]\!]$ is the least fixpoint of the map on $\wp(\wp(A^{\mathrm{ar}(X)+\mathrm{fv}(X)})$ given by

$$T \mapsto [\![\phi]\!]_{\mathscr{V}[X:=T]};$$

and $[\![\nu X(x).\phi(x)[\vec{x}, \vec{z}]]\!]$ is the greatest fixpoint. $\mu^\zeta X(x).\phi$ means the ζth approximant of $\mu X(x).\phi$, defined recursively by $\mu^\zeta X(x).\phi = \phi(\bigcup_{\xi<\zeta} \mu^\xi X(x).\phi)$.

A distinctive feature of the definition, compared to the normal LFP definition, is the way that free variables are explicitly mentioned. Normally, one can fix values for the free variables, and then compute the fixpoint, but because of independent quantification this is not possible in the IF setting. For example, consider the formula fragment

$$\forall z. \ldots \mu X(x). \ldots \vee \exists y/\{z\}. X(y)$$

The independent choice of y means that the trumps for the fixpoint depend on the possible deals for z, not just a single deal.

5 Independence-friendly modal mu-calculus

Now that we have the machinery for using fixpoints in an IF context, we can finally produce a truly independence-friendly temporal logic, the idea for which was sketched in [Bra04], relying on the IF modal logic we introduced in [BrF02].

5.1 IF modal logic

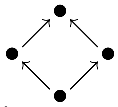
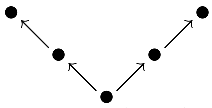
In designing an IF modal logic (in the style of Hintikka and Sandu rather than of Henkin), there are a number of issues to consider. The fullest exploration of these is in Tero Tulenheimo's thesis [Tul04], but here we use the syntax introduced in [BrF02]. As usual, we are avoiding the problem of negation, and sticking to a positive form.

Definition 22 The core 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*. $\mathtt{tt}$ and $\mathtt{ff}$ 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 $[a]_{\alpha/\beta_1,\ldots,\beta_m}\Phi_1$.

We shall impose the following requirement on all IFML formulae:

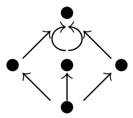
Definition 23 An IFML formula Φ is *well-formed* if
(a) in every subformula $\langle a\rangle_{\alpha/\beta_1,\ldots,\beta_m}\Psi$, the *bound tag* α is uniquely bound in Φ;
(b) every *independent tag* β_i *of* α is bound in some higher modality in Φ.

This is mainly a convenience to avoid issues of alpha-conversion. An alternative would be to follow the approach used by Tulenheimo in [Tul04], and label each modality with a de Bruijn index; however, the tagging approach is perhaps easier to read and write.

Naturally, the intention of this syntax is that the choice made at a modality should be independent of the choices made at the tagged independent modalities. Following the standard IF pattern, this means that in a uniform strategy Eloise's choice at $\langle a\rangle_{\alpha/\beta}$ should be uniform in the choice at β. One question then is, when is it *possible* for this to happen, given that the choices available at α themselves depend on the choice at β. In [BrF02], we saw this as a problem, and therefore we used enriched models with a built-in model of concurrency, such that the set of choices available at α would in fact not depend on the choice made at β. Moreover, the significance or otherwise of this question is sensitive to what exactly is being chosen: a successor state, or a transition (i.e. a source state, label, successor state tuple). In [Tul04], Tulenheimo observes that there is no need to depart from standard modal models: for example, in the model  the apex of the diamond can be uniformly chosen as a second step regardless of whether the first step went left or right, and this property, which in our notation is $[]_\beta\langle\rangle_{\alpha/\beta}\mathtt{tt}$, distinguishes it from the bisimilar model .

This works because the objects being chosen are, as in the usual semantics of basic modal logic, simply successor states, and so the uniformity requirement allows one to express that two potential successor states are actually the same.

On the other hand, there is another question one might ask: modal logics, including modal mu-calculus, are especially tractable precisely because they are local and forward-looking, which in game-theoretic terms manifests itself as the fact that if Eloise has a winning strategy in the model-checking game, she has a history-free winning strategy: that is, one that depends only on the current state and subformula, and is therefore uniform in all previous states. It thus seems a little paradoxical that in IFML as interpreted above, which is intended to allow us to add uniformity requirements to ML, we have formulae where Eloise has a winning strategy but no uniform winning strategy. The reason is that IFML allows us to impose uniformity requirements on the current state as well as the history – a rather unintuitive concept. From this, one can see that adding independence requirements to ML on Kripke structures essentially only adds the ability to express the idea of a common successor, as in the example above: any other statements about independence or dependence disappear in the general modal logic wash.

If we pass to a concurrent semantics as in [BrF02], things look a little different, and perhaps more intuitive. Now the objects being chosen are not just successor states, but transitions available from the current state, or more precisely *events*, which can be thought of as transitions local to a particular component of a system, and which may therefore be possible from many global states. In this setting, the formula $[]_\beta\langle\rangle_{\alpha/\beta}\mathtt{tt}$ is not true of the diamond model, since the two upper transitions are, in a normal transition system, distinct. In some enriched versions of transition systems, they might be identified as the same event, although in many cases this would create additional transitions in the models. For example, in the transition systems with concurrency used in [BrF02], if the upper transitions are the same event, this event must be concurrent with each of the lower transitions, which in turn means that the event must also be able to occur initially, resulting in the model actually being  which is indeed only possible if the bottom transitions carry different labels. In other models, such as Petri nets, these extra transitions would not be required.

Again, if, as we proposed in [BrF02], the event chosen at β is concurrent with the event chosen at α/β, then the paradox of $[]_\beta\langle\rangle_{\alpha/\beta}$ becomes less paradoxical: rather than seeing $/\beta$ as imposing a uniformity requirement on 'the current state', we can see it as imposing a uniformity requirement on *a non-local component* of the current state, while retaining the ability to depend on what we know locally about the current state.

There are, then, still some intuitive arguments for studying IFML on concurrent models. However, as can be guessed from the remarks above, the models used in [BrF02] are complex, and the semantics was defined via histories of transitions, together with conditions on histories enforcing that the strategies really choose events rather than transitions. Although the semantic model was convenient for the purpose of [BrF02], namely studying process equivalences, it is perhaps less intuitive than other models. Therefore we shall here present IFML on one of the most fundamental models of concurrency, and expand on the intuitive discussion.

Definition 24 A *labelled coherent event structure* E is set Ev of events, a labelling function $l : Ev \to \mathscr{L}$, a partial order causal dependency relation $\leq$ between events such that $\{ e' \mid e' \leq e \}$ is finite, and a symmetric irreflexive conflict relation $\#$ between events such that $e \leq e'$ and $e\#f$ implies $e'\#f$. Two events are *consistent* if they are not in conflict; they are *concurrent* if they are consistent and not ordered by $\leq$. A *configuration* is a set of pairwise consistent events, downward closed under $\leq$. A configuration C' is an a-successor (for $a \in \mathscr{L}$) of a configuration C, $C \xrightarrow{a} C'$, if $C' = C \cup \{e\}$ for some e such that $l(e) = a$. The size of an event structure is $|Ev| + |\leq| + |\#|$. Henceforth 'event structure' means 'labelled coherent event structure'.

Event structures provide a very basic, unstructured, presentation of concurrent systems. A configuration is a global state of the system, and so, for example, the event structure with n concurrent events has 2^n configurations, but configurations are derived entities rather than primitive entities.

Event structures can be generated from many other formalisms. For example, our n-component systems can be viewed as event structures where a synchronization event depends causally on all the preceding events in the synchronizing components. In particular, the reduction from C-SAT can be cast in terms of event structures, thereby showing that IFML on concurrent models also has the power of Henkin quantification – and again the number of configurations is quadratic, not exponential, in the number of literals, so the expressive power is genuine rather than a hidden state explosion.

At first sight, IFML is quite straightforwardly defined by games on event structures:

Definition 25 The game for a formula ϕ on an event structure E starts at the empty configuration and ϕ. At a position $(C, \langle a\rangle_{\alpha/\beta}\psi)$, Eloise chooses an a-labelled event e consistent with C, and play moves to $(C \cup \{e\}, \psi)$; dually for $[]$, and as usual for the booleans. $E \vDash \phi$ if Eloise has a winning strategy where the choice of events at a tag α/β is uniform in the events chosen at tag β.

Let us now consider the requirement in [BrF02], discussed above, that (translating into event structure terms) the event e_α chosen at α/β must be concurrent with the event e_β chosen at β. This seemed a natural way to tie together logical and concurrent independence. However, the necessity of this requirement was not shown, and Tulenheimo's interpretation of IFML has no need of concurrency. We argued above that the requirement has a natural intuition; consider now the consequences of not imposing it. It is clear that if e_α and e_β are not concurrent, then there are strong constraints on the system and the possible winning strategies, for the following reason. Suppose that Eloise's strategy at α chooses some $e_\alpha \geq e_\beta$. If Abelard chose instead e'_β, then for e_α to be a possible move, the event e_β must still occur in the play before α, as configurations are closed under causal dependency. This gives rise to a complex web of conditions on the dependencies between $e_\beta, e'_\beta, e_\alpha$ and their availability as moves at points in the formula between β and α; whereas if e_α and e_β are concurrent, no such conditions apply.

Moreover, even if e_α and e_β are concurrent, Abelard changing his choice could still have knock-on effects. Consider $[]_\gamma[]_\beta\langle\rangle_{\alpha/\gamma}$. Suppose Abelard chooses e_γ at γ and $e_\beta \geq e_\gamma$ at β. If he instead chooses e'_γ at γ, then he cannot choose e_β, so must have a different choice e'_β – and therefore Eloise's choice at α cannot be $\geq e_\beta$. This also means that Eloise can infer some information about the choice at γ from the choice at β. This is a phenomenon similar to the well-known phenomenon of 'signalling' in IF logic, but mediated through knowledge of the constraints on Abelard rather than directly via Eloise's early choices. (The classic example of signalling is $\forall x.\, \exists y.\, \exists z/x.$, where Eloise can use y to remember x when she chooses z.) For this reason, in [BrF02] we also required that IFML formula should be *good*, meaning that the dependency relation between tags should be transitive. $[]_\gamma[]_\beta\langle\rangle_{\alpha/\gamma}$ is not good, since (in our convention) β depends on γ. The formula $[]_\gamma[]_\beta\langle\rangle_{\alpha/\{\gamma,\beta\}}$ is good, with the effect that e_α must be concurrent with and uniform in e_α and e_β. The formula $[]_\gamma[]_{\beta/\gamma}\langle\rangle_{\alpha/\gamma}$ is also good, and raises the question of whether Abelard should also be constrained to choose concurrent events. (Requiring Abelard to have a uniform winning *strategy* is not relevant for the truth of IF formulae, only for the falsity; but one might still impose a constraint on the *moves* of the game, requiring Abelard to choose an event at β concurrent with his choice at γ.) On the grounds that one might wish to talk about the falsity of formulae, meaning that Abelard has a uniform winning strategy, it seems appropriate to impose the same restrictions on moves of both players.

We therefore arrive at the following modified definition of the IFML game on concurrent structures:

Definition 26 The game for a formula ϕ on an event structure E starts at the empty configuration and ϕ. At a position $(C, \langle a\rangle_{\alpha/\beta}\psi)$, Eloise chooses an a-labelled event e consistent with C and concurrent with the events chosen at β, and play moves to $(C \cup \{e\}, \psi)$; dually for $[]$, and as usual for the booleans. $E \vDash \phi$ if Eloise has a winning strategy where the choice of events at a tag α/β is uniform in the events chosen at tag β.

5.2 *Trump semantics for IFML*

To provide a traditional fixpoint semantics for an IF modal mu-calculus, it is first necessary to provide a traditional, non-game-theoretic, semantics for IFML. We will first briefly consider Tulenheimo's setting, and then our concurrent setting.

If IFML is interpreted simply on Kripke structures, defining a trump semantics is also simple, following the obvious IF-FOL translation of IFML. Each modality, indexed by α, has an associated first-order variable x_α recording the target state, the last of which is also the current state; the deals for a formula assign states to the variables, and the semantic clauses for the modal operator follow exactly those for IF-FOL. For example, a set D of deals is a trump for $\langle\rangle_{\alpha/W}\phi$ iff there is a trump E for ϕ such that for every $\simeq_W$-set $F \subseteq D$ there is a b such that $\{\, \vec{a}b \mid \vec{a} \in F \,\} \subseteq E$, where $\simeq_W$ is as in IF-FOL.

However, as shown in [Tul04] and already mentioned above, this is more than is actually needed. Because of the locality and forward-looking properties of modal logic, as soon as a dependent choice is made, all previous choices can be forgotten. In essence, the only non-ML formulae of IFML are those of the form $[]_\alpha\langle\rangle_{\beta/\alpha}\phi$, that assert the existence of a common successor state to Abelard's choice in the $[]$. By adding more independent modalities, variations on this theme can be expressed, but nothing really different.

Now consider the case of IFML on event structures. The straightforward approach is again to use an IF-FOL meta-language, with one variable x_α recording the event chosen at a modality, so that the deals for a formula are tuples of all the previously chosen events. The requirement for concurrency of independent choices is directly expressed in the meta-language translation: $\langle\rangle_{\alpha/\beta}\ldots$ translates to $\exists x_\alpha.\, x_\alpha \mathrel{\mathrm{co}} x_\beta \wedge \ldots$. The trumps are then defined by the IF-FOL semantics.

In this approach, a deal for a sub-formula contains the entire history of the game so far, and therefore contains implicitly the state (configuration) of the system. Knowing the state of the system is necessary to know the possible next events, but knowing the exact history seems overkill – and we would like to recover as much as possible of the local, history-free nature of usual modal logics. First, we note that a configuration of an event structure is uniquely determined by its set of $\geq$-maximal events, so as far as keeping state is concerned, we could throw away any events on which later events causally depend. As far as the logical dependencies are concerned, we would like to have a result similar to that for IFML on Kripke structures, that a dependent Eloise choice discards all previous information about independencies. However, this is not entirely true, because a dependent choice only determines part of the state; it remains possible to make independent choices about other parts of the state. In general, all we can say is that knowledge of a variable x_i can be discarded when its value is a non-maximal event and x_i is not mentioned in the remainder of the formula. If x_i is not mentioned in the remainder of the formula, but has a maximal event as its value, then that value has to be kept for state information.

5.3 *Trump semantics for IF-μ*

The keeping of information becomes problematic when we move to consider the fixpoint extension of IFML. The definition we adopted of IF-LFP has the effect that variables go out of scope as we pass through fixpoint variables, and from then on all Eloise choices are necessarily independent of the out-of-scope variables – but not necessarily uniform, since in-scope variables may have been determined by out-of-scope variables – an extension of the signalling phenomenon already seen in IF-FOL. Hence we must explicitly add a variable to the translation to maintain the full state information. To do this, we must keep a finite but potentially unbounded set of events in one variable; or, in the case that the event structure has bounded concurrency (i.e. there is a bound on the number of maximal events in a configuration), we could maintain a fixed number of auxiliary variables each recording one maximal event.

Taking the first approach, we translate a modality $\langle\rangle_{\alpha/\beta}\ldots$ by binding two variables, one of which records the event chosen, and one which records the updated state. If x, s are the variables holding the last chosen event and state, the translation is $\exists x_\alpha, s_\alpha/\beta.\, s \xrightarrow{x_\alpha} s_\alpha \wedge x_\alpha \text{ co } x_\beta \wedge \ldots$, where β here means all variables subscripted with β.

The box modality is translated similarly, and then fixpoints are translated directly as fixpoints. The result is a trump semantics for IF-μ. Whether such a semantics is usable in practice is perhaps open to argument.

6 Logics of dynamic parallelism

In the first part of this article, we discussed logics based on the Henkin quantifier. Although much simpler to deal with that IF style logics, they are perhaps unduly restricted by the use of n-component systems and the global synchronization requirements. On the other hand, the IF style logics are event based, although distributed programmers think in terms of parallel local processing rather than concurrent events. They are also hard to understand, even without fixpoints. One can imagine a modal logic for distributed systems which compromises between these two points of view, and allows for distrubuted statements when required, while also allowing global statements at other times. In this final section, we present a sketch for such a logic.

We have already used the notation $\otimes$ to describe the concurrent, independent combination of sequences of modalities. A natural extension is to take it as an operator on formulae instead, so one can write, for example, $([a](P_1 \wedge (\langle b\rangle * \wedge \langle c\rangle *))) \otimes ([\overline{a}](P_2 \wedge \langle - \rangle *))Q$, with the intended interpretation that the first component satisfies the first factor (with P_1 probably, but not necessarily, being a *local* proposition as in [HNW97]), making its existential choices independently; the second satisfies the second; and in the process, the two components synchronize on the a; and when they rendezvous at $*$, the system satisfies Q.

The most natural way to define such operators formally is to use a game semantics in IF style. Namely, take the syntax of modal logic, in positive form, and add: an n-ary (for each n) operator $\otimes$ ("parallel composition"), an atomic formula $*$ ("end of parallel"), and a binary operator . (roughly, "sequential composition"). (In the framework, of the first part of this article, only one level of $\otimes$ nesting is allowed, and $\otimes$ and $*$ may only occur on the left side of a "sequential composition".)

Given a system, a formula defines a game thus: the normal modalities are played as usual. At a formula $(\phi_1 \otimes \phi_2).\psi$, both the game and the system split into two concurrent parts: every modality move in one half must match a modality move in the other half, with the global system advancing as dictated by the synchronization algebra, but the choices in one half are independent of those in the other. When play reaches $*$ in both halves, play continues at ψ; if play stops because an atomic formula (other than $*$) is reached in either component, Eloise wins if the formula is locally satisfied (in both halves, in the event that both halves reach an internal atomic formula).

Such a game can, following the approach of the second part of this article, be expressed in terms of IF-FOL, albeit with a rather complex translation.

References

[AHK97] R. Alur, T. Henzinger and O. Kupferman, Alternating-time temporal logic, in *Proc. 38th FOCS (1997)*, 100–109.

[And93] H. R. Andersen, *Verification of Temporal Properties of Concurrent Systems*, DAIMI PB – 445, Computer Science Dept, Aarhus University, 1993.

[Bar79] J. Barwise, On branching quantifiers in English, *J. Philos. Logic* **8** 47–80 (1979).

[BlG86] A. Blass and Y. Gurevich. Henkin quantifiers and complete problems. *Ann. Pure Appl. Logic* **32**:1, 1–16 (1986).

[Bra97] J. C. Bradfield, The modal mu-calculus alternation hierarchy is strict, *Theor. Comput. Sci.* **195** 133–153 (1997).

[Bra99a] J. C. Bradfield, Fixpoint alternation and the game quantifier, *Proc. CSL 1999*, LNCS **1683** 350–361 (1999).

[Bra99b] J. C. Bradfield, Fixpoint alternation: arithmetic, transition systems, and the binary tree, *Theoret. Informatics Appl.* **33** 341–356 (1999).

[Bra00] J. C. Bradfield, Independence: logics and concurrency, *Proc. CSL 2000*, LNCS **1862** 247–261 (2000).

[Bra04] J. C. Bradfield, On independence-friendly fixpoint logics, *Philosophia Scientiae* **8**(2) 125–144 (2004).

[BrK05] J. C. Bradfield and S. Kreutzer, The complexity of independence-friendly fixpoint logic, *Proc. CSL 2005*, LNCS **3634** 355-368 (2005).

[BrF02] J. C. Bradfield and S. B. Fröschle, Independence-friendly modal logic and true concurrency, *Nordic J. Computing* **9** 102–117 (2002).

[Cen76] D. Cenzer, Monotone inductive definitions over the continuum, *J. Symbolic Logic* **41**:1 188–198 (1976).

[End70] H. B. Enderton, Finite partially ordered quantifiers, *Z. für Math. Logik u. Grundl. Math.* **16** 393–397 (1970).

[Got97] G. Gottlob, Relativized logspace and generalized quantifiers over finite ordered structures. *J. Symbolic Logic* **62**:2, 545–574 (1997).

[Hen61] L. Henkin, Some remarks on infinitely long formulas, *Infinitistic Methods*, Pergamon Press, Oxford and PAN, Warsaw, 167–183 (1961).

[HiS96] J. Hintikka and G. Sandu, A revolution in logic?, *Nordic J. Philos. Logic* **1**(2) 169–183 (1996).

[Hod97] W. Hodges, Compositional semantics for a language of imperfect information, *Int. J. IGPL* **5**(4), 539–563.

[HNW97] M. Huhn, P. Niebert and F. Wallner, Verification on local states, *Proc. TACAS '98*, LNCS **1384** 36–51.

[Lub93] R. S. Lubarsky, μ-definable sets of integers, *J. Symbolic Logic* **58** (1993) 291–313.

[KrM95] M. Krynicki and M. Mostowski, Henkin Quantifiers, in: M. Krynicki et al. (eds.), *Quantifiers: Logics, Models and Computation, Volume I*, 193–262 (1995).

[McM92] K. L. McMillan, Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits, *Proc. CAV'92*, 161–171 (1992).

[NiC94] M. Nielsen and C. Clausen, Bisimulations, games and logic. BRICS report RS-94-6. BRICS, Aarhus (1994).

[Pen92] W. Penczek, On undecidability of propositional temporal logics on trace systems, *Inf. Proc. Let.* **43**(3) 147–153 (1992).

[PK95] W. Penczek and R. Kuiper, Traces and Logic, in: Diekert and Rozenberg, ed. *The Book of Traces*, World Scientific (1995).

[San93] G. Sandu, On the logic of information independence and its applications, *J. Philos. Logic* **22** 361-372 (1993).

[StW90] C. P. Stirling and D. J. Walker, A general tableau technique for verifying temporal properties of concurrent programs. *Proc. Int. BCS–FACS Workshop on Semantics for Concurrency* Workshops in Computing (Springer–Verlag, Berlin, 1990) (1–15).

[Tul04] T. Tulenheimo, *Independence-Friendly Modal Logic: Studies in its Expressive Power and Theoretical Relevance*, Philosophical Studies from the University of Helsinki, vol. 4 (2004).

[Val91] A. Valmari, A stubborn attack on state explosion. *Proc. CAV '90*, DIMACS vol. 3, 25–42 (1991).

[Wal70] W. J. Walkoe, Jr, Finite partially-ordered quantification. *J. Symbolic Logic* **35** 535–555 (1970).

Version Control: hml.tex:1.24.