A Hybrid Linear Logic for Constrained Transition Systems

Joelle Despeyroux, Kaustuv Chaudhuri

A Hybrid Linear Logic for Constrained Transition Systems

Joëlle Despeyroux[1] and Kaustuv Chaudhuri[2]

1 **INRIA and CNRS, I3S, Sophia-Antipolis, France**
`joelle.despeyroux@inria.fr`
2 **INRIA, France**
`kaustuv.chaudhuri@inria.fr`

Abstract

Linear implication can represent state transitions, but real transition systems operate under temporal, stochastic or probabilistic constraints that are not directly representable in ordinary linear logic. We propose a general modal extension of intuitionistic linear logic where logical truth is indexed by constraints and hybrid connectives combine constraint reasoning with logical reasoning.The logic has a focused cut-free sequent calculus that can be used to internalize the rules of particular constrained transition systems; we illustrate this with an adequate encoding of the synchronous stochastic pi-calculus.

1998 ACM Subject Classification F.4.1. Mathematical Logic: Modal and Temporal Logics; F.1.2. Modes of Computation: Parallelism and Concurrency

Keywords and phrases linear logic, hybrid logic, stochastic pi-calculus, focusing, adequacy

Digital Object Identifier 10.4230/LIPIcs.TYPES.2013.XXX

1 Introduction

To reason about state transition systems, we need a logic of state. Linear logic [21] is such a logic and has been successfully used to model such diverse systems as process calculi [25], references and concurrency in programming languages [38], and formal security [7, 8], to give a few examples. Linear logic achieves this versatility by representing propositions as *resources* that are combined using $\otimes$, which can then be transformed using the linear implication ($\multimap$). However, linear implication is timeless: there is no way to correlate two concurrent transitions. If resources have lifetimes and state changes have temporal, probabilistic or stochastic *constraints*, then the logic will allow inferences that may not be realizable in the system being modelled. The need for formal reasoning in such constrained systems has led to the creation of specialized formalisms such as Computation Tree Logic (*CTL*)[18], Continuous Stochastic Logic (*CSL*) [2] or Probabilistic CTL (*PCTL*) [22]. These approaches pay a considerable encoding overhead for the states and transitions in exchange for the constraint reasoning not provided by linear logic. A prominent alternative to the logical approach is to use a suitably enriched process algebra such as the stochastic and probabilistic π-calculi or the κ-calculus [14]. Processes are animated by means of simulation and then compared with the observations. Process calculi do not however completely fill the need for *formal reasoning for constrained transition systems.*

We propose a simple yet general method to add constraint reasoning to linear logic. It is an old idea—*labelled deduction* [37] with *hybrid* connectives [6]—applied to a new domain. Precisely, we parameterize ordinary logical truth on a *constraint domain*: $A@w$ stands for the truth of A under constraint w. Only a basic monoidal structure is assumed about

19th International Conference on Types for Proofs and Programs (TYPES 2013).
Editors: Ralph Matthes and Aleksy Schubert; pp. 151–169

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the constraints from a proof-theoretic standpoint. We then use the hybrid connectives of *satisfaction* (`at`) and *localization* ($\downarrow$) to perform generic symbolic reasoning on the constraints at the propositional level. We call the result *hybrid linear logic* (HyLL); it has a generic cut-free (but cut admitting) sequent calculus that can be strengthened with a focusing restriction [1] to obtain a normal form for proofs. Any instance of HyLL that gives a semantic interpretation to the constraints enjoys these proof-theoretic properties.

Focusing allows us to treat HyLL as a *logical framework* for constrained transition systems. Logical frameworks with hybrid connectives have been considered before; hybrid *LF* (*HLF*), for example, is a generic mechanism to add many different kinds of resource-awareness, including linearity, to ordinary *LF* [33]. *HLF* follows the usual *LF* methodology of keeping the logic of the framework minimal: its proof objects are β-normal η-long natural deduction terms, but the equational theory of such terms is sensitive to permutative equivalences [39]. With a focused sequent calculus, we have more direct access to a canonical representation of proofs, so we can enrich the framework with any connectives that obey the focusing discipline. The representational adequacy of an encoding in terms of (partial) focused sequent derivations tends to be more straightforward than in a natural deduction formulation. We illustrate this by encoding the synchronous stochastic π-calculus ($S\pi$) in HyLL using rate functions as constraints.

In addition to the novel stochastic component, our encoding of $S\pi$ is a conceptual improvement over other encodings of π-calculi in linear logic. In particular, we perform a full propositional reflection of processes as in [25], but our encoding is first-order and adequate as in [9]. HyLL does not itself prescribe an operational semantics for the encoding of processes; thus, bisimilarity in continuous time Markov chains (*CTMC*) is not the same as logical equivalence in stochastic HyLL, unlike in *CSL* [16]. This is not a deficiency; rather, the *combination* of focused HyLL proofs and a proof search strategy tailored to a particular encoding is necessary to produce faithful symbolic executions. This exactly mirrors $S\pi$ where it is the simulation rather than the transitions in the process calculus that is shown to be faithful to the *CTMC* semantics [29].

This work has the following main contributions. First is the logic HyLL itself and its associated proof-theory, which has a very standard and well understood design. Second, we show how to obtain many different instances of HyLL for particular constraint domains because we only assume a basic monoidal structure for constraints. Third, we illustrate the use of focused sequent derivations to obtain adequate encodings by giving a novel adequate encoding of $S\pi$. Our encoding is, in fact, *fully adequate*, *i.e.*, partial focused proofs are in bijection with traces. The ability to encode $S\pi$ gives an indication of the versatility of HyLL.

This paper is organized as follows: in sec. 2, we present the inference (sequent calculus) rules for HyLL and describe the two main semantic instances: temporal and probabilistic constraints. In sec. 3 we sketch the general focusing restriction on HyLL sequent proofs. In sec. 4 we give the encoding of $S\pi$ in probabilistic HyLL, and show that the encoding is representationally adequate for focused proofs (theorems 16 and 18). We end with an overview of related (sec. 5) and future work (sec. 6). The full version of this paper is available as a technical report [12].

2 Hybrid Linear Logic

In this section we define HyLL, a conservative extension of intuitionistic first-order linear logic (ILL) [21] where the truth judgements are labelled by worlds representing constraints. Like in ILL, propositions are interpreted as *resources* which may be composed into a *state*

using the usual linear connectives, and the linear implication ($\multimap$) denotes a transition between states. The world label of a judgement represents a constraint on states and state transitions; particular choices for the worlds produce particular instances of HyLL. The common component in all the instances of HyLL is the proof theory, which we fix once and for all. We impose the following minimal requirement on the kinds of constraints that HyLL can deal with.

▶ **Definition 1.** A *constraint domain* $\mathcal{W}$ is a monoid structure $\langle W, \cdot, \iota \rangle$. The elements of W are called *worlds*, and the partial order $\preceq : W \times W$—defined as $u \preceq w$ if there exists $v \in W$ such that $u \cdot v = w$—is the *reachability relation* in $\mathcal{W}$.

The identity world ι is $\preceq$-initial and is intended to represent the lack of any constraints. Thus, the ordinary ILL is embeddable into any instance of HyLL by setting all world labels to the identity. When needed to disambiguate, the instance of HyLL for the constraint domain $\mathcal{W}$ will be written HyLL($\mathcal{W}$). Two design choices are important to note. First, we only require the worlds to be monoids, not lattices, because this lets us give a sufficiently general system that it can be instantiated with rate functions as the constraint domain. Second, we do not assume that the monoid is commutative so that we can still choose to use lattices for the constraint domain.

Atomic propositions are written using lowercase letters ($a, b, \ldots$) applied to a sequence of *terms* ($s, t, \ldots$), which are drawn from an untyped term language containing term variables ($x, y, \ldots$) and function symbols ($f, g, \ldots$) applied to a list of terms. Non-atomic propositions are constructed from the connectives of first-order intuitionistic linear logic and the two hybrid connectives *satisfaction* (`at`), which states that a proposition is true at a given world ($w, u, v, \ldots$), and *localization* ($\downarrow$), which binds a name for the (current) world the proposition is true at. The following grammar summarizes the syntax of HyLL propositions.

$$\begin{aligned} A, B, \ldots ::= {} & a\ \vec{t} \mid A \otimes B \mid \mathbf{1} \mid A \multimap B \mid A \,\&\, B \mid \top \mid A \oplus B \mid \mathbf{0} \mid !\,A \mid \forall x.\, A \mid \exists x.\, A \\ \mid {} & (A\ \mathtt{at}\ w) \mid \downarrow u.\, A \mid \forall u.\, A \mid \exists u.\, A \end{aligned}$$

Note that in the propositions $\downarrow u.\, A$, $\forall u.\, A$ and $\exists u.\, A$, world u is bound in A. World variables cannot be used in terms, and neither can term variables occur in worlds; this restriction is important for the modular design of HyLL because it keeps purely logical truth separate from constraint truth. We let α range over variables of either kind. As we shall prove later (Theorem 5), the $\downarrow$ connective commutes with every propositional connective, including itself. That is, $\downarrow u.\, (A * B)$ is equivalent to $(\downarrow u.\, A) * (\downarrow u.\, B)$ for all binary connectives $*$, and $\downarrow u.\, * A$ is equivalent to $*(\downarrow u.\, A)$ for every unary connective $*$, assuming the commutation will not cause an unsound capture of u. It is purely a matter of taste where to place the $\downarrow$, and repetitions are harmless.

The unrestricted connectives $\wedge$, $\vee$, $\supset$, *etc.* of intuitionistic (non-linear) logic can also be defined in terms of the linear connectives and the exponential ! using any of the available embeddings of intuitionistic logic into linear logic, such as Girard's embedding [21]. For example, $A \supset B$ can be defined as $!\,A \multimap B$.

2.1 Sequent Calculus for HyLL

In this section, we give a sequent calculus presentation of HyLL and prove a cut-admissibility theorem. The sequent formulation in turn will lead to an analysis of the polarities of the connectives in order to get a focused sequent calculus that can be used to compile a logical theory into a system of derived inference rules with nice properties (sec. 3). For instance, if a given theory defines a transition system, then the derived rules of the focused calculus will

exactly exhibit the same transitions. This is key to obtain the necessary representational adequacy theorems, as we shall see for the $S\pi$-calculus example chosen in this paper (sec. 4.1).

We start with the judgements from linear logic [21] and enrich them with a modal situated truth. We present the syntax of hybrid linear logic in a sequent calculus style, using Martin-Löf's principle of separating judgements and logical connectives. Instead of the ordinary mathematical judgement "A is true", for a proposition A, judgements of HyLL are of the form "A is true at world w", abbreviated as $A@w$. We use sequents of the form $\Gamma\ ;\ \Delta \implies C@w$ where Γ and Δ are sets of judgements of the form $A@w$, with Δ being moreover a *multiset*. Γ is called the *unrestricted context*: its hypotheses can be consumed any number of times. Δ is a *linear context*: every hypothesis in it must be consumed singly in the proof. Note that in a judgement $A@w$ (as in a proposition $A\ \mathtt{at}\ w$), w can be any expression in $\mathcal{W}$, not only a variable. The notation $A[\tau/\alpha]$ stands for the replacement of all free occurrences of the variable α in A with the expression τ, avoiding capture. Note that the expressions in the rules are to be read up to alpha-conversion.

The full collection of rules of the HyLL sequent calculus is in fig. 1. The rules for the linear connectives are borrowed from [11] where they are discussed at length, so we omit a more thorough discussion here. The rules for the first-order quantifiers are completely standard. A brief discussion of the hybrid rules follows. To introduce the *satisfaction* proposition $(A\ \mathtt{at}\ u)$ (at any world v') on the right, the proposition A must be true in the world u. The proposition $(A\ \mathtt{at}\ u)$ itself is then true at any world, not just in the world u. In other words, $(A\ \mathtt{at}\ u)$ carries with it the world at which it is true. Therefore, suppose we know that $(A\ \mathtt{at}\ u)$ is true (at any world v); then, we also know that $A@u$. The other hybrid connective of *localisation*, $\downarrow$, is intended to be able to name the current world. That is, if $\downarrow u.\, A$ is true at world w, then the variable u stands for w in the body A. This interpretation is reflected in its introduction rule on the right $\downarrow R$. For left introduction, suppose we have a proof of $\downarrow u.\, A@v$ for some world v. Then, we also know $[v/u]A@v$.

There are only two structural rules: the init rule infers an atomic initial sequent, and the copy rule introduces a contracted copy of an unrestricted assumption into the linear context (reading from conclusion to premise). Weakening and contraction are admissible rules:

▶ **Theorem 2** (structural properties)**.**

1. *If* $\Gamma\ ;\ \Delta \Longrightarrow C@w$*, then* $\Gamma, \Gamma'\ ;\ \Delta \Longrightarrow C@w$*. (weakening)*
2. *If* $\Gamma, A@u, A@u\ ;\ \Delta \Longrightarrow C@w$*, then* $\Gamma, A@u\ ;\ \Delta \Longrightarrow C@w$*. (contraction)*

Proof. By straightforward structural induction on the given derivations. ◀

The most important structural properties are the admissibility of the identity and the cut principles. The identity theorem is the general case of the init rule and serves as a global syntactic completeness theorem for the logic. Dually, the cut theorem below establishes the syntactic soundness of the calculus; moreover there is no cut-free derivation of $\cdot\ ;\ \cdot \Longrightarrow \mathbf{0}@w$, so the logic is also globally consistent.

▶ **Theorem 3** (identity)**.** $\Gamma\ ;\ A@w \Longrightarrow A@w$.

Proof. By induction on the structure of A (see [12]). ◀

▶ **Theorem 4** (cut)**.**

1. *If* $\Gamma\ ;\ \Delta \Longrightarrow A@u$ *and* $\Gamma\ ;\ \Delta', A@u \Longrightarrow C@w$*, then* $\Gamma\ ;\ \Delta, \Delta' \Longrightarrow C@w$.
2. *If* $\Gamma\ ;\ \cdot \Longrightarrow A@u$ *and* $\Gamma, A@u\ ;\ \Delta \Longrightarrow C@w$*, then* $\Gamma\ ;\ \Delta \Longrightarrow C@w$.

Judgemental rules

$$\frac{}{\Gamma\ ;\ a\ \vec{t}\ @u \Longrightarrow a\ \vec{t}\ @u}\ \text{init} \qquad \frac{\Gamma, A@u\ ;\ \Delta, A@u \Longrightarrow C@w}{\Gamma, A@u\ ;\ \Delta \Longrightarrow C@w}\ \text{copy}$$

Multiplicatives

$$\frac{\Gamma\ ;\ \Delta \Longrightarrow A@w \quad \Gamma\ ;\ \Delta' \Longrightarrow B@w}{\Gamma\ ;\ \Delta, \Delta' \Longrightarrow A \otimes B@w}\ \otimes R \qquad \frac{\Gamma\ ;\ \Delta, A@u, B@u \Longrightarrow C@w}{\Gamma\ ;\ \Delta, A \otimes B@u \Longrightarrow C@w}\ \otimes L$$

$$\frac{}{\Gamma\ ;\ \cdot \Longrightarrow \mathbf{1}@w}\ \mathbf{1}R \qquad \frac{\Gamma\ ;\ \Delta \Longrightarrow C@w}{\Gamma\ ;\ \Delta, \mathbf{1}@u \Longrightarrow C@w}\ \mathbf{1}L$$

$$\frac{\Gamma\ ;\ \Delta, A@w \Longrightarrow B@w}{\Gamma\ ;\ \Delta \Longrightarrow A \multimap B@w}\ \multimap R \qquad \frac{\Gamma\ ;\ \Delta \Longrightarrow A@u \quad \Gamma\ ;\ \Delta', B@u \Longrightarrow C@w}{\Gamma\ ;\ \Delta, \Delta', A \multimap B@u \Longrightarrow C@w}\ \multimap L$$

Additives

$$\frac{}{\Gamma\ ;\ \Delta \Longrightarrow \top@w}\ \top R \qquad \frac{}{\Gamma\ ;\ \Delta, \mathbf{0}@u \Longrightarrow C@w}\ \mathbf{0}L$$

$$\frac{\Gamma\ ;\ \Delta \Longrightarrow A@w \quad \Gamma\ ;\ \Delta \Longrightarrow B@w}{\Gamma\ ;\ \Delta \Longrightarrow A\ \&\ B@w}\ \&R \qquad \frac{\Gamma\ ;\ \Delta, A_i@u \Longrightarrow C@w}{\Gamma\ ;\ \Delta, A_1\ \&\ A_2@u \Longrightarrow C@w}\ \&L_i$$

$$\frac{\Gamma\ ;\ \Delta \Longrightarrow A_i@w}{\Gamma\ ;\ \Delta \Longrightarrow A_1 \oplus A_2@w}\ \oplus R_i \qquad \frac{\Gamma\ ;\ \Delta, A@u \Longrightarrow C@w \quad \Gamma\ ;\ \Delta, B@u \Longrightarrow C@w}{\Gamma\ ;\ \Delta, A \oplus B@u \Longrightarrow C@w}\ \oplus L$$

Quantifiers

$$\frac{\Gamma\ ;\ \Delta \Longrightarrow A@w}{\Gamma\ ;\ \Delta \Longrightarrow \forall\alpha.\ A@w}\ \forall R^{\alpha} \qquad \frac{\Gamma\ ;\ \Delta, [\tau/\alpha]A@u \Longrightarrow C@w}{\Gamma\ ;\ \Delta, \forall\alpha.\ A@u \Longrightarrow C@w}\ \forall L$$

$$\frac{\Gamma\ ;\ \Delta \Longrightarrow [\tau/\alpha]A@w}{\Gamma\ ;\ \Delta \Longrightarrow \exists\alpha.\ A@w}\ \exists R \qquad \frac{\Gamma\ ;\ \Delta, A@u \Longrightarrow C@w}{\Gamma\ ;\ \Delta, \exists\alpha.\ A@u \Longrightarrow C@w}\ \exists L^{\alpha}$$

For $\forall R^{\alpha}$ and $\exists L^{\alpha}$, α is assumed to be fresh with respect to the conclusion. For $\exists R$ and $\forall L$, τ stands for a term or world, as appropriate.

Exponentials

$$\frac{\Gamma\ ;\ \cdot \Longrightarrow A@w}{\Gamma\ ;\ \cdot \Longrightarrow\ !A@w}\ !R \qquad \frac{\Gamma, A@u\ ;\ \Delta \Longrightarrow C@w}{\Gamma\ ;\ \Delta, !A@u \Longrightarrow C@w}\ !L$$

Hybrid connectives

$$\frac{\Gamma\ ;\ \Delta \Longrightarrow A@u}{\Gamma\ ;\ \Delta \Longrightarrow (A\ \mathtt{at}\ u)@v}\ \mathtt{at}R \qquad \frac{\Gamma\ ;\ \Delta, A@u \Longrightarrow C@w}{\Gamma\ ;\ \Delta, (A\ \mathtt{at}\ u)@v \Longrightarrow C@w}\ \mathtt{at}L$$

$$\frac{\Gamma\ ;\ \Delta \Longrightarrow [w/u]A@w}{\Gamma\ ;\ \Delta \Longrightarrow \downarrow u.\ A@w}\ \downarrow R \qquad \frac{\Gamma\ ;\ \Delta, [v/u]A@v \Longrightarrow C@w}{\Gamma\ ;\ \Delta, \downarrow u.\ A@v \Longrightarrow C@w}\ \downarrow L$$

Figure 1 The sequent calculus for HyLL

.

Proof. By lexicographic structural induction on the given derivations, with cuts of kind 2 additionally allowed to justify cuts of kind 1. The style of proof sometimes goes by the name of *structural cut-elimination* [11]. See [12] for the details. ◀

We can use the admissible cut rules to show that the following rules are invertible: $\otimes L$, $\mathbf{1}L$, $\oplus L$, $\mathbf{0}L$, $\exists L$, $\multimap R$, $\& R$, $\top R$, and $\forall R$. In addition, the four hybrid rules, $\mathtt{at}R$, $\mathtt{at}L$, $\downarrow R$ and $\downarrow L$ are invertible. In fact, $\downarrow$ and $\mathtt{at}$ commute freely with all non-hybrid connectives:

▶ **Theorem 5** (Invertibility). *The following rules are invertible:*

1. *On the right:* $\& R$, $\top R$, $\multimap R$, $\forall R$, $\downarrow R$ *and* $\mathtt{at}R$;
2. *On the left:* $\otimes L$, $\mathbf{1}L$, $\oplus L$, $\mathbf{0}L$, $\exists L$, $!L$, $\downarrow L$ *and* $\mathtt{at}L$. ◀

▶ **Corollary 6** (consistency). *There is no proof of* $\cdot\ ;\ \cdot \Longrightarrow \mathbf{0}\,@w$.

Proof. A straightforward consequence of thm. 4. ◀

HyLL is conservative with respect to ordinary intuitionistic linear logic: as long as no hybrid connectives are used, the proofs in HyLL are identical to those in ILL [11]. The proof (omitted) is by simple structural induction.

▶ **Theorem 7** (conservativity). *Call a proposition or multiset of propositions* pure *if it contains no instance of the hybrid connectives and no instance of quantification over a world variable, and let* Γ, Δ *and* A *be pure. Then, If* $\Gamma\ ;\ \Delta \Longrightarrow_{HyLL} C\,@w$ *is derivable, then so is* $\Gamma\ ;\ \Delta \Longrightarrow_{ILL} C$. ◀

In the rest of this paper we use the following derived connectives:

▶ **Definition 8** (modal connectives).

$$\Box A \triangleq \downarrow u.\, \forall w.\, (A\ \mathtt{at}\ u \cdot w) \qquad \Diamond A \triangleq \downarrow u.\, \exists w.\, (A\ \mathtt{at}\ u \cdot w)$$

$$\rho_v\, A \triangleq \downarrow u.\, (A\ \mathtt{at}\ u \cdot v) \qquad \dagger A \triangleq \forall u.\, (A\ \mathtt{at}\ u)$$

The connective ρ represents a form of delay. Note its derived right rule:

$$\frac{\Gamma\ ;\ \Delta \vdash A@w \cdot v}{\Gamma\ ;\ \Delta \vdash \rho_v\, A@w}\ \rho R$$

The proposition $\rho_v\, A$ thus stands for an *intermediate state* in a transition to A. Informally it can be thought to be "v before A"; thus, $\Box A = \forall v.\ \rho_v\, A$ represents *all* intermediate states in the path to A, and $\Diamond A = \exists v.\ \rho_v\, A$ represents *some* such state. The modally unrestricted proposition $\dagger A$ represents a resource that is consumable in any world; it is intended to be used to make the transition rules available everywhere.

It is worth remarking that the reachability relation in HyLL is trivial: every world that can be defined is reachable from every other. To illustrate, the (linear form of the) axioms of the S5 modal logic are derivable in HyLL; in particular, the sequent $\cdot\ ;\ \Diamond A@w \Longrightarrow \Box \Diamond A@w$, which represents the 5 axiom, is provable. HyLL is, in fact, more expressive than S5 as it allows direct manipulation of the worlds using the hybrid connectives: for example, the ρ connective is not definable in S5.

2.2 Temporal Constraints

As a pedagogical example, consider the constraint domain $\mathcal{T} = \langle \mathbf{R}^+, +, 0 \rangle$ representing instants of time. This domain can be used to define the lifetime of resources, such as keys, sessions, or delegations of authority. Delay (defn. 8) in HyLL($\mathcal{T}$) represents intervals of time;

$\rho_d\, A$ means "A will become available after delay d", similar to metric tense logic [32]. This domain is very permissive because addition is commutative, resulting in the equivalence of $\rho_u\, \rho_v\, A$ and $\rho_v\, \rho_u\, A$. The "forward-looking" connectives G (always in the future) and F (sometimes in the future) of ordinary tense logic are precisely $\Box$ and $\Diamond$ of defn. 8.

In addition to the future connectives, the domain $\mathcal{T}$ also admits past connectives if we add saturating subtraction (*i.e.*, $a - b = 0$ if $b \geq a$) to the language of worlds. We can then define the duals H (historically) and O (once) of G and F as:

$$\mathrm{H}\ \mathrm{A} \triangleq \downarrow u.\, \forall w.\, (A\ \mathtt{at}\ u - w) \qquad \mathrm{O}\ \mathrm{A} \triangleq \downarrow u.\, \exists w.\, (A\ \mathtt{at}\ u - w)$$

While this domain does not have any branching structure like CTL, it is expressive enough for many common idioms because of the branching structure of derivations involving $\oplus$. CTL reachability ("in some path in some future"), for instance, is the same as our $\Diamond$; similarly CTL steadiness ("in some path for all futures") is the same as $\Box$. CTL stability ("in all paths in all futures"), however, has no direct correspondance in HyLL (see however [15] for a correspondance in particular cases). Note that model checking cannot cope with temporal expresssions involving the "in all paths" notion anyway [1].

On the other hand, the availability of linear reasoning, enriched with modalities, makes certain kinds of reasoning in HyLL much more natural than in ordinary temporal logics. One important example is of *oscillation* between states in systems with kinetic feedback. In a temporal specification language such as BIOCHAM [10], only aperiodic oscillations are representable, while in HyLL an oscillation between A and B with delay d is represented by the rule $\dagger(A \multimap \rho_d\, B)\ \&\ (B \multimap \rho_d\, A)$ (or $\dagger(A \multimap \Diamond\, B)\ \&\ (B \multimap \Diamond\, A)$ if the oscillation is aperiodic). If HyLL($\mathcal{T}$) were extended with constrained implication and conjunction in the style of CILL [36] or η [17], then we can define localized versions of $\Box$ and $\Diamond$, such as "A is true everywhere/somewhere in an interval".

For examples of applications of HyLL with temporal constraints, the interested reader can see [15], which gives an encoding of a simple biological system and its temporal properties in HyLL($\mathcal{T}'$), where $\mathcal{T}' = \langle \mathbb{N}, +, 0 \rangle$ represents discrete instants of time. We will, instead, use a version of HyLL dedicated to continuous time Markov Chains with exponential distribution, as used in Sπ. We introduce this type of constraints below.

2.3 Probabilistic Constraints

Transitions in practice rarely have precise delays. Phenomenological and experimental evidence is used to construct a probabilistic model of the transition system where the delays are specified as probability distributions of continuous (or discrete) variables. A number of variations of monoids representing probabilistic and stochastic constraints are presented in [12], both for the general case and for the special case of Markov processes.

One of the standard models of stochastic transition systems is continuous time Markov chains (CTMCs) where the delays of transitions between states are distributed according to the Markov assumption of memorylessness (Markov processes) with the further condition that their state-spaces are countable [35]. In the synchronous stochastic π-calculus (Sπ), the probability of a reaction with *rate* r is given by continuous time Markov chains with exponential distribution of parameter r (See [31]). To describe such processes, we shall take $\mathbf{R}^+$ to represent the *rates* of their exponential distribution. To encode the Sπ calculus in a suitable instanciation of HyLL, we only need a *symbolic* operation on the rates. This abstract

[1] at least in their full generality, involving an infinite number of states.

treatment can be made fully precise, but this would require a detour into measure theory that is beyond the scope of this paper; see [12] for the details.

▶ **Definition 9.** The *rates domain* $\mathcal{R}$ is the monoid $\mathcal{R} = \langle \mathbf{R}^{+*}, \cdot, [] \rangle$ of lists of positive reals, where $\cdot$ is concatenation of lists, and [] is the empty list.

Worlds $r \in \mathcal{R}$ represent the (rates of the) *sequence* of actions that have led to the current world from a given fixed initial world. We might equivalently have chosen $\mathcal{T} = \langle \mathbf{R}^{+}, +, 0 \rangle$ to represent average time delays, which would be the sum of the reciprocals of the rates in the list of rates. We choose to use lists of rates because they are more informative than average time delays. Note that since our rate functions are assumed to be memoryless, the order of the list of rates is immaterial, so we can easily relax it to a multi-set of rates; this change could not substantially alter the development of this paper.

3 Focusing

As HyLL is intended to represent transition systems adequately, it is crucial that HyLL derivations in the image of an encoding have corresponding transitions. However, transition systems are generally specified as rewrite algebras over an underlying congruence relation. These congruences have to be encoded propositionally in HyLL, so a HyLL derivation will generally require several inference rules to implement a single transition; moreover, several trivially different reorderings of these "micro" inferences would correspond to the same transition. It is therefore futile to attempt to define an operational semantics directly on HyLL inferences.

We restrict the syntax to focused derivations [1], which ignores many irrelevant rule permutations in a sequent proof and divides the proof into clear *phases* that define the grain of atomicity. The logical connectives are divided into two classes, *negative* and *positive*, and rule permutations for connectives of like polarity are confined to *phases*. A *focused derivation* is one in which the positive and negative rules are applied in alternate maximal phases in the following way: in the *active* phase, all negative rules are applied (in irrelevant order) until no further negative rule can apply; the phase then switches and one positive proposition is selected for *focus*; this focused proposition is decomposed under focus (*i.e.*, the focus persists to its sub-formulas) until it becomes negative, and the phase switches again.

As noted before, the logical rules of the hybrid connectives $\mathtt{at}$ and $\downarrow$ are invertible, so they can be considered to have both polarities. It would be valid to decide a polarity for each occurrence of each hybrid connective independently; however, as they are mainly intended for book-keeping during logical reasoning, we define the polarity of these connectives in the following *parasitic* form: if its immediate subformula is positive (resp. negative) connective, then it is itself positive (resp. negative). These connectives therefore become invisible to focusing. This choice of polarity can be seen as a particular instance of a general scheme that divides the $\downarrow$ and $\mathtt{at}$ connectives into two polarized forms each. To complete the picture, we also assign a polarity for the atomic propositions; this restricts the shape of focusing phases further [13]. The full syntax of positive ($P, Q, \ldots$) and negative ($M, N, \ldots$) propositions is as follows:

$$\begin{array}{rcl} P, Q, \ldots & ::= & p\ \vec{t} \mid P \otimes Q \mid \mathbf{1} \mid P \oplus Q \mid \mathbf{0} \mid !\,N \mid \exists \alpha.\ P \mid \downarrow u.\, P \mid (P\ \mathtt{at}\ w) \mid \Downarrow N \\ N, M, \ldots & ::= & n\ \vec{t} \mid N\ \&\ N \mid \top \mid P \multimap N \mid \forall \alpha.\ N \mid \downarrow u.\, N \mid (N\ \mathtt{at}\ w) \mid \Uparrow P \end{array}$$

The two syntactic classes refer to each other via the new *shift* connectives $\Uparrow$ and $\Downarrow$. Sequents in the focusing calculus are of the following forms.

Focused logical rules

$$\frac{}{\Gamma\ ;\ \left[n\ \vec{t}@w\right] \Longrightarrow \Downarrow n\ \vec{t}@w}\ \mathrm{li} \qquad \frac{\Gamma\ ;\ \Delta\ ;\ P@u \Longrightarrow \cdot\ ;\ Q@w}{\Gamma\ ;\ \Delta\ ;\ \left[\Uparrow P@u\right] \Longrightarrow Q@w}\ \Uparrow L \qquad \frac{\Gamma\ ;\ \Delta\ ;\ \cdot \Longrightarrow N@w\ ;\ \cdot}{\Gamma\ ;\ \Delta \Longrightarrow \left[\Downarrow N@w\right]}\ \Downarrow R$$

$$\frac{\Gamma\ ;\ \Delta\ ;\ \left[N_i@u\right] \Longrightarrow Q@w}{\Gamma\ ;\ \Delta\ ;\ \left[N_1\ \&\ N_2@u\right] \Longrightarrow Q@w}\ \&L_i \qquad \frac{\Gamma\ ;\ \Delta \Longrightarrow \left[P@u\right] \quad \Gamma\ ;\ \Xi\ ;\ \left[N@u\right] \Longrightarrow Q@w}{\Gamma\ ;\ \Delta, \Xi\ ;\ \left[P \multimap N@u\right] \Longrightarrow Q@w}\ \multimap L$$

$$\frac{\Gamma\ ;\ \Delta\ ;\ \left[[\tau/\alpha]N@u\right] \Longrightarrow Q@w}{\Gamma\ ;\ \Delta\ ;\ \left[\forall\alpha.\ N@u\right] \Longrightarrow Q@w}\ \forall L \qquad \frac{\Gamma\ ;\ \Delta\ ;\ \left[[v/u]N@v\right] \Longrightarrow Q@w}{\Gamma\ ;\ \Delta\ ;\ \left[\downarrow u.\ N@v\right] \Longrightarrow Q@w}\ \downarrow LF$$

$$\frac{\Gamma\ ;\ \Delta\ ;\ \left[N@u\right] \Longrightarrow Q@w}{\Gamma\ ;\ \Delta\ ;\ \left[(N\ \mathtt{at}\ u)@v\right] \Longrightarrow Q@w}\ \mathtt{at}LF \qquad \frac{}{\Gamma\ ;\ \Uparrow p\ \vec{t}@w \Longrightarrow \left[p\ \vec{t}@w\right]}\ \mathrm{ri}$$

$$\frac{\Gamma\ ;\ \Delta \Longrightarrow \left[P@w\right] \quad \Gamma\ ;\ \Xi \Longrightarrow \left[Q@w\right]}{\Gamma\ ;\ \Delta, \Xi \Longrightarrow \left[P \otimes Q@w\right]}\ \otimes R \qquad \frac{\Gamma\ ;\ \Delta \Longrightarrow \left[P_i@w\right]}{\Gamma\ ;\ \Delta \Longrightarrow \left[P_1 \oplus P_2@w\right]}\ \oplus R_i$$

$$\frac{\Gamma\ ;\ \Delta \Longrightarrow \left[[\tau/\alpha]P@w\right]}{\Gamma\ ;\ \Delta \Longrightarrow \left[\exists\alpha.\ P@w\right]}\ \exists R \qquad \frac{\Gamma\ ;\ \cdot\ ;\ \cdot \Longrightarrow N@w\ ;\ \cdot}{\Gamma\ ;\ \cdot \Longrightarrow \left[!N\right]@w}\ !R \qquad \frac{\Gamma\ ;\ \Delta \Longrightarrow \left[[w/u]P@w\right]}{\Gamma\ ;\ \Delta \Longrightarrow \left[\downarrow u.\ P@w\right]}\ \downarrow RF$$

$$\frac{\Gamma\ ;\ \Delta \Longrightarrow \left[P@u\right]}{\Gamma\ ;\ \Delta \Longrightarrow \left[(P\ \mathtt{at}\ u)@w\right]}\ \mathtt{at}RF \qquad \frac{}{\Gamma\ ;\ \cdot \Longrightarrow \left[\mathbf{1}@w\right]}\ \mathbf{1}R$$

Active logical rules

(R of the form $\cdot\ ;\ Q@w$ or $N@w\ ;\ \cdot$, and L of the form $\Gamma\ ;\ \Delta\ ;\ \Omega$)

$$\frac{\mathsf{L}, P@u, Q@u \Longrightarrow \mathsf{R}}{\mathsf{L}, P \otimes Q@u \Longrightarrow \mathsf{R}}\ \otimes L \qquad \frac{\mathsf{L} \Longrightarrow \mathsf{R}}{\mathsf{L}, \mathbf{1}@u \Longrightarrow \mathsf{R}}\ \mathbf{1}L \qquad \frac{\mathsf{L}, P@u \Longrightarrow \mathsf{R} \quad \mathsf{L}, Q@u \Longrightarrow \mathsf{R}}{\mathsf{L}, P \oplus Q@u \Longrightarrow \mathsf{R}}\ \oplus L$$

$$\frac{\mathsf{L}, [v/u]P@v \Longrightarrow \mathsf{R}}{\mathsf{L}, \downarrow u.\ P@v \Longrightarrow \mathsf{R}}\ \downarrow LA \qquad \frac{\mathsf{L}, P@u \Longrightarrow \mathsf{R}}{\mathsf{L}, (P\ \mathtt{at}\ u)@v \Longrightarrow \mathsf{R}}\ \mathtt{at}LA \qquad \frac{\mathsf{L}, P@u \Longrightarrow \mathsf{R}}{\mathsf{L}, \exists\alpha.\ P@u \Longrightarrow \mathsf{R}}\ \exists L^{\alpha}$$

$$\frac{\Gamma, N@u\ ;\ \Delta\ ;\ \Omega \Longrightarrow \mathsf{R}}{\Gamma\ ;\ \Delta\ ;\ \Omega, !N@u \Longrightarrow \mathsf{R}}\ !L \qquad \frac{\Gamma\ ;\ \Delta, N@w\ ;\ \Omega \Longrightarrow \mathsf{R}}{\Gamma\ ;\ \Delta\ ;\ \Omega, \Downarrow N@w \Longrightarrow \mathsf{R}}\ \Downarrow L \qquad \frac{\Gamma\ ;\ \Delta, \Uparrow p\ \vec{t}\ ;\ \Omega \Longrightarrow \mathsf{R}}{\Gamma\ ;\ \Delta\ ;\ \Omega, p\ \vec{t}@w \Longrightarrow \mathsf{R}}\ \mathrm{lp}$$

$$\frac{\mathsf{L} \Longrightarrow M@w\ ;\ \cdot \quad \mathsf{L} \Longrightarrow N@w\ ;\ \cdot}{\mathsf{L} \Longrightarrow M\ \&\ N@w\ ;\ \cdot}\ \&R \qquad \frac{}{\mathsf{L} \Longrightarrow \top@w\ ;\ \cdot}\ \top R \qquad \frac{\mathsf{L}, P@w \Longrightarrow N@w\ ;\ \cdot}{\mathsf{L} \Longrightarrow P \multimap N@w\ ;\ \cdot}\ \multimap R$$

$$\frac{\mathsf{L} \Longrightarrow [w/u]N@w\ ;\ \cdot}{\mathsf{L} \Longrightarrow \downarrow u.\ N@w\ ;\ \cdot}\ \downarrow RA \qquad \frac{\mathsf{L} \Longrightarrow N@u}{\mathsf{L} \Longrightarrow (N\ \mathtt{at}\ u)@w}\ \mathtt{at}RA \qquad \frac{\mathsf{L} \Longrightarrow N@u\ ;\ \cdot}{\mathsf{L} \Longrightarrow \forall\alpha.\ N@u\ ;\ \cdot}\ \forall R^{\alpha}$$

$$\frac{\mathsf{L} \Longrightarrow \cdot\ ;\ P@w}{\mathsf{L} \Longrightarrow \Uparrow P@w\ ;\ \cdot}\ \Uparrow R \qquad \frac{\mathsf{L} \Longrightarrow \cdot\ ;\ \Downarrow n\ \vec{t}@w}{\mathsf{L} \Longrightarrow n\ \vec{t}@w\ ;\ \cdot}\ \mathrm{rp} \qquad \frac{}{\mathsf{L}, \mathbf{0}@u \Longrightarrow \mathsf{R}}\ \mathbf{0}L$$

Focusing decisions

$$\frac{\Gamma\ ;\ \Delta\ ;\ \left[N@u\right] \Longrightarrow Q@w \quad N \text{ not } \Uparrow p\ \vec{t}}{\Gamma\ ;\ \Delta, N@u\ ;\ \cdot \Longrightarrow \cdot\ ;\ Q@w}\ \mathrm{lf} \qquad \frac{\Gamma, N@u\ ;\ \Delta\ ;\ \left[N@u\right] \Longrightarrow Q@w}{\Gamma, N@u\ ;\ \Delta\ ;\ \cdot \Longrightarrow \cdot\ ;\ Q@w}\ \mathrm{cplf}$$

$$\frac{\Gamma\ ;\ \Delta \Longrightarrow \left[P@w\right] \quad P \text{ not } \Downarrow n\ \vec{t}}{\Gamma\ ;\ \Delta\ ;\ \cdot \Longrightarrow \cdot\ ;\ P@w}\ \mathrm{rf}$$

Figure 2 Focusing rules for HyLL.

$$\left.\begin{array}{l}\Gamma\ ;\ \Delta\ ;\ \Omega \Longrightarrow \cdot\ ;\ P@w \\ \Gamma\ ;\ \Delta\ ;\ \Omega \Longrightarrow N@w\ ;\ \cdot\end{array}\right\}\ \text{active} \qquad \left.\begin{array}{l}\Gamma\ ;\ \Delta\ ;\ \left[N@u\right] \Longrightarrow P@w \\ \Gamma\ ;\ \Delta \Longrightarrow \left[P@w\right]\end{array}\right\}\ \text{focused}$$

In each case, Γ and Δ contain only negative propositions (*i.e.*, of the form $N@u$) and Ω only positive propositions (*i.e.*, of the form $P@u$). The full collection of inference rules are in fig. 2. The sequent form $\Gamma\ ;\ \Delta\ ;\ \cdot \Longrightarrow \cdot\ ;\ P@w$ is called a *neutral sequent*; from such a sequent, a left or right focused sequent is produced with the rules lf, cplf or rf. Focused logical rules are applied (non-deterministically) and focus persists unto the subformulas of the focused proposition as long as they are of the same polarity; when the polarity switches, the result is an active sequent, where the propositions in the innermost zones are decomposed in an irrelevant order until once again a neutral sequent results.

Soundness of the focusing calculus with respect to the ordinary sequent calculus is immediate by simple structural induction. In each case, if we forget the additional structure in the focused derivations, then we obtain simply an unfocused proof. We omit the obvious theorem. Completeness, on the other hand, is a hard result. We omit the proof because focusing is by now well known for linear logic, with a number of distinct proofs via focused cut-elimination (see *e.g.* the detailed proof in [13]). The hybrid connectives pose no problems because they allow all cut-permutations.

▶ **Theorem 10** (focusing completeness). *Let Γ^- and $C^-@w$ be negative polarizations of Γ and $C@w$ (that is, adding $\Uparrow$ and $\Downarrow$ to make C and each proposition in Γ negative) and Δ^+ be a positive polarization of Δ. If $\Gamma\ ;\ \Delta \Longrightarrow C@w$, then $\cdot\ ;\ \cdot\ ;\ !\Gamma^-, \Delta^+ \Longrightarrow C^-@w\ ;\ \cdot$.*

4 Encoding the Synchronous Stochastic π-calculus

In this section, we shall illustrate the use of HyLL($\mathcal{R}$) (Definiition 9) as a logical framework for constrained transition systems by encoding the syntax and the operational semantics of the synchronous stochastic π-calculus ($S\pi$), which extends the ordinary π-calculus by assigning to every channel and internal action an *inherent* rate of synchronization. In $S\pi$, each rate characterises an exponential distribution such that the probability of a reaction with rate r occuring within time t is given by $1 - e^{-rt}$ [31], where the *rate* r is a parameter. We shall encode $S\pi$ in HyLL($\mathcal{R}$): a $S\pi$ reaction with rate r will be encoded by a transition of a probability described by a random variable with exponential distribution of parameter r; Worlds r in $\mathcal{R}$ will represent the list of the *rates* of the transitions performed so far.

HyLL($\mathcal{R}$) can therefore be seen as a formal language for expressing $S\pi$ executions (traces). For the rest of this section we shall use $r, s, t, \ldots$ instead of $u, v, w, \ldots$ to highlight the fact that the worlds represent (lists of) rates (overloading single elements and the list of single elements). We do not directly use rates because the syntax and transitions of $S\pi$ are given generically for a π-calculus with labelled actions, and it is only the interpretation of the labels that involves probabilities.

We first summarize the syntax of $S\pi$, which is a minor variant of a number of similar presentations such as [31]. For hygienic reasons we divide entities into the syntactic categories of *processes* ($P, Q, \ldots$) and *sums* ($M, N, \ldots$), defined as follows. We also include environments of recursive definitions (E) for constants.

$$\begin{array}{lll} \textit{(Processes)} & P, Q, \ldots & ::= \nu_r\ P \mid P \mid Q \mid 0 \mid X_n\, x_1 \cdots x_n \mid M \\ \textit{(Sums)} & M, N, \ldots & ::= \overline{x}\langle y\rangle.\ P \mid x.\ P \mid \tau_r.\ P \mid M + N \\ \textit{(Environments)} & E & ::= E, X_n \triangleq P \mid \cdot \end{array}$$

$P \mid Q$ is the parallel composition of P and Q, with unit 0. The restriction $\nu_r\ P$ abstracts over a free channel x in the process $P\,x$. We write the process using higher-order abstract

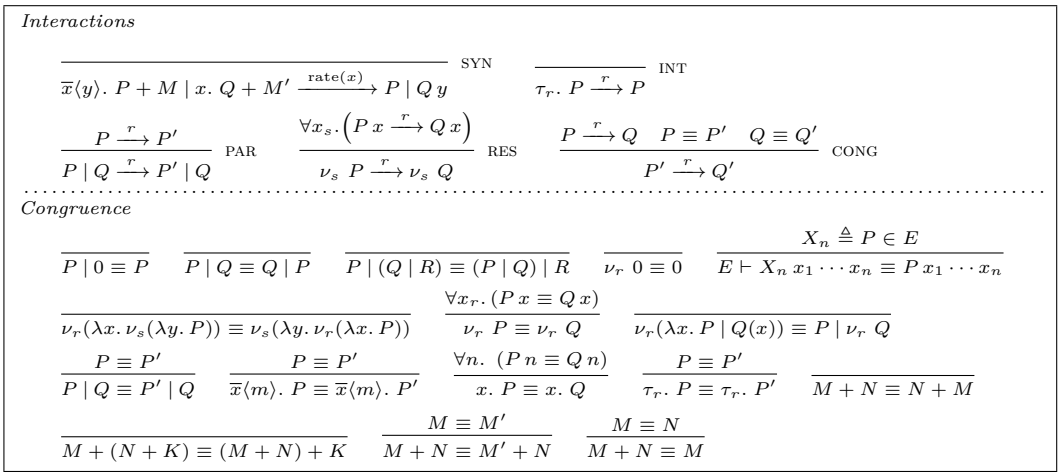

Figure 3 Interactions and congruence in $S\pi$. The environment E is elided in most rules.

syntax [28], *i.e.*, P in $\nu_r\ P$ is (syntactically) a function from channels to processes. This style lets us avoid cumbersome binding rules in the interactions because we reuse the well-understood binding structure of the λ-calculus. A similar approach was taken in the earliest encoding of the (ordinary) π-calculus in (unfocused) linear logic [25], and is also present in the encoding in CLF [9].

A sum is a non-empty choice ($+$) over terms with *action prefixes*: the output action $\overline{x}\langle y\rangle$ sends y along channel x, the input action x reads a value from x (which is applied to its continuation process), and the internal action τ_r has no observable I/O behaviour. Replication of processes happens via guarded recursive definitions [26]; in [34] it is argued that they are more practical for programming than the replication operator !. In a definition $X_n \triangleq P$, X_n denotes a (higher-order) defined constant of arity n; given channels $x_1, \ldots, x_n$, the process $X_n\, x_1 \cdots x_n$ is synonymous with $P\, x_1 \cdots x_n$. The constant X_n may occur on the right hand side of any definition in E, including in its body P, as long as it is prefixed by an action; this prevents infinite recursion without progress.

Interactions are of the form $E \vdash P \xrightarrow{r} Q$ denoting a transition from the process P to the process Q, in a global environment E, by performing an action at rate r. Each channel x is associated with an inherent rate specific to the channel, and internal actions τ_r have rate r. The restriction $\nu_r\ P$ defines the rate of the abstracted channel as r.

The full set of interactions and congruences are in fig. 3. We generally omit the global environment E in the rules as it never changes. It is possible to use the congruences to compute a normal form for processes that are a parallel composition of sums and each reaction selects two suitable sums to synchronise on a channel until there are no further reactions possible; this refinement of the operational semantics is used in $S\pi$ simulators such as SPiM [30].

▶ **Definition 11** (syntax encoding).

1. The encoding of the process P as a positive proposition, written $[\![P]\!]_{\mathrm{p}}$, is as follows (`sel` is a positive atom and `rt` a negative atom).

$$[\![P \mid Q]\!]_{\mathrm{p}} = [\![P]\!]_{\mathrm{p}} \otimes [\![Q]\!]_{\mathrm{p}} \qquad [\![\nu_r\ P]\!]_{\mathrm{p}} = \exists x.\ !(\mathtt{rt}\ x\ \mathtt{at}\ r) \otimes [\![P\, x]\!]_{\mathrm{p}}$$
$$[\![0]\!]_{\mathrm{p}} = \mathbf{1} \qquad [\![X_n\, x_1 \cdots x_n]\!]_{\mathrm{p}} = X_n\, x_1 \cdots x_n$$
$$[\![M]\!]_{\mathrm{p}} = \Downarrow(\mathtt{sel} \multimap [\![M]\!]_{\mathrm{s}})$$

2. The encoding of the sum M as a negative proposition, written $[\![M]\!]_{\mathrm{s}}$, is as follows (`out`, `in` and `tau` are positive atoms).

$$[\![M+N]\!]_{\mathrm{s}} = [\![M]\!]_{\mathrm{s}} \mathbin{\&} [\![N]\!]_{\mathrm{s}} \qquad [\![\overline{x}\langle m\rangle.\ P]\!]_{\mathrm{s}} = \Uparrow(\mathtt{out}\ x\ m \otimes [\![P]\!]_{\mathrm{p}})$$
$$[\![x.\ P]\!]_{\mathrm{s}} = \forall n. \Uparrow(\mathtt{in}\ x\ n \otimes [\![P\,n]\!]_{\mathrm{p}}) \qquad [\![\tau_r.\ P]\!]_{\mathrm{s}} = \Uparrow(\mathtt{tau}\ r \otimes [\![P]\!]_{\mathrm{p}})$$

3. The encoding of the definitions E as a context, written $[\![E]\!]_{\mathrm{e}}$, is as follows.

$$\left[\!\!\left[E, X_n \triangleq P\right]\!\!\right]_{\mathrm{e}} = [\![E]\!]_{\mathrm{e}}, \dagger\forall x_1, \ldots, x_n.\ X_n\, x_1 \cdots x_n \equiv [\![P\, x_1 \cdots x_n]\!]_{\mathrm{p}}$$
$$[\![\cdot]\!]_{\mathrm{e}} = \cdot$$

where $P \equiv Q$ is defined as $(P \multimap \Uparrow Q) \mathbin{\&} (Q \multimap \Uparrow P)$.

The encoding of processes is positive, so they will be decomposed in the active phase when they occur on the left of the sequent arrow, leaving a collection of sums. The encoding of restrictions will introduce a fresh unrestricted assumption about the rate of the restricted channel. Each sum encoded as a processes undergoes a polarity switch because $\multimap$ is negative; the antecedent of this implication is a *guard* `sel`. This pattern of guarded switching of polarities prevents unsound congruences such as $\overline{x}\langle m\rangle.\ \overline{y}\langle n\rangle.\ P \equiv \overline{y}\langle n\rangle.\ \overline{x}\langle m\rangle.\ P$ that do not hold for the synchronous π calculus. To see this, note that $[\![\overline{x}\langle m\rangle.\ \overline{y}\langle n\rangle.\ P]\!]_{\mathrm{p}}$ has the form $X \multimap (A \otimes (X \multimap B \otimes C))$ (eliding the polarity shifts) which is not provably equivalent to $X \multimap (B \otimes (X \multimap A \otimes C))$ in both linear logic and HyLL. Thus, even though we use a commutative connective $\otimes$ in $[\![\overline{x}\langle m\rangle.\ P]\!]_{\mathrm{s}}$, output actions are still sequential and synchronous.

The guard `sel` also *locks* the sums in the context: the $S\pi$ interaction rules INT and SYN discard the non-interacting terms of the sum, so the environment will contain the requisite number of `sel`s only when an interaction is in progress. The action prefixes themselves are also synchronous, which causes another polarity switch. Each action releases a token of its respective kind—`out`, `in` or `tau`—into the context. These tokens must be consumed by the interaction before the next interaction occurs. For each action, the (encoding of the) continuation process is also released into the context.

The proof of the following congruence lemma is omitted. Because the encoding is (essentially) a $\otimes$/& structure, there are no distributive laws in linear logic that would break the process/sum structure.

▶ **Theorem 12** (congruence). *$E \vdash P \equiv Q$ iff both $[\![E]\!]_{\mathrm{e}} @\iota\ ;\ \cdot\ ;\ [\![P]\!]_{\mathrm{p}} @\iota \Longrightarrow \cdot\ ;\ [\![Q]\!]_{\mathrm{p}} @\iota$ and $[\![E]\!]_{\mathrm{e}} @\iota\ ;\ \cdot\ ;\ [\![Q]\!]_{\mathrm{p}} @\iota \Longrightarrow \cdot\ ;\ [\![P]\!]_{\mathrm{p}} @\iota$.*

Now we encode the interactions. Because processes were lifted into propositions, we can be parsimonious with our encoding of interactions by limiting ourselves to the atomic interactions SYN and INT (below); the PAR, RES and CONG interactions will be ambiently implemented by the logic. Because there are no concurrent interactions—only one interaction can trigger at a time in a trace—the interaction rules must obey a locking discipline. We represent this lock as the proposition `act` that is consumed at the start of an interaction and produced again at the end. This lock also carries the net rate of the prefix of the trace so far: that is, an interaction $P \xrightarrow{r} Q$ will update the lock from $\mathtt{act}@s$ to $\mathtt{act}@s \cdot r$. The encoding of individual atomic interactions must also remove the `in`, `out` and `tau` tokens introduced in context by the interacting processes.

▶ **Definition 13** (interaction).
Let $\mathtt{inter} \triangleq \dagger(\mathtt{act} \multimap \Uparrow\mathtt{int} \mathbin{\&} \Uparrow\mathtt{syn})$ where `act` is a positive atom and `int` and `syn` are as follows:

$$\mathtt{int} \triangleq (\mathtt{sel}\ \mathtt{at}\ \iota) \otimes \Downarrow\forall r. \left((\mathtt{tau}\, r\ \mathtt{at}\ \iota) \multimap \rho_r \Uparrow\mathtt{act}\right)$$

$$\mathtt{syn} \triangleq (\mathtt{sel} \otimes \mathtt{sel}\,\mathtt{at}\,\iota) \otimes \Downarrow\forall x, r, m.\, \Big((\mathtt{out}\,x\;m \otimes \mathtt{in}\,x\;m\,\mathtt{at}\,\iota) \multimap \Downarrow(\mathtt{rt}\,x\,\mathtt{at}\,r) \multimap \rho_r \Uparrow\mathtt{act}\Big).$$

The number of interactions that are allowed depends on the number of instances of `inter` in the linear context: each focus on `inter` implements a single interaction. If we are interested in all finite traces, we will add `inter` to the unrestricted context so it may be reused as many times as needed.

4.1 Representational Adequacy.

Adequacy consists of two components: completeness and soundness. Completeness is the property that every $S\pi$ execution is obtainable as a HyLL derivation using this encoding, and is the comparatively simpler direction (see thm. 16). Soundness is the reverse property, and is false for unfocused HyLL as such. However, it *does* hold for focused proofs (see thm. 18). In both cases, we reason about the following canonical sequents of HyLL.

▶ **Definition 14.** The *canonical context of* P, written $\lfloor P \rfloor$, is given by:

$$\lfloor X_n\,x_1 \cdots x_n \rfloor = \Uparrow X_n\,x_1 \cdots x_n \qquad \lfloor P \mid Q \rfloor = \lfloor P \rfloor, \lfloor Q \rfloor \qquad \lfloor 0 \rfloor = \cdot \qquad \lfloor \nu_r\; P \rfloor = \lfloor P\,a \rfloor$$
$$\lfloor M \rfloor = \mathtt{sel} \multimap [\![M]\!]_{\mathrm{s}}$$

For $\lfloor \nu_r\; P \rfloor$, the right hand side uses a *fresh* channel a that is not free in the rest of the sequent it occurs in.

As an illustration, take $P \triangleq \overline{x}\langle a \rangle.\, Q \mid x.\, R$. We have:

$$\lfloor P \rfloor = \quad \mathtt{sel} \multimap \Uparrow(\mathtt{out}\,x\;a \otimes [\![Q]\!]_{\mathrm{p}}), \mathtt{sel} \multimap \forall y. \Uparrow(\mathtt{in}\,x\;y \otimes [\![R\,y]\!]_{\mathrm{p}})$$

Obviously, the canonical context is what would be emitted to the linear zone at the end of the active phase if $[\![P]\!]_{\mathrm{p}}$ were to be present in the left active zone.

▶ **Definition 15.** A neutral sequent is *canonical* iff it has the shape

$$[\![E]\!]_{\mathrm{e}}, \mathtt{rates}, \mathtt{inter}@\iota\;;\; \Uparrow\mathtt{act}@s, \lfloor P_1 \mid \cdots \mid P_k \rfloor @\iota\;;\; \cdot \Longrightarrow \cdot\;;\; ([\![Q]\!]_{\mathrm{p}}\;\mathtt{at}\;\iota) \otimes \mathtt{act}@t$$

where `rates` contains elements of the form $\mathtt{rt}\,x@r$ defining the rate of the channel x as r, and all free channels in $[\![E]\!]_{\mathrm{e}}, \lfloor P_1 \mid \cdots \mid P_k \mid Q \rfloor$ have a single such entry in `rates`.

Figure 4 contains an example of a derivation for a canonical sequent involving P. Focusing on any (encoding of a) sum in $\lfloor P \rfloor @\iota$ will fail because there is no `sel` in the context, so only `inter` can be given focus; this will consume the `act` and release two copies of $(\mathtt{sel}\,\mathtt{at}\,\iota)$ and the continuation into the context. Focusing on the latter will fail now (because $\mathtt{out}\,x\;m$ and $\mathtt{in}\,x\;m$ (for some m) are not yet available), so the only applicable foci are the two sums that can now be "unlocked" using the `sel`s. The output and input can be unlocked in an irrelevant order, producing two tokens $\mathtt{in}\,x\;a$ and $\mathtt{out}\,x\;a$. Note in particular that the witness a was chosen for the universal quantifier in the encoding of $x.\, Q$ because the subsequent consumption of these two tokens requires the messages to be identical. (Any other choice will not lead to a successful proof.) After both tokens are consumed, we get the final form $\mathtt{act}@s \cdot r$, where r is the inherent rate of x (found from the `rates` component of the unrestricted zone). This sequent is canonical and contains $\lfloor Q \mid R\,a \rfloor$.

Our encoding therefore represents every $S\pi$ action in terms of "micro" actions in the following rigid order: one micro action to determine what kind of action (internal or synchronization), one micro action per sum to select the term(s) that will interact, and finally

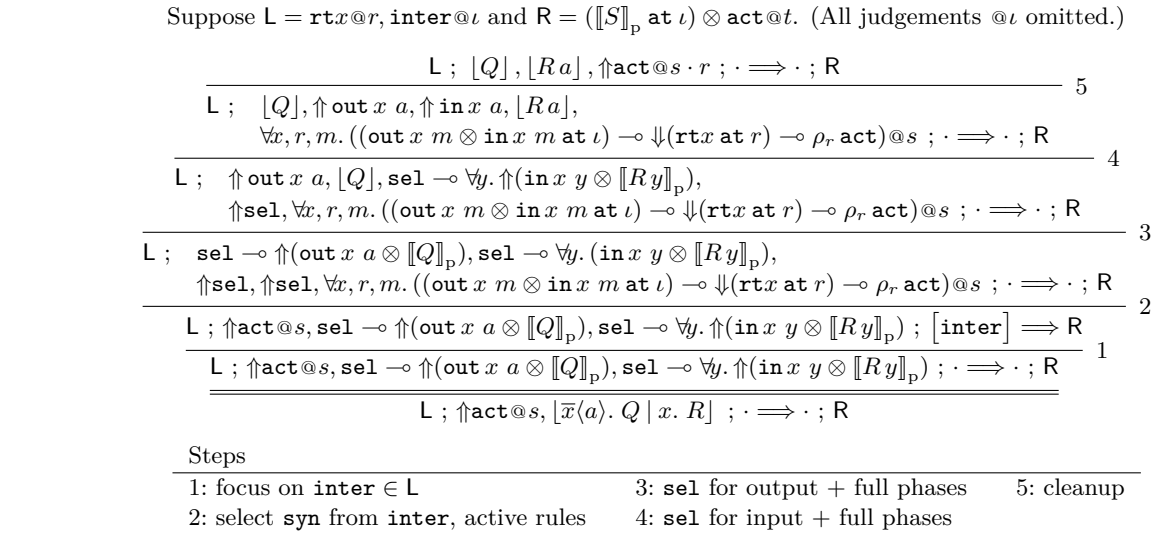

Figure 4 Example interaction in the $S\pi$-encoding.

one micro action to establish the contract of the action. Thus we see that focusing is crucial to maintain the semantic interpretation of (neutral) sequents. In an unfocused calculus, several of these steps could have partial overlaps, making such a semantic interpretation inordinately complicated. We do not know of any encoding of the π calculus that can provide such interpretations in unfocused sequents without changing the underlying logic. In CLF [9] the logic is extended with explicit monadic staging, and this enables a form of adequacy [9]; however, the encoding is considerably more complex because processes and sums cannot be fully lifted and must instead be specified in terms of a lifting computation. Adequacy is then obtained via a permutative equivalence over the lifting operation. Other encodings of π calculi in linear logic, such as [20] and [3], concentrate on the easier asynchronous fragment and lack adequacy proofs anyhow.

▶ **Theorem 16** (completeness). *If $E \vdash P \xrightarrow{r} Q$, then the following canonical sequent is derivable.*

$$[\![E]\!]_{\mathrm{e}}, \mathtt{rates}, \mathtt{inter}\,@\iota \;;\; \Uparrow \mathtt{act}\,@s, \lfloor P \rfloor\,@\iota \;;\; \cdot \Longrightarrow \cdot \;;\; ([\![Q]\!]_{\mathrm{p}}\;\mathtt{at}\;\iota) \otimes \mathtt{act}\,@s \cdot r.$$

Proof. By structural induction of the derivation of $E \vdash P \xrightarrow{r} Q$. Every interaction rule of $S\pi$ is implementable as an admissible inference rule for canonical sequents. For CONG, we appeal to thm. 12. ◀

Completeness is a testament to the expressivity of the logic – all executions of $S\pi$ are also expressible in HyLL. However, we also require the opposite (soundness) direction: that every canonical sequent encodes a possible $S\pi$ trace. The proof hinges on the following canonicity lemma.

▶ **Lemma 17** (canonical derivations). *In a derivation for a canonical sequent, the derived inference rules for* inter *are of one of the two following forms (conclusions and premises canonical).*

$$\frac{}{[\![E]\!]_{\mathrm{e}}, \mathtt{rates}, \mathtt{inter}\,@\iota \;;\; \Uparrow \mathtt{act}\,@s, \lfloor P \rfloor\,@\iota \;;\; \cdot \Longrightarrow \cdot \;;\; ([\![P]\!]_{\mathrm{p}}\;\mathtt{at}\;\iota) \otimes \mathtt{act}\,@s}$$

$$\frac{[\![E]\!]_{\mathrm{e}}, \mathtt{rates}, \mathtt{inter}\,@\iota \;;\; \Uparrow \mathtt{act}\,@s \cdot r, \lfloor Q \rfloor\,@\iota \;;\; \cdot \Longrightarrow \cdot \;;\; ([\![R]\!]_{\mathrm{p}}\,\mathtt{at}\,\iota) \otimes \mathtt{act}\,@t}{[\![E]\!]_{\mathrm{e}}, \mathtt{rates}, \mathtt{inter}\,@\iota \;;\; \Uparrow \mathtt{act}\,@s, \lfloor P \rfloor\,@\iota \;;\; \cdot \Longrightarrow \cdot \;;\; ([\![R]\!]_{\mathrm{p}}\,\mathtt{at}\,\iota) \otimes \mathtt{act}\,@t}$$

where: either $E \vdash P \equiv Q$ *with* $r = \iota$ *or* $E \vdash P \xrightarrow{r} Q$.

Proof. This is a formal statement of the phenomenon observed earlier in the example (fig. 4): $[\![R]\!]_{\mathrm{p}} \otimes \mathtt{act}$ cannot be focused on the right unless $P \equiv R$, in which case the derivation ends with no more foci on `inter`. If not, the only elements available for focus are `inter` and one of the congruence rules $[\![E]\!]_{\mathrm{e}}$ in the unrestricted context. In the former case, the definition of a top level X_n in $\lfloor P \rfloor$ is (un)folded (without advancing the world). In the latter case, the derived rule consumes the $\Uparrow \mathtt{act}\,@s$, and by the time `act` is produced again, its world has advanced to $s \cdot r$. The proof proceeds by induction on the structure of P. ◀

Lemma 17 is a strong statement about HyLL derivations using this encoding: every partial derivation using the derived inference rules represents a prefix of an $S\pi$ trace. This is sometimes referred to as *full adequacy*, to distinguish it from adequacy proofs that require complete derivations [27]. The structure of focused derivations is crucial because it allows us to close branches early (using init). It is impossible to perform a similar analysis on unfocused proofs for this encoding; both the encoding and the framework will need further features to implement a form of staging [9, Chapter 3].

▶ **Corollary 18.** *If* $[\![E]\!]_{\mathrm{e}}, \mathtt{rates}, \mathtt{inter}\,@\iota \;;\; \Uparrow \mathtt{act}\,@\iota, \lfloor P \rfloor\,@\iota \;;\; \cdot \Longrightarrow \cdot \;;\; ([\![Q]\!]_{\mathrm{p}}\,\mathtt{at}\,\iota) \otimes \mathtt{act}\,@r$ *is derivable, then* $E \vdash P \xrightarrow{r} {}^{*} Q$.

Proof. Directly from lem. 17. ◀

4.2 Stochastic Correctness with respect to simulation

So far the HyLL($\mathcal{R}$) encoding of $S\pi$ represents any $S\pi$ trace symbolically. However, not every symbolic trace of an $S\pi$ process can be produced according to the operational semantics of $S\pi$, which is traditionally given by a simulator. This is the main difference between HyLL (and $S\pi$) and the approach of CSL [2], where the truth of a proposition is evaluated against a CTMC, which is why equivalence in CSL is identical to CTMC bisimulation [16]. In this section we sketch how the execution could be used directly on the canonical sequents to produce only correct traces (proofs). The proposal in this section should be seen by analogy to the execution model of $S\pi$ simulators such as SPiM [29], although we do not use the Gillespie algorithm.

The main problem of simulation is determining which of several competing enabled actions in a canonical sequent to select as the "next" action from the *race condition* of the actions enabled in the sequent. Because of the focusing restriction, these enabled actions are easy to compute. Each element of $\lfloor P \rfloor$ is of the form $\mathtt{sel} \multimap [\![M]\!]_{\mathrm{s}}$, so the enabled actions in that element are given precisely by the topmost occurrences of $\Uparrow$ in $[\![M]\!]_{\mathrm{s}}$. Because none of the sums can have any restricted channels (they have all been removed in the active decomposition of the process earlier), the rates of all the channels will be found in the `rates` component of the canonical sequent.

The effective rate of a channel x is related to its inherent rate by scaling by a factor proportional to the *activity* on the channel, as defined in [29]. Note that this definition is on the *rate constants* of exponential distributions, not the rates themselves. The distribution of the minimum of a list of random variables with exponential distribution is itself an exponential distribution whose rate constant is the sum of those of the individual variables.

Each individual transition on a channel is then weighted by the contribution of its rate to this sum. The choice of the transition to select is just the ordinary logical non-determinism. Note that the rounds of the algorithm do not have an associated *delay* element as in [29]; instead, we compute (symbolically) a distribution over the delays of a sequence of actions.

Because stochastic correctness is not necessary for the main adequacy result in the previous subsection, we leave the details of simulation to future work.

5 Related Work

Logically, the HyLL sequent calculus is a variant of labelled deduction [37], a very broad topic not elaborated on here. The combination of linear logic with labelled deduction isn't new to this work. In the η-logic [17] the constraint domain is intervals of time, and the rules of the logic generate constraint inequalities as a side-effect; however its sole aim is the representation of proof-carrying authentication, and it does not deal with the full generality of constraint domains or with focusing. The main feature of η not in HyLL is a separate constraint context that gives new constrained propositions. HyLL is also related to the Hybrid Logical Framework (HLF) [33] which captures linear logic itself as a labelled form of intuitionistic logic. Encoding constrained π calculi directly in HLF would be an interesting exercise: we would combine the encoding of linear logic with the constraints of the process calculus. Because HLF is a very weak logic with a proof theory based on natural deduction, it is not clear whether (and in what forms) an adequacy result in HyLL can be transferred to HLF.

Temporal logics such as CSL and PCTL [22] are popular for logical reasoning on temporal properties of transition systems with probabilities. In such logics, truth is defined in terms of correctness with respect to a constrained forcing relation on the constraint algebra. In CSL and PCTL states are formal entities (names) labeled with atomic propositions. Formulae are interpreted on algebraic structures that are discrete (in PCTL) or continuous (in CSL) time Markov chains. Transitions between states are viewed as pairs of states labeled with a probability (the probability of the transition), which is defined as a function from $S \times S$ into $[0, 1]$, where S is the set of states. While such logics have been very successful in practice with efficient tools, the proof theory of these logics is very complex. Indeed, such modal logics generally cannot be formulated in the sequent calculus, and therefore lack cut-elimination and focusing. In contrast, HyLL has a very traditional proof theoretic pedigree, but lacks such a close correspondence between logical and algebraic equivalence. Probably the most well known and relevant stochastic formalism not already discussed is that of stochastic Petri-nets [24], which have a number of sophisticated model checking tools, including the PRISM framework [23]. Recent advances in proof theory suggest that the benefits of model checking can be obtained without sacrificing proofs and proof search [4].

6 Conclusion and Future Work

We have presented HyLL, a hybrid extension of intuitionistic linear logic with a simple notion of situated truth, a traditional sequent calculus with cut-elimination and focusing, and a modular and instantiable constraint system (set of worlds) that can be directly manipulated using hybrid connectives. We have proposed two instances of HyLL (i.e two particular instances of the set of worlds): one modelling temporal constraints and the others modelling stochastic (continuous time Markov processes) constraints. We have shown how to obtain representationally adequate encodings of constrained transition systems, such as the

synchronous stochastic π-calculus in a suitable instance of HyLL.

Several instantiations of HyLL besides the ones in this paper seem interesting. For example, we can already use disjunction ($\oplus$) to explain disjunctive states, but it is also possible to obtain a more extensional branching by treating the worlds as points in an arbitrary partially-ordered set instead of a monoid. Another possibility is to consider lists of worlds instead of individual worlds – this would allow defining periodic availability of a resource, such as one being produced by an oscillating process. The most interesting domain is that of discrete probabilities: here the underlying semantics is given by discrete time Markov chains instead of CTMCs, which are often better suited for symbolic evaluation [40].

The logic we have provided so far is a logical framework well suited *to represent* constrained transition systems. The design of a logical framework *for* (i.e. to reason about) constrained transition systems is left for future work -and might be envisioned by using a two-levels logical framework such as the Abella system [19]. The work presented in [15] provides a first step in this direction in the area of systems biology (where biological systems are viewed as transition systems), using the Coq proof assistant [5] with HyLL($\mathcal{T}'$) (with discrete instants of time) as an object logic. This work can be seen as a first possible implementation of HyLL with temporal constraints.

An important open question is whether a general logic such as HyLL can serve as a framework for specialized logics such as CSL and PCTL. A related question is what benefit linearity truly provides for such logics – linearity is obviously crucial for encoding process calculi that are inherently stateful, but CSL requires no such notion of single consumption of resources.

In the κ-calculus, reactions in a biological system are modeled as reductions on graphs with certain state annotations. It appears (though this has not been formalized) that the κ-calculus can be embedded in HyLL even more naturally than $S\pi$, because a solution—a multiset of chemical products—is simply a tensor of all the internal states of the binding sites together with the formed bonds. One important innovation of κ is the ability to extract semantically meaningful "stories" from simulations. We believe that HyLL provides a natural formal language for such stories.

We became interested in the problem of encoding stochastic reasoning in a resource aware logic because we were looking for the logical essence of biochemical reactions. What we envision for the domain of "biological computation" is a resource-aware stochastic or probabilistic λ-calculus that has HyLL propositions as (behavioral) types.

Acknowledgements

This work was partially supported by the Information Society Technologies programme of the European Commission, Future and Emerging Technologies under the IST-2005-015905 MOBIUS project, and by the European TYPES project. We thank François Fages, Sylvain Soliman, Alessandra Carbone, Vincent Danos and Jean Krivine for fruitful discussions on various preliminary versions of the work presented here. Thanks also go to Nicolas Champagnat, Luc Pronzato and André Hirschowitz who helped us understand the algebraic nature of stochastic constraints.

References

1 Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *J. of Logic and Computation*, 2(3):297–347, 1992.

2 A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Model checking continuous time Markov chains. *ACM Transactions on Computational Logic*, 1(1):162–170, 2000.

3 David Baelde. Logique linéaire et algèbre de processus. Technical report, INRIA Futurs, LIX and ENS, 2005.

4 David Baelde, Andrew Gacek, Dale Miller, Gopalan Nadathur, and Alwen Tiu. The Bedwyr system for model checking over syntactic expressions. In F. Pfenning, editor, *21th Conf. on Automated Deduction (CADE)*, number 4603 in LNAI, pages 391–397, New York, 2007. Springer.

5 Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions*. Springer, 2004.

6 Torben Braüner and Valeria de Paiva. Intuitionistic hybrid logic. *Journal of Applied Logic*, 4:231–255, 2006.

7 Luís Caires and Frank Pfenning. Session types as intuitionistic linear propositions. In Paul Gastin and François Laroussinie, editors, *Proceedings of the 21th International Conference on Concurrency Theory (CONCUR)*, volume 6269 of *Lecture Notes in Computer Science*, pages 222–236. Springer, 2010.

8 Iliano Cervesato. Typed multiset rewriting specifications of security protocols. *Electronic Notes on Theoretical Computer Sciience*, 40:8–51, 2000.

9 Iliano Cervesato, Frank Pfenning, David Walker, and Kevin Watkins. A concurrent logical framework II: Examples and applications. Technical Report CMU-CS-02-102, Carnegie Mellon University, 2003. Revised, May 2003.

10 Nathalie Chabrier-Rivier, François Fages, and Sylvain Soliman. The biochemical abstract machine BIOCHAM. In *International Workshop on Computational Methods in Systems Biology (CMSB-2)*, LNCS. Springer, 2004.

11 Bor-Yuh Evan Chang, Kaustuv Chaudhuri, and Frank Pfenning. A judgmental analysis of linear logic. Technical Report CMU-CS-03-131R, Carnegie Mellon University, December 2003.

12 Kaustuv Chaudhuri and Joëlle Despeyroux. A hybrid linear logic for constrained transition systems with applications to molecular biology. Technical Report hal-00402942, INRIA-HAL, October 2013.

13 Kaustuv Chaudhuri, Frank Pfenning, and Greg Price. A logical characterization of forward and backward chaining in the inverse method. *J. of Automated Reasoning*, 40(2-3):133–177, March 2008.

14 Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theor. Comput. Sci.*, 325(1):69–110, 2004.

15 Elisabetta de Maria, Joëlle Despeyroux, and Amy Felty. A logical framework for systems biology. Technical Report hal-00981409, INRIA-HAL, April 2014.

16 Josée Desharmais and Prakash Panangaden. Continuous stochastic logic characterizes bisimulation of continuous-time Markov processes. *Journal of Logic and Algebraic Programming*, 56:99–115, 2003.

17 Henry DeYoung, Deepak Garg, and Frank Pfenning. An authorization logic with explicit time. In *Computer Security Foundations Symposium (CSF-21)*, pages 133–145. IEEE Computer Society, 2008.

18 E. Allen Emerson. Temporal and modal logic. In *TCS*, pages 995–1072. Elsevier, 1995.

19 Andrew Gacek. *A Framework for Specifying, Prototyping, and Reasoning about Computational Systems*. PhD thesis, University of Minnesota, September 2009.

20 Deepak Garg and Frank Pfenning. Type-directed concurrency. In Martín Abadi and Luca de Alfaro, editors, *16th International Conference on Concurrency Theory (CONCUR)*, volume 3653 of *LNCS*, pages 6–20. Springer, 2005.

21 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

22 H. Hansson and B. Jonsson. A logic for reasoning about time and probability. *Formal Aspects of Computing*, 6, 1994.

23 M. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking using PRISM: a hybrid approach. *International Journal of Software Tools for Technology Transfer*, 6(2), 2004.

24 M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Francenschinis. *Modelling with Generalised Stochastic Petri Nets*. Wiley Series in Parallel Computing. Wiley and Sons, 1995.

25 Dale Miller. The π-calculus as a theory in linear logic: Preliminary results. In E. Lamma and P. Mello, editors, *3rd Workshop on Extensions to Logic Programming*, number 660 in LNCS, pages 242–265, Bologna, Italy, 1993. Springer.

26 Robin Milner. *Communicating and Mobile Systems : The π-Calculus*. Cambridge University Press, New York, NY, USA, 1999.

27 Vivek Nigam and Dale Miller. Focusing in linear meta-logic. In *Proceedings of IJCAR: International Joint Conference on Automated Reasoning*, volume 5195 of *LNAI*, pages 507–522. Springer, 2008.

28 Frank Pfenning and Conal Elliott. Higher-order abstract syntax. In *Proceedings of the ACM-SIGPLAN Conference on Programming Language Design and Implementation*, pages 199–208. ACM Press, June 1988.

29 Andrew Phillips and Luca Cardelli. A correct abstract machine for the stochastic pi-calculus. *Concurrent Models in Molecular Biology*, August 2004.

30 Andrew Phillips and Luca Cardelli. A correct abstract machine for the stochastic pi-calculus. In *Proceedings of BioConcur'04*, ENTCS, 2004.

31 Andrew Phillips, Luca Cardelli, and Giuseppe Castagna. A graphical representation for biological processes in the stochastic pi-calculus. *Transactions on Computational Systems Biology VII*, pages 123–152, 2006.

32 Arthur N. Prior. *Time and Modality*. Oxford: Clarendon Press, 1957.

33 Jason Reed. Hybridizing a logical framework. In *International Workshop on Hybrid Logic (HyLo)*, Seattle, USA, August 2006.

34 A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the π-calculus and process algebra. In L. Hunter R. B. Altman, A. K. Dunker and T. E. Klein, editors, *Pacific Symposium on Biocomputing*, volume 6, pages 459–470, Singapore, 2001. World Scientific Press.

35 L. C. G. Rogers and D. Williams. *Diffusions, Markov Processes and Martingales*, volume 1: Foundations. Cambridge Mathematical Library, 2nd edition, 2000.

36 Uluç Saranli and Frank Pfenning. Using constrained intuitionistic linear logic for hybrid robotic planning problems. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3705–3710. IEEE, 2007.

37 Alex Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD thesis, University of Edinburgh, 1994.

38 Philip Wadler. Linear types can change the world. In M. Broy and C. B. Jones, editors, *Proceedings of the IFIP TC 2 Working Conference on Programming Concepts and Methods*, pages 561–581. North Holland, 1990.

39 Kevin Watkins, Iliano Cervesato, Frank Pfenning, and David Walker. A concurrent logical framework I: Judgments and properties. Technical Report CMU-CS-02-101, Carnegie Mellon University, 2003. Revised, May 2003.

40 Peng Wu, Catuscia Palamidessi, and Huimin Lin. Symbolic bisimulations for probabilistic systems. In *QEST'07*, pages 179–188. IEEE Computer Society, 2007.