Math. Struct. in Comp. Science (2019), *vol.* 29, *pp.* 1151–1176. © Cambridge University Press 2019
doi:10.1017/S0960129518000439 First published online 22 April 2019

Hybrid linear logic, revisited

KAUSTUV CHAUDHURI[†], JOËLLE DESPEYROUX[‡],
CARLOS OLARTE[§] and ELAINE PIMENTEL[¶]

[†] *Inria & LIX/École Polytechnique, Palaiseau, France*
Email: `kaustuv.chaudhuri@inria.fr`
[‡] *INRIA and CNRS, I3S, Sophia-Antipolis, France*
Email: `joelle.despeyroux@i3s.unice.fr`
[§] *ECT – Universidade Federal do Rio Grande do Norte, Natal, Brazil*
Email: `carlos.olarte@gmail.com`
[¶] *Departamento de Matemática – Universidade Federal do Rio Grande do Norte, Natal, Brazil*
Email: `elaine.pimentel@gmail.com`

Received 1 June 2017; revised 25 September 2018

HyLL (Hybrid Linear Logic) is an extension of intuitionistic linear logic (ILL) that has been used as a framework for specifying systems that exhibit certain modalities. In HyLL, truth judgements are labelled by *worlds* (having a monoidal structure) and hybrid connectives (`at` and $\downarrow$) relate worlds with formulas. We start this work by showing that HyLL's axioms and rules can be adequately encoded in linear logic (LL), so that one focused step in LL will correspond to a step of derivation in HyLL. This shows that any proof in HyLL can be exactly mimicked by a LL focused derivation. Another extension of LL that has extensively been used for specifying systems with modalities is Subexponential Linear Logic (SELL). In SELL, the LL exponentials ($!$, $?$) are decorated with labels representing *locations*, and a pre-order on such labels defines the provability relation. We propose an encoding of HyLL into SELL$^{\Cap}$ (SELL plus quantification over locations) that gives better insights about the meaning of worlds in HyLL. More precisely, we identify worlds as locations, and show that a flat subexponential structure is sufficient for representing any world structure in HyLL. This shows that HyLL's monoidal structure is not reflected in LL derivations, hence not increasing the expressiveness of LL, from a proof theoretical point of view. We conclude by proposing the notion of fixed points in multiplicative additive HyLL (μHyMALL), which can be encoded into multiplicative additive linear logic with fixed points (μMALL). As an application, we propose encodings of Computational Tree Logic (CTL) into both μMALL and μHyMALL. In the former, states are represented as atoms in the linear context, hence reflecting a more *operational* view of CTL connectives. In the latter, worlds represent states of the transition system, thus exhibiting a pleasant similarity with the *semantics* of CTL.

1. Introduction

Logical frameworks are adequate tools for specifying proof systems, since they support levels of abstraction that facilitate writing declarative specifications of object-level logical systems. Thus, designing suitable logical frameworks for adequately specifying different proof systems has become one of the main tasks for many logicians working in computer science.

Among the many frameworks that have been used for the specification of proof systems, linear logic (LL) (Girard 1987) is one of the most successful ones. This is mainly because LL is resource conscious and, at the same time, it can internalize classical and intuitionistic behaviours (see, for example, (Cervesato and Pfenning 2002; Miller and Pimentel 2013)). However, since specifications of object-level systems into the logical framework should be natural and direct, there are some features that often cannot be adequately captured in LL, e.g., modalities different from the ones present in LL.

Extensions of LL, or its intuitionistic version ILL (Girard 1987), have been proposed in order to fill this gap, hence having stronger logical frameworks that preserve the elegant properties of LL as the underlying logic. Two of such extensions are HyLL (Hybrid Linear Logic) (Despeyroux and Chaudhuri 2014), an extension of ILL, and SELL (Subexponential Linear Logic) (Danos et al. 1993; Nigam and Miller 2009), an extension of ILL/LL.† These logics have been extensively used for specifying systems that exhibit modalities, such as temporal or spatial ones. The difference between HyLL and SELL relies on the way modalities are handled.

In HyLL, truth judgements are labelled by worlds and two hybrid connectives relate worlds with formulas: the satisfaction `at` which states that a proposition is true at a given world, and the localization $\downarrow$ which binds a name for the (current) world the proposition is true at. These constructors allow for the specification of modal connectives, such as $\Box A$ (A is true in all the accessible worlds) and $\Diamond A$ (there exists an accessible world where A holds). The underlying structure on worlds allows for the modelling of transitions systems and the specification of temporal formulas (Despeyroux and Chaudhuri 2014; de Maria et al. 2014).

In SELL, the LL exponentials ($!$, $?$) are decorated with labels: the formula $?^a A$ can be interpreted as *A holds in a location*, *modality*, or *world a*. Such labels are organized in a pre-order, so that if A holds in a, then it can be deduced in any location b smaller then a. Moreover, the formula $?^a !^a A$ means that *A is confined into the location a*, that is, the information A is not propagated to other worlds/locations related to a (Nigam et al. 2017). While LL has only seven logically distinct prefixes of bangs and question-marks (none, $!$, $?$, $!?$, $?!$, $!?!$, $?!?$), SELL allows for an unbounded number of such prefixes (e.g., $!^a ?^c ?^d$). For this, SELL enhances the expressive power of LL as a logical framework.

Since HyLL and SELL share LL/ILL as the base logic, it is reasonable to investigate the relationship between worlds and locations. The first contribution of this work is a careful comparison study of LL, HyLL and SELL. We start by showing a direct encoding of the HyLL's logical rules into LL with the highest level of adequacy, namely, on the level of derivations (Nigam and Miller 2010). This means that there is a 1-1 relation between the set of *derivations* in HyLL with the set of their *interpretations* in LL.

We then propose an encoding of HyLL into SELL$^{\Cap}$ (SELL with quantification over locations (Nigam et al. 2013, 2017)) that gives better insights about the meaning of worlds in HyLL. More precisely, we represent HyLL worlds as locations in SELL and encode HyLL into SELL$^{\Cap}$. We show that a flat subexponential structure is sufficient for representing any world structure in HyLL. This explains better why the worlds in HyLL do not add any expressive power to LL: they cannot control the logical context as the subexponentials do with the promotion rule.

It is worthy noticing that, in HyLL, using judgements that attach formulas to worlds provides a neat tool for specifying systems with modalities (see e.g., the models of biological systems

---

† Intuitionistic and classical SELL are equally expressive (Chaudhuri 2010).

in de Maria et al. (2014)). An elegant property of these models is that, in the same logical framework, it is possible to model the system and also the properties of interest. This is done by first specifying in (a fragment of) Computational Tree Logic (CTL) the desired property and then encoding it as a HyLL formula.

The next contribution of this article is to show that neither the universal CTL path quantifier A (for all paths), nor the temporal CTL formula $\mathsf{EG}F$ (there exists a path where F always holds) can be encoded in HyLL. The main reason is that the definition of such formulas is recursive and hence, one needs to use induction, at the meta-level, to accurately capture their behaviour. Instead of using meta-reasoning, as done in de Maria et al. (2014), we show that CTL formulas can be encoded into multiplicative, additive LL with fixed points (μMALL) (Baelde 2012). For that, we specify the (current) state of the transition system (Kripke structure) as atoms in the linear context and, following the fixed point characterization of CTL (Burch et al. 1992), we encode the whole set of CTL formulas. Such encoding gives a sort of *operational* view of the CTL connectives: when a fixed point formula is unfolded, the current state s is consumed and the resulting premises in the derivation represent some (or all) the successor states from s where the given CTL formula must be proved again. Hence, in order to accurately represent the state transitions as μMALL derivations, the encoding is parametric in the given Kripke structure and it internalizes the accessibility relation as conjunctions/disjunctions on all possible transitions.

In order to give a more loosely coupled encoding with respect to the transition system, we add fixed point operators to multiplicative, additive HyLL (μHyMALL) and present an encoding of CTL into this system. In this case, worlds in HyLL represent states of the transition system and the encoding of CTL connectives quantifies and *moves* formulas on those worlds. Hence, the resulting encoding has a pleasant duality with the semantics of CTL.

The rest of the article is organized as follows. We briefly recall LL in Section 2.1 and HyLL in Section 2.2. The encoding of HyLL logical rules into LL is discussed in Section 3.1. Section 3.2 presents the encoding of HyLL into SELL$^{\Cap}$. We also prove that information confinement, a feature in SELL that is needed to specify spatial systems, cannot be captured in HyLL. Section 4 proposes the system μHyMALL, that enhances multiplicative, additive HyLL with fixed points. The encodings of CTL into μMALL and μHyMALL are described in Sections 5.2 and 5.3, respectively. Section 6 concludes the article.

This article is an extended version of Despeyroux et al. (2017). In the present article, we not only refine several technical details from that work but we also add the notion of fixed points to HyLL. In Despeyroux et al. (2017), we used the well-known system μMALL for showing an encoding of CTL into LL (with fixed points). Although this entails a correct specification, the encoding is itself complex. Our new encoding of CTL into μHyMALL is not only simpler, but closer to the semantical specification of CTL itself. Moreover, the representation of the transition system is less coupled than the one in Despeyroux et al. (2017), thus allowing us to prove meta-theoretical properties of CTL inside the same logical framework.

2. Preliminaries

In this section, we review some of the basic proof theory for LL (Girard 1987) and HyLL (Despeyroux and Chaudhuri 2014).

2.1. *Linear logic and focusing*

By the name LL, we shall mean the logic that results from merging the logical connectives and proof rules of LL (Girard 1987) with the term and quantificational structure of Church's simple theory of types (Church 1940). More precisely, simple types are either *primitive types*, of which o is a reserved primitive type denoting formulas, or *functional types* that are written using an infix arrow $\tau \to \tau'$. A type is a *predicate type* if it is of the form $\tau_1 \to \cdots \to \tau_n \to o$, where $n \geqslant 0$. *Terms* are simply typed λ-terms and we identify two terms up to the usual α, β, and η-conversions. The substitution notation $B[t/x]$ denotes the λ-normal form of the β-redex $(\lambda x.B)t$.

The set of LL formulas is given by the following grammar:

$$F, G ::= p(\vec{t}) \mid p(\vec{t})^{\perp} \mid 1 \mid \mathbf{0} \mid \top \mid \bot \mid F \otimes G \mid F \mathbin{⅋} G \mid F \mathbin{\&} G \mid F \oplus G \mid \exists_{\tau} x.F \mid \forall_{\tau} x.F \mid ?F \mid !\,F,$$

where atomic propositions are applied to a sequence of terms. The logical connectives for LL can be divided into the following groups: the *multiplicative* version of conjunction, true, disjunction, and false, which are written as $\otimes$, 1, $⅋$, $\bot$, respectively; and the *additive* version of these connectives, which are written as $\&$, $\top$, $\oplus$, $\mathbf{0}$, respectively; the *exponentials* $!$ and $?$; and the (typed) universal and existential *quantifiers* $\forall_\tau$ and $\exists_\tau$. In the quantifiers, the syntactic variable τ can range over all non-predicate types: $\forall_\tau$ and $\exists_\tau$ both have type $(\tau \to o) \to o$. The expressions $\forall_\tau \lambda x.B$ and $\exists_\tau \lambda x.B$ are abbreviated as the more usual $\forall_\tau x.B$ and $\exists_\tau x.B$. From this point on, we will drop the subscript τ when it is not important or it can be determined from context. Formulas are taken to be in *negation normal form* using the standard classical LL dualities, e.g., $(F \otimes G)^{\perp} \equiv F^{\perp} \mathbin{⅋} G^{\perp}$. Hence, negation in LL has only atomic scope.

First proposed by Andreoli (1992) for LL, focused proof systems provide normal form proofs for cut-free proofs. The connectives of LL can be divided into two classes: *negative* ($⅋$, $\bot$, $\&$, $\top$, $\forall$, $?$) and *positive* ($\otimes$, 1, $\oplus$, $\mathbf{0}$, $\exists$, $!$). Note that the dual of a negative connective is positive and vice-versa. In general, the introduction rules for negative connectives are all invertible, meaning that the conclusion of any of these introduction rules is equivalent to its premises. The introduction rules for the positive connectives are not necessarily invertible. The notions of negative and positive polarities are extended to formulas in the natural way by considering the outermost connective. Although any bias can be assigned to atomic formulas, this work will consider only *negative* atoms.

The focused system LLF for classical LL is presented in Figure 1. There are two kinds of arrows in this proof system: $\Downarrow$ and $\Uparrow$, and a pair of contexts to the left of the arrows: Γ is a set of formulas whose main connective is a question-mark (being hence the unbounded context), while Δ is a multi-set of linear formulas, behaving as the bounded context. Sequents with the $\Downarrow$ arrow belong to the positive phase and introduce the logical connective of the 'focused' formula (the one to the right of the arrow). Building proofs of such sequents may require non-invertible proof steps to be taken. Sequents with the $\Uparrow$ arrow belong to the negative phase and decompose the multi-set of formulas on the right of the arrow in such a way that only inference rules over negative formulas are applied; the others are 'stored' in the linear context using the rule $R \Uparrow$. The structural rules D_1, D_2 and $R \Downarrow$ make the transition between negative and positive phases. The *positive* phase begins by choosing a positive formula F on which to focus (using D_1, D_2). Positive rules are applied to F until either: 1 or a negated atom is encountered (and the proof must end by applying the initial rules); or the promotion rule ($!$) is applied; or a negative subformula is encountered and the proof switches to the negative phase (using $R \Downarrow$).

This change of phases on proof search is particularly interesting when the focused formula is a *bipole* (Andreoli 1992).

Definition 2.1 (Bipoles). We call a *monopole* a LL formula that is built up from atoms and occurrences of the negative connectives, with the restriction that $?$ has atomic scope. *Bipoles*, on the other hand, are positive formulas built from monopoles and negated atoms using only positive connectives, with the additional restriction that $!$ can only be applied to a monopole.

Using the LL distributive properties, monopoles are equivalent to formulas of the form

$$\forall x_1 \ldots \forall x_p [\&_{i=1,\ldots,n} \parr_{j=1,\ldots,m_i} F_{i,j}],$$

where the $F_{i,j}$ are either atoms or the result of applying $?$ to an atomic formula. Similarly, bipoles can be rewritten as formulas of the form

$$\exists x_1 \ldots \exists x_p [\oplus_{i=1,\ldots,n} \otimes_{j=1,\ldots,m_i} G_{i,j}],$$

where $G_{i,j}$ are either negated atoms, monopole formulas, or the result of applying $!$ to a monopole formula. Notice that the units $\top$, 0, $\bot$ and 1 are 0-ary versions of $\&$, $\oplus$, $\parr$ and $\otimes$, respectively.

Given this normal representation of bipoles and according to the focusing discipline, it turns out that, once introduced, a bipole is completely decomposed into its atomic subformulas, a fact illustrated by the following bipole derivation.

$$\cfrac{\cdots \quad \cfrac{\cfrac{\cdots \quad \cfrac{\Gamma' ; \Delta' \Uparrow \cdot}{\Gamma ; \Delta' \Uparrow \parr_{j=1,\ldots,m_i} ?p_{i,j}}\,[\parr, ?, R \Uparrow] \quad \cdots}{\Gamma ; \Delta' \Uparrow \forall x_1 \ldots \forall x_p [\&_{i=1,\ldots,n} \parr_{j=1,\ldots,m_i} ?p_{i,j}]}\,[\forall, \&]}{\Gamma ; \Delta' \Downarrow\, !\forall x_1 \ldots \forall x_p [\&_{i=1,\ldots,n} \parr_{j=1,\ldots,m_i} ?p_{i,j}]}\,[!] \quad \cdots}{\Gamma ; \Delta \Downarrow \exists x_1 \ldots \exists x_t [\oplus_{i=1,\ldots,k} \otimes_{j=1,\ldots,q_i} G_{i,j}]}\,[\exists, \oplus, \otimes].$$

Here $p_{i,j}$ is atomic for all i, j. If the connective $!$ is not present, then the rule $!$ is replaced by the rule $R \Downarrow$. Notice that the derivation above contains a single positive and a single negative phase. This two phase decomposition will enable us to adequately capture the application of object-level inference rules as will be shown in Section 3.

ILL (Girard 1987), the intuitionistic version of LL, is obtained as usual by restricting, in the two sided presentation of LL, the right multi-set so to have exactly one formula. Hence, the system ILL does not allow the connectives $\parr$ and $?$ and the unit $\bot$, and the rules are the ones for (the two-sided presentation of) LL restricted accordingly, having explicit rules for the linear implication $\multimap$. The specification and verification of systems may use intuitionistic systems (as in, e.g., Caires et al. (2016), Chaudhuri and Reis (2015), de Maria et al. (2014), Nigam (2014), Nigam and Miller (2009), Nigam et al. (2017), Olarte et al. (2018)), or classical systems (see, e.g., Miller and Pimentel (2013), Nigam et al. (2011), Nigam et al. (2016)). In this work, we will specify object logics in LL-based systems.

Negative rules

$$\frac{\Gamma;\Delta \Uparrow L}{\Gamma;\Delta \Uparrow \bot, L}\,[\bot] \qquad \frac{\Gamma;\Delta \Uparrow F, G, L}{\Gamma;\Delta \Uparrow F \parr G, L}\,[\parr] \qquad \frac{\Gamma, F;\Delta \Uparrow L}{\Gamma;\Delta \Uparrow ?F, L}\,[?]$$

$$\frac{}{\Gamma;\Delta \Uparrow \top, L}\,[\top] \qquad \frac{\Gamma;\Delta \Uparrow F, L \quad \Gamma;\Delta \Uparrow G, L}{\Gamma;\Delta \Uparrow F \,\&\, G, L}\,[\&] \qquad \frac{\Gamma;\Delta \Uparrow F[y/x], L}{\Gamma;\Delta \Uparrow \forall x.F, L}\,[\forall]$$

Positive rules

$$\frac{}{\Gamma;\cdot \Downarrow 1}\,[1] \qquad \frac{\Gamma;\Delta_1 \Downarrow F \quad \Gamma;\Delta_2 \Downarrow G}{\Gamma;\Delta_1,\Delta_2 \Downarrow F \otimes G}\,[\otimes] \qquad \frac{\Gamma;\Delta \Downarrow F_i}{\Gamma;\Delta \Downarrow F_1 \oplus F_2}\,[\oplus_i]$$

$$\frac{\Gamma;\Delta \Downarrow F[t/x]}{\Gamma;\Delta \Downarrow \exists x.F}\,[\exists] \qquad \frac{\Gamma;\cdot \Uparrow F}{\Gamma;\cdot \Downarrow\, !F}\,[!]$$

Identity, Decide, and Release rules

$$\frac{}{\Gamma; p(\vec{t}) \Downarrow p(\vec{t})^{\perp}}\,[I_1] \qquad \frac{}{\Gamma, p(\vec{t});\cdot \Downarrow p(\vec{t})^{\perp}}\,[I_2] \qquad \frac{\Gamma;\Delta \Downarrow P}{\Gamma;\Delta, P \Uparrow \cdot}\,[D_1] \qquad \frac{\Gamma, P;\Delta \Downarrow P}{\Gamma, P;\Delta \Uparrow \cdot}\,[D_2]$$

In $[D_1]$ and $[D_2]$, P is not an atom.

$$\frac{\Gamma;\Delta, P_a \Uparrow L}{\Gamma;\Delta \Uparrow P_a, L}\,[R\Uparrow] \qquad \text{provided that } P_a \text{ is positive or an atom}$$

$$\frac{\Gamma;\Delta \Uparrow N}{\Gamma;\Delta \Downarrow N}\,[R\Downarrow] \qquad \text{provided that } N \text{ is negative}$$

Fig. 1. Focused proof linear logic system LLF. Γ is a set, Δ is a multi-set and L is a list of formulas.

2.2. *Hybrid linear logic*

HyLL is a conservative extension of ILL where the truth judgements are labelled by worlds representing constraints on states and state transitions. Judgements of HyLL are of the form 'A is true at world w,' abbreviated as A @ w. Particular choices of worlds produce particular instances of HyLL, e.g., A @ t can be interpreted as 'A is true at time t.' HyLL was first proposed in Despeyroux and Chaudhuri (2014) and it has been used as a logical framework for specifying modalities as well as biological systems (de Maria et al. 2014). Formally, worlds are defined as follows.

Definition 2.2 (HyLL worlds). A *constraint domain* W is a monoid structure $\langle W, ., \iota\rangle$. The elements of W are called *worlds* and its *reachability relation* $\preceq\, : W \times W$ is defined as $u \preceq w$ iff there exists $v \in W$ such that $u.v = w$.

The identity world ι is the $\preceq$-initial and it is intended to represent the lack of any constraints. Thus, the ordinary first-order ILL can be embedded into any instance of HyLL by setting all world labels to the identity. A typical example of constraint domain is $\mathcal{T} = \langle \mathbb{N}, +, 0\rangle$, representing instants of time.

Formulas in HyLL are constructed from atomic propositions, connectives of first-order intuitionistic linear logic (ILL) and the following two hybrid connectives: *satisfaction* (`at`), which states that a proposition is true at a given world ($w, \iota, u.v, \ldots$), and *localization* ($\downarrow$), which binds a name for the current world where the proposition is true at. More precisely, formulas in HyLL are built from

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

Note that world u is bounded in the propositions $\downarrow u.\, A$, $\forall u.\, A$ and $\exists u.\, A$. 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 note that $\downarrow$ and $\mathtt{at}$ commute freely with all non-hybrid connectives (Despeyroux and Chaudhuri 2014).

The sequent calculus presentation of HyLL uses sequents of the form $\Gamma; \Delta \vdash C \ @\ w$, where Γ (*unbounded context*) is a set and Δ (*linear context*) is a multi-set of judgements of the form $A \ @\ w$. Note that in a judgement $A \ @\ w$ and in a proposition $A \ \mathtt{at}\ w$, w can be any expression in W, not only a variable.

The inference rules are depicted in Figure 2. Note that $(A \ \mathtt{at}\ u)$ is a *mobile* proposition: it carries with it the world at which it is true. Both introduction rules for the the other hybrid connective, $\downarrow$, bind the current world. Weakening and contraction are admissible rules for the unbounded context.

The most important structural properties are the admissibility of the general identity and cut theorems. While the first provides a syntactic completeness theorem for the logic, the latter guarantees consistency (i.e., that there is no proof of $.;. \vdash \mathbf{0} \ @\ w$).

Theorem 2.1 (Identity/Cut (Despeyroux and Chaudhuri 2014)).

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

HyLL is conservative with respect to ILL: as long as no hybrid connectives are used, the proofs in HyLL are identical to those in ILL. Moreover, HyLL is more expressive than S5, as it allows direct manipulation of the worlds using the hybrid connectives, while HyLL's δ connective (see Section 5) is not definable in S5.

Finally, we also note that HyLL admits a complete focused proof system. The interested reader can find proofs and further meta-theoretical theorems about HyLL in Despeyroux and Chaudhuri (2014).

3. Relative expressiveness power of HyLL

Different frameworks can be more or less adequate for specifying different systems. While very specific frameworks often provide better encodings for a small range of systems, general frameworks can handle more systems, sometimes not efficiently or in a natural way. Therefore, finding frameworks that are general enough while still adequate and efficient is a key issue. With that in mind, we will compare HyLL with two other LL-based frameworks: LL itself and linear logic with subexponentials (SELL).

We start by proving that HyLL's axioms and rules can be adequately specified in LL. It turns out that any interpretation of a system into another must be *adequate*, in the sense that there must be a 1-1 relation between the sets of interpreted objects with the set of their interpretations. The

Judgmental rules

$$\frac{}{\Gamma; p(\vec{t}) @ w \vdash p(\vec{t}) @ w}\ [\mathit{init}] \qquad \frac{\Gamma, A @ u; \Delta, A @ u \vdash C @ w}{\Gamma, A @ u; \Delta \vdash C @ w}\ [\mathit{copy}]$$

Multiplicative rules

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

$$\frac{}{\Gamma; \cdot \vdash 1 @ w}\ [1R] \qquad \frac{\Gamma; \Delta \vdash C @ w}{\Gamma; \Delta, 1 @ u \vdash C @ w}\ [1L]$$

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

Additive rules

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

$$\frac{\Gamma; \Delta \vdash A @ w \qquad \Gamma; \Delta \vdash B @ w}{\Gamma; \Delta \vdash A \mathbin{\&} B @ w}\ [\& R] \qquad \frac{\Gamma; \Delta, A_i @ u \vdash C @ w}{\Gamma; \Delta, A_1 \mathbin{\&} A_2 @ u \vdash C @ w}\ [\& L_i]$$

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

Quantifier rules

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

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

In $\forall R$ and $\exists L$, α is assumed to be fresh with respect to Γ, Δ, and C.
In $\exists R$ and $\forall L$, τ stands for a term or world, as appropriate.

Exponential rules

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

Hybrid connectives

$$\frac{\Gamma; \Delta \vdash A @ u}{\Gamma; \Delta \vdash (A \ \mathsf{at}\ u) @ w}\ [\mathsf{at}\ R] \qquad \frac{\Gamma; \Delta, A @ u \vdash C @ w}{\Gamma; \Delta, (A \ \mathsf{at}\ u) @ v \vdash C @ w}\ [\mathsf{at}\ L]$$

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

.

Fig. 2. The sequent calculus for HyLL.

level of adequacy can then determine how tight are those systems. We show that our encoding has the highest possible level of adequacy (on the level of derivations – see Nigam and Miller (2010)), so that one step of derivation in HyLL corresponds to one focused step in LL. This means that every proof in HyLL can be exactly mimicked by a derivation in LLF. We note, however, that HyLL enables for more semantical driven specifications when compared to LL, as it will be discussed in Section 5.

Since linear logic with subexponentials (SELL) is a conservative extension of LL, the specification of HyLL into LL trivially implies that HyLL can be similarly encoded in SELL as well. Our approach in Section 3.2, however, will be entirely different: we will interpret worlds as subexponentials, hence having a better meta-level understanding of the behaviour of worlds in HyLL.

3.1. *HyLL and LL*

We briefly recapitulate the basic concepts of the specification of sequent-style calculi in LLF (see Miller and Pimentel (2013) for a more detailed presentation). Let *obj* be the type of object-level formulas and let $\llbracket\cdot\rrbracket$, $\lfloor\cdot\rfloor$, and $\lceil\cdot\rceil$ be meta-level predicates of type $obj \to o$, where o is the primitive type denoting formulas. HyLL sequents of the form $\Gamma;\Delta \vdash C$ will be encoded in LL as $?\llbracket\Gamma\rrbracket \parr \lfloor\Delta\rfloor \parr \lceil C\rceil$ where, if $\Psi = \{F_1, ..., F_n\}$, then $\lfloor\Psi\rfloor = \lfloor F_1\rfloor \parr ... \parr \lfloor F_n\rfloor$ and $?\llbracket\Psi\rrbracket = ?\llbracket F_1\rrbracket \parr ... \parr ?\llbracket F_n\rrbracket$. In that way, the $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ predicate identify which object-level formulas appear on which side of the sequent: brackets down for left (useful mnemonic: $\lfloor$ for 'left') and brackets up for right, while the double brackets $\llbracket\cdot\rrbracket$ identify formulas in the (left) unbounded context.

Inference rules are specified as a rewriting clause that replaces the active formula in the conclusion by the active formulas in the premises. The LL connectives indicate how these object level formulas are connected: contexts are copied (&) or split ($\otimes$), in different inference rules ($\oplus$) or in the same sequent ($\parr$). As a matter of example, the additive version of the inference rules for conjunction in intuitionistic logic

$$\frac{\Gamma, A \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\wedge_{L1} \qquad \frac{\Gamma, B \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\wedge_{L2} \qquad \frac{\Gamma \longrightarrow A \quad \Gamma \longrightarrow B}{\Gamma \longrightarrow A \wedge B}\wedge_R,$$

can be specified as the following bipoles:

$$\wedge_L : \exists A, B.(\lfloor A \wedge B\rfloor^{\perp} \otimes (\lfloor A\rfloor \oplus \lfloor B\rfloor)) \qquad \wedge_R : \exists A, B.(\lceil A \wedge B\rceil^{\perp} \otimes (\lceil A\rceil \,\&\, \lceil B\rceil)).$$

The following definition shows how to encode HyLL inference rules into LL.

Definition 3.1 (HyLL rules into LL). Let `w`, `d`, `h` and `o` denote, respectively, the types for worlds, (first-order) objects, HyLL formulas and LL formulas. Let $\lceil\cdot\rceil$, $\lfloor\cdot\rfloor$ and $\llbracket\cdot\rrbracket$ be predicate of the type $\mathtt{h} \to \mathtt{w} \to \mathtt{o}$ and A, B, C have, respectively, types $\mathtt{w} \to \mathtt{h}$, $\mathtt{d} \to \mathtt{h}$ and `h`. The encoding of HyLL inference rules into LL is depicted in Figure 3 (we omit the encoding of most of the LL connectives that can be found in Miller and Pimentel (2013)).

Observe that left and right inference rules for the hybrid connectives (`at` and $\downarrow$) are the same (Figure 2). This is reflected in the duality of the encoding where we only replace $\lceil\cdot\rceil$ with $\lfloor\cdot\rfloor$. Observe also that the inference rules for the quantifiers (first-order and worlds) look the same.

$$
\begin{array}{lcl lcl}
\multimap L & : & \exists C, C', H, w, v.(\lfloor (C \multimap C')@w \rfloor^{\perp} \otimes \lceil H@v \rceil^{\perp} \otimes \lceil C@w \rceil \otimes (\lfloor C'@w \rfloor \parr \lceil H@v \rceil)) & & & \\
\multimap R & : & \exists C, C', w.(\lceil (C \multimap C')@w \rceil^{\perp} \otimes (\lfloor C@w \rfloor \parr \lceil C'@w \rceil)) & & & \\
!\, L & : & \exists C, w.(\lfloor !C@w \rfloor^{\perp} \otimes ?\llfloor C@w \rrfloor) & !\, R & : & \exists C, w.(\lceil !C@w \rceil^{\perp} \otimes ! \lceil C@w \rceil) \\
Init & : & \exists C, w.(\lfloor C@w \rfloor^{\perp} \otimes \lceil C@w \rceil^{\perp}) & Copy & : & \exists C, w.(\llfloor C@w \rrfloor^{\perp} \otimes \lfloor C@w \rfloor) \\
\mathtt{at}\ R & : & \exists C, u, w.(\lceil (C\ \mathtt{at}\ u)@w \rceil^{\perp} \otimes \lceil C@u \rceil) & \mathtt{at}\ L & : & \exists C, u, w.(\lfloor (C\ \mathtt{at}\ u)@w \rfloor^{\perp} \otimes \lfloor C@u \rfloor) \\
\downarrow R & : & \exists A, u, w.(\lceil \downarrow u.A@w \rceil^{\perp} \otimes \lceil (A\ w)@w \rceil) & \downarrow L & : & \exists A, u, w.(\lfloor \downarrow u.A@w \rfloor^{\perp} \otimes \lfloor (A\ w)@w \rfloor) \\
\forall R(F) & : & \exists B, u.(\lceil \forall x.B@u \rceil^{\perp} \otimes \forall x.\lceil (B\ x)@u \rceil) & \forall L(F) & : & \exists B, u.(\lfloor \forall x.B@u \rfloor^{\perp} \otimes \exists x.\lfloor (B\ x)@u \rfloor) \\
\forall R(W) & : & \exists A, u.(\lceil \forall v.A@u \rceil^{\perp} \otimes \forall v.\lceil (A\ v)@u \rceil) & \forall L(W) & : & \exists A, u.(\lfloor \forall v.A@u \rfloor^{\perp} \otimes \exists v.\lfloor (A\ v)@u \rfloor)
\end{array}
$$

Fig. 3. Specification of HyLL rules into LL (see Definition 3.1).

The difference is on the type of the variables involved. Since A has type $\mathtt{w} \to \mathtt{h}$, the encoding clause $\forall R(W)$ guarantees that the variable v has type $\mathtt{w}$. Analogously, since B has type $\mathtt{d} \to \mathtt{h}$, then x has type $\mathtt{d}$ in the clause $\forall R(F)$. This neat way of controlling the behaviour of objects by using types is also inherited by the encoding of the other object level inference rules.

The following theorem shows that the encoding of HyLL into LL is adequate in the sense that a focused step in LLF corresponds *exactly* to the application of one HyLL inference rule.

Theorem 3.1 (Adequacy). Let Υ be the set of clauses in Figure 3. The sequent $\Gamma; \Delta \vdash F@w$ is provable in HyLL iff $\Upsilon; \cdot \Uparrow \lfloor \Delta \rfloor, ?\llfloor \Gamma \rrfloor, \lceil F@w \rceil$ is provable in LLF. Moreover, the adequacy of the encodings is on the *level of derivations* meaning that, when focusing on a specification clause, the bipole derivation corresponds exactly to applying the introduction rule at the object level.

Proof. We will illustrate here the case for rule $\mathtt{at}_L$, the other cases are similar. Applying the object-level rule

$$
\frac{\Gamma; \Delta, A@u \vdash C@v}{\Gamma; \Delta, (A\ \mathtt{at}\ u)@w \vdash C@v}\ [\mathtt{at}\ L],
$$

corresponds to deciding on the LL formula given by the encoding of the rule $\mathtt{at}_L$ (stored in Υ). Due to focusing, the derivation in LL has necessarily the shape

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\Upsilon, \llfloor \Gamma \rrfloor; \lfloor (A\ \mathtt{at}\ u)@w \rfloor \Downarrow \lfloor (A\ \mathtt{at}\ u)@w \rfloor^{\perp}}\ I_1 \qquad \dfrac{\Upsilon, \llfloor \Gamma \rrfloor; \lfloor \Delta \rfloor, \lceil C@v \rceil, \lfloor A@u \rfloor \Uparrow \cdot}{\Upsilon, \llfloor \Gamma \rrfloor; \lfloor \Delta \rfloor, \lceil C@v \rceil \Downarrow \lfloor A@u \rfloor}\ R\Downarrow, R\Uparrow}{\Upsilon, \llfloor \Gamma \rrfloor; \lfloor \Delta \rfloor, \lfloor (A\ \mathtt{at}\ u)@w \rfloor, \lceil C@v \rceil \Downarrow \lfloor (A\ \mathtt{at}\ u)@w \rfloor^{\perp} \otimes \lfloor A@u \rfloor}\ \otimes}{\Upsilon, \llfloor \Gamma \rrfloor; \lfloor \Delta \rfloor, \lfloor (A\ \mathtt{at}\ u)@w \rfloor, \lceil C@v \rceil \Downarrow \exists C, u, w.(\lfloor (C\ \mathtt{at}\ u)@w \rfloor^{\perp} \otimes \lfloor C@u \rfloor)}\ 3 \times \exists}{\Upsilon, \llfloor \Gamma \rrfloor; \lfloor \Delta \rfloor, \lfloor (A\ \mathtt{at}\ u)@w \rfloor, \lceil C@v \rceil \Uparrow \cdot}\ D_2
$$

Note that the LL formula corresponding to $(A\ \mathtt{at}\ u)@w$ is consumed and, in the end of the focused phase, the encoding of $A@u$ is stored into the linear context. This mimics exactly the application of the Rule $\mathtt{at}_L$ in HyLL. ☐

One may wonder whether it is possible to define an encoding of *formulas* from HyLL to LL by adding an extra argument on atomic predicates to represent the current world. We think that such encoding would not be completely compositional and probably not adequate. First, note that the HyLL judgement $F@w$ applies to arbitrary formulas (not only to atomic propositions). Hence, such an encoding must define an operator $\nabla(F, w)$ that adds w to all the atomic propositions in F. However, this makes more complicated the definition of the hybrid connectives $\downarrow$ and $\mathtt{at}$ since, statically, it is not possible to know the correct binding.

3.2. *HyLL and SELL*

Linear logic with subexponentials (SELL)[‡] shares with LL all its connectives except the exponentials: instead of having a single pair of exponentials ! and ?, SELL may contain as many *subexponentials* (Danos et al. 1993; Nigam and Miller 2009), written $!^a$ and $?^a$, as one needs. The grammar of formulas in SELL is as follows:

$$\begin{aligned} F, G \ ::= \ & p(\vec{t}) \mid p(\vec{t})^\perp \mid \mathbf{0} \mid 1 \mid \top \mid \bot \mid F \otimes G \mid F \oplus G \mid F \parr G \mid F \mathbin{\&} G \mid \\ & \exists x.F \mid \forall x.F \mid !^a F \mid ?^a F \end{aligned}$$

The proof system for SELL is specified by a *subexponential signature* $\langle I, \preceq, U \rangle$, where I is a set of labels, $U \subseteq I$ is a set specifying which subexponentials allow weakening and contraction, and $\preceq$ is a pre-order among the elements of I. We shall use $a, b, \ldots$ to range over elements in I and we will assume that $\preceq$ is upwardly closed with respect to U, *i.e.*, if $a \in U$ and $a \preceq b$, then $b \in U$.

The system SELL is constructed by adding all the rules for the LL connectives except those for the exponentials. The rules for subexponentials are dereliction and promotion of the subexponentials labelled with $a \in I$

$$\frac{\vdash ?^{a_1}F_1, \ldots ?^{a_n}F_n, G}{\vdash ?^{a_1}F_1, \ldots ?^{a_n}F_n, !^a G}\ !^a \qquad \frac{\vdash \Delta, G}{\vdash \Delta, ?^a G}\ ?^a,$$

where the rule $!^a$ has the side condition that $a \preceq a_i$ for all i. Moreover, for all indices $a \in U$, we add the usual rules of weakening and contraction to $?^a$.

We can enhance the expressiveness of SELL with the subexponential quantifiers $\Cap$ and $\Cup$[§] Nigam et al. (2017) given by the rules (omitting the subexponential signature)

$$\frac{\vdash \Delta, G[l_e/l_x]}{\vdash \Delta, \Cap l_x : a.G}\ \Cap \qquad \frac{\vdash \Delta, G[l/l_x]}{\vdash \Delta, \Cup l_x : a.G}\ \Cup,$$

where l_e is fresh. Intuitively, subexponential variables play a similar role as eigenvariables. The generic variable $l_x : a$ represents any subexponential, constant or variable in the ideal of a. Hence l_x can be substituted by any subexponential l of type b (i.e., $l : b$) if $b \preceq a$. We call the resulting system $\text{SELL}^{\Cap}$.

$\text{SELL}^{\Cap}$ admits a cut-free, complete focused proof system (Figure 4). The sequent notation is close to the one for LLF and differs only on the treatment of contexts. $\text{SELL}^{\Cap}$ makes use of indexed contexts $\mathcal{K}$ that maps a subexponential index to multi-set of formulas, e.g., if s is a subexponential index, then $\mathcal{K}[s]$ is a multi-set of formulas, where intuitively they are all marked with $?^s$. That is, $\mathcal{K}[s] = \{F_1, \ldots, F_n\}$ should be interpreted as the multi-set of formulas $?^s F_1, \ldots, ?^s F_n$. We also make use of the operations on contexts depicted in Figure 5. Most of the operations are straightforward. For instance, $(\mathcal{K}_1 \otimes \mathcal{K}_2)[s]$, used to specify the tensor introduction rule ($\otimes$), is defined as follows: when s is a bounded subexponential index, $(\mathcal{K}_1 \otimes \mathcal{K}_2)[s]$ is obtained by multi-set union of $\mathcal{K}_1[s]$ and $\mathcal{K}_2[s]$; when s is an unbounded subexponential index,

---

[‡] We note that intuitionistic and classical SELL are equally expressive, as shown in Chaudhuri (2010). Hence, although we will introduce here the classical version of SELL, we could also present SELL as an extension of ILL.

[§] $\Cap$ can be read as “for all locations” while $\Cup$ is meant to be ‘there exists a location.’

Negative rules

$$\frac{\vdash \mathcal{K} : \Delta \Uparrow L}{\vdash \mathcal{K} : \Delta \Uparrow \bot, L}\,[\bot] \qquad \frac{\vdash \mathcal{K} : \Delta \Uparrow F, G, L}{\vdash \mathcal{K} : \Delta \Uparrow F \parr G, L}\,[\parr] \qquad \frac{\vdash \mathcal{K} +_l F : \Delta \Uparrow L}{\vdash \mathcal{K} : \Delta \Uparrow ?^l F, L}\,[?^l]$$

$$\frac{}{\vdash \mathcal{K} : \Delta \Uparrow \top, L}\,[\top] \qquad \frac{\vdash \mathcal{K} : \Delta \Uparrow F, L \quad \vdash \mathcal{K} : \Delta \Uparrow G, L}{\vdash \mathcal{K} : \Delta \Uparrow F \,\&\, G, L}\,[\&]$$

$$\frac{\vdash \mathcal{K} : \Delta \Uparrow F[c/x], L}{\vdash \mathcal{K} : \Delta \Uparrow \forall x.F, L}\,[\forall] \qquad \frac{\vdash \mathcal{K} : \Delta \Uparrow F[l_e/l_x], L}{\vdash \mathcal{K} : \Delta \Uparrow \Cap l_x : a.F, L}\,[\Cap_R]$$

Positive rules

$$\frac{}{\vdash \mathcal{K} : \cdot \Downarrow 1}\,[1] \text{ given } \mathcal{K}[\mathcal{I} \setminus \mathcal{U}] = \emptyset \qquad \frac{\vdash \mathcal{K}_1 : \Delta \Downarrow F \quad \vdash \mathcal{K}_2 : \Delta \Downarrow G}{\vdash \mathcal{K}_1 \otimes \mathcal{K}_2 : \Delta, \Delta \Downarrow F \otimes G}\,[\otimes] \text{ given } (\mathcal{K}_1 = \mathcal{K}_2)|_{\mathcal{U}}$$

$$\frac{\vdash \mathcal{K} : \Delta \Downarrow F_i}{\vdash \mathcal{K} : \Delta \Downarrow F_1 \oplus F_2}\,[\oplus_i] \qquad \frac{\vdash \mathcal{K} : \Delta \Downarrow F[t/x]}{\vdash \mathcal{K} : \Delta \Downarrow \exists x.F}\,[\exists] \qquad \frac{\vdash \mathcal{K} : \Delta \Downarrow G[l/l_x]}{\vdash \mathcal{K} : \Delta \Downarrow \Cup l_x : a.G}\,[\Cup_L]$$

$$\frac{\vdash \mathcal{K} \leq_l : \cdot \Uparrow F}{\vdash \mathcal{K} : \cdot \Downarrow !^l F}\,[!^l] \text{ given } \mathcal{K}[\{x \mid l \not\preceq x \wedge x \notin \mathcal{U}\}] = \emptyset$$

Initial, Reaction and Decision Rules

$$\frac{}{\vdash \mathcal{K} : \Delta \Downarrow p(\vec{t})^\perp}\,[I] \text{ given } p(\vec{t}) \in (\Delta \cup \mathcal{K}[\mathcal{I}]) \text{ and } (\Delta \cup \mathcal{K}[\mathcal{I} \setminus \mathcal{U}]) \subseteq \{p(\vec{t})\}$$

$$\frac{\vdash \mathcal{K} +_l P : \Delta \Downarrow P}{\vdash \mathcal{K} +_l P : \Delta \Uparrow \cdot}\,[D_l], \text{ given } l \in \mathcal{U} \qquad \frac{\vdash \mathcal{K} : \Delta \Downarrow P}{\vdash \mathcal{K} +_l P : \Delta \Uparrow \cdot}\,[D_l], \text{ given } l \notin \mathcal{U}$$

$$\frac{\vdash \mathcal{K} : \Delta \Downarrow P}{\vdash \mathcal{K} : \Delta, P \Uparrow \cdot}\,[D_1] \qquad \frac{\vdash \mathcal{K} : \Delta, P_a \Uparrow L}{\vdash \mathcal{K} : \Delta \Uparrow L, P_a}\,[R \Uparrow] \qquad \frac{\vdash \mathcal{K} : \Delta \Uparrow N}{\vdash \mathcal{K} : \Delta \Downarrow N}\,[R \Downarrow]$$

Fig. 4. Focused linear logic system with (quantified) subexponentials. Here, L is a list of formulas, Δ is a multi-set of formulas, P is not an atom, P_a is positive or an atom and N is negative.

then it is $K_1[s]$.[¶] On the other side, for the promotion rule, we use the operation $K \leqslant_l$ that restricts the indexed context K to the formulas marked with a subexponentials greater than l. Hence, $K \leqslant_l [s] = K[s]$ if $l \preceq s$ and $K \leqslant_l [s] = \varnothing$ otherwise.

By using different prefixes, SELL$^\Cap$ is an adequate framework for the specification of richer systems where subexponentials are used to mark different modalities/states. For instance, subexponentials can be used to represent contexts of proof systems (Nigam et al. 2011); to specify systems with temporal, epistemic and spatial modalities (Nigam et al. 2013, 2017; Olarte et al. 2015) and soft-constraints or preferences (Pimentel et al. 2014); to specify Bigraphs (Chaudhuri and Reis 2015); and to specify and verify biological (Olarte et al. 2016) and multimedia interacting systems (Arias et al. 2015).

LL allows for the specification of two kinds of context maintenance: both weakening and contraction are available (unbounded context) or neither is available (linear context). That is, when we encode (linear) judgements in HyLL belonging to different worlds, the resulting

---

[¶] As specified by the side-condition of the $\otimes$ rule in Figure 4, it must be the case that that $K_1[s] = K_2[s]$ when s is unbounded.

- $(\mathcal{K}_1 \otimes \mathcal{K}_2)[i] = \begin{cases} \mathcal{K}_1[i] \cup \mathcal{K}_2[i] & \text{if } i \notin \mathcal{U} \\ \mathcal{K}_1[i] & \text{if } i \in \mathcal{U} \end{cases}$
- $\mathcal{K}[\mathcal{S}] = \bigcup\{\mathcal{K}[i] \mid i \in \mathcal{S}\}$
- $(\mathcal{K} +_l A)[i] = \begin{cases} \mathcal{K}[i] \cup \{A\} & \text{if } i = l \\ \mathcal{K}[i] & \text{otherwise} \end{cases}$
- $\mathcal{K} \leq_i [l] = \begin{cases} \mathcal{K}[l] & \text{if } i \preceq l \\ \emptyset & \text{if } i \npreceq l \end{cases}$
- $(\mathcal{K}_1 = \mathcal{K}_2)\mid_{\mathcal{S}}$ is true if and only if $(\mathcal{K}_1[j] = \mathcal{K}_2[j])$ for all $j \in \mathcal{S}$.

Fig. 5. Specification of operations on contexts.

$$\begin{array}{lcl}
\otimes\, R & : & \exists C, C'.\, \Cup\, w : \infty.(!^{w}\lceil (C \otimes C')@w\rceil^{\perp} \;\otimes\; ?^{w}\lceil C@w\rceil \;\otimes\; ?^{w}\lceil C'@w\rceil) \\
\mathtt{at}\ R & : & \exists A.\, \Cup\, u : \infty, w : \infty.(!^{w}\lceil (A\ \mathtt{at}\ u)@w\rceil^{\perp} \;\otimes\; ?^{u}\lceil A@u\rceil) \\
\mathtt{at}\ L & : & \exists A.\, \Cup\, u : \infty, w : \infty.(!^{w}\lfloor (A\ \mathtt{at}\ u)@w\rfloor^{\perp} \;\otimes\; ?^{u}\lfloor A@u\rfloor) \\
\downarrow R & : & \exists A.\, \Cup\, u : \infty, w : \infty.(!^{w}\lceil \downarrow u.A@w\rceil^{\perp} \;\otimes\; ?^{w}\lceil (A\ w)@w\rceil) \\
\downarrow L & : & \exists A.\, \Cup\, u : \infty, w : \infty.(!^{w}\lfloor \downarrow u.A@w\rfloor^{\perp} \;\otimes\; ?^{w}\lfloor (A\ w)@w\rfloor) \\
\forall R(F) & : & \exists A, \Cup w : \infty.(!^{w}\lceil \forall x.B@w\rceil^{\perp} \otimes \forall x.?^{w}\lceil (B\ x)@w\rceil) \\
\forall R(W) & : & \exists A, \Cup w : \infty.(!^{w}\lceil \forall v.A@w\rceil^{\perp} \otimes \Cap v : \infty.?^{w}\lceil (A\ v)@w\rceil) \\
!\, L & : & \exists C.\, \Cup\, w : \infty.(!^{w}\lfloor !C@w\rfloor^{\perp} \otimes ?^{\mathtt{c}_w}\llfloor C@w\rrfloor) \\
!\, R & : & \exists C.\, \Cup\, w : \infty.(!^{w}\lceil !C@w\rceil^{\perp} \otimes !^{\mathtt{copy}}?^{c_w}\lceil C@w\rceil) \\
\mathit{copy} & : & \exists C.\, \Cup\, w : \infty.(!^{c_w}\llfloor C@w\rrfloor^{\perp} \otimes ?^{w}\lfloor C@w\rfloor)
\end{array}$$

Fig. 6. HyLL rules into SELL$^{\Cap}$. (Definition 3.2).

meta-level atomic formulas will be stored in the same (linear) LL context. The same happens with unbounded HyLL judgements and the unbounded LL context.

Encoding HyLL into SELL$^{\Cap}$ allows for a better understanding of worlds in HyLL. More precisely, we use subexponentials to represent worlds, where each world w has its own linear and unbounded contexts, represented as w and c_w, respectively. Hence, a HyLL judgement of the shape $F@w$ in the (left) linear context is encoded as the SELL$^{\Cap}$ formula $?^{w}\lfloor F@w\rfloor$. That is, HyLL judgements that hold at world w are stored at the w linear context of SELL$^{\Cap}$. A judgement of the form $G@w$ in the unbounded HyLL context is encoded as the SELL$^{\Cap}$ formula $?^{c_w}\llfloor G@w\rrfloor$. Thus, the encoding of $G@w$ is stored in the unbounded subexponential context c_w.

The next definition introduces the encoding of HyLL inference rules into SELL$^{\Cap}$. Observe that, surprisingly, the subexponential structure needed is *flat* on worlds, hence not reflecting their monoidal structure. This is explained by the fact that worlds in HyLL do not control the context on rules as the promotion rule in SELL does.

Definition 3.2 (HyLL rules into SELL$^{\Cap}$). Let $\mathtt{w}, \mathtt{d}, \mathtt{h}, \lceil\cdot\rceil, \lfloor\cdot\rfloor, \llfloor\cdot\rrfloor, A, B, C$ be as in Definition 3.1 and o be the type for SELL$^{\Cap}$ formulas. Given a HyLL constraint domain W, consider a subexponential signature $\Sigma = \langle I, \preceq, U\rangle$ such that $U = \{\mathtt{c}, \mathtt{copy}, \infty\} \cup \{\mathtt{c}_w \mid w \in \mathcal{W}\}$, $I = \mathcal{W} \cup U$. For any $w \in \mathcal{W}$ we have $w \preceq \infty$, $\mathtt{copy} \preceq c_w \preceq \infty$, $\mathtt{copy} \preceq \mathtt{c}$ and, for any other $u, w \in I$, $u \npreceq w$. The encoding of HyLL inference rules into SELL$^{\Cap}$ is depicted in Figure 6 (we omit the encodings of the other connectives, that follow similarly).

Let us give some intuition on the above defined subexponential structure. The unbounded subexponential $\mathtt{c}$ will be used to store the clauses defining the encoding of the rules (see Theorem 3.2). The unbounded subexponential $\mathtt{copy}$ is the least of all the unbounded subexponentials. It is a *dummy* subexponential,[‖] useful to correctly specify $!_R$: when $!^{\mathtt{copy}}$ is introduced, only formulas stored in the unbounded subexponentials can be present (i.e., the theory in $\mathtt{c}$ and the atoms of the form $\lfloor\!\lfloor \cdot \rfloor\!\rfloor$, stored in c_w). Moreover, all the linear locations w (not related to $\mathtt{copy}$) must be empty. This reflects the fact that the HyLL linear context must be empty when $!$ is introduced. Note also that $w : \infty$ represents *any subexponential* in the ideal of ∞ (note that ∞ is also a dummy subexponential). This means that, in the formula $\mathbb{U} w : \infty . F$, the subexponential variable w could be substituted, in principle, by *any* element of $\mathcal{W} \cup \{\mathtt{c}_w \mid w \in \mathcal{W}\}$. That is, the proposed subexponential signature correctly specifies the role of worlds in HyLL, as shown next.

Theorem 3.2 (Adequacy). Let Υ be the set of formulas resulting from the encoding in Definition 3.2. The sequent $\Gamma ; \Delta \vdash F@w$ is provable in HyLL iff the sequent

$$\mathtt{c} : \{\Upsilon\}, \mathtt{c}_{w_i} : \lfloor\!\lfloor \Gamma \rfloor\!\rfloor, w_i : \lfloor \Delta \rfloor, w : \lceil F@w \rceil ; \cdot \Uparrow,$$

is provable in $\mathrm{SELL}^{\Cap}$.[††] Moreover, the adequacy of the encodings is on the *level of derivations*.

Proof. Again, we will consider the rule $\mathtt{at}_L$, as the other cases are similar. If we decide to focus on the $\mathrm{SELL}^{\Cap}$ formula corresponding to the encoding of $\mathtt{at}_L$ (stored in $?^{\mathtt{c}}\Upsilon$), we obtain

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{w : \lfloor (A\ \mathtt{at}\ u)@w \rfloor ; \cdot \Uparrow \lfloor (A\ \mathtt{at}\ u)@w \rfloor^{\perp}}\ D_l, I}{\mathtt{c} : \{\Upsilon\}, \mathtt{c}_{w_i} : \lfloor\!\lfloor \Gamma \rfloor\!\rfloor, w : \lfloor (A\ \mathtt{at}\ u)@w \rfloor ; \cdot \Downarrow !^w \lfloor (A\ \mathtt{at}\ u)@w \rfloor^{\perp}}\ !^w \qquad \dfrac{\mathtt{c} : \{\Upsilon\}, \mathtt{c}_{w_i} : \lfloor\!\lfloor \Gamma \rfloor\!\rfloor, w_i : \lfloor \Delta \rfloor, v : \lceil C@v \rceil, u : \lfloor A@u \rfloor ; \cdot \Uparrow \cdot}{\mathtt{c} : \{\Upsilon\}, \mathtt{c}_{w_i} : \lfloor\!\lfloor \Gamma \rfloor\!\rfloor, w_i : \lfloor \Delta \rfloor, v : \lceil C@v \rceil ; \cdot \Downarrow ?^u \lfloor A@u \rfloor}\ R\Uparrow, ?^u}{\mathtt{c} : \{\Upsilon\}, \mathtt{c}_{w_i} : \lfloor\!\lfloor \Gamma \rfloor\!\rfloor, w_i : \lfloor \Delta \rfloor, w : \lfloor (A\ \mathtt{at}\ u)@w \rfloor, v : \lceil C@v \rceil ; \cdot \Downarrow !^w \lfloor (A\ \mathtt{at}\ u)@w \rfloor^{\perp} \otimes ?^u \lfloor A@u \rfloor}\ \otimes}{\mathtt{c} : \{\Upsilon\}, \mathtt{c}_{w_i} : \lfloor\!\lfloor \Gamma \rfloor\!\rfloor, w_i : \lfloor \Delta \rfloor, w : \lfloor (A\ \mathtt{at}\ u)@w \rfloor, v : \lceil C@v \rceil ; \cdot \Downarrow \exists C, \mathbb{U} u, w.(!^w \lfloor (C\ \mathtt{at}\ u)@w \rfloor^{\perp} ?^u \otimes \lfloor C@u \rfloor)}\ \exists, \mathbb{U}}{\mathtt{c} : \{\Upsilon\}, \mathtt{c}_{w_i} : \lfloor\!\lfloor \Gamma \rfloor\!\rfloor, w_i : \lfloor \Delta \rfloor, w : \lfloor (A\ \mathtt{at}\ u)@w \rfloor, v : \lceil C@v \rceil ; \cdot \Uparrow \cdot}\ D_l$$

Observe that, in a (focused) derivation proving $!^w F$, the only contexts that can be present are w and the ∞ due to the promotion rule and the ordering in Σ. Since the encoding does not store any formula into the context ∞, the formula $!^w F$ must necessarily be proved from the formulas stored in w. Thus, unlike the LL derivation in the proof of Theorem 3.1, the context $\mathtt{c}$ is weakened in the left-hand side derivation since $\mathtt{c} \not\preceq w$. In the end, $\lfloor (A\ \mathtt{at}\ u)@w \rfloor$, initially stored in the location w, is substituted by $\lfloor A@u \rfloor$ in the location u, in one focused step. □

Information Confinement. A brief final comment on the expressiveness of worlds in HyLL. One of the features needed for specifying spatial modalities is information confinement: a space (or world) can be inconsistent and this does not imply the inconsistency of the whole system. It turns out that information confinement can be specified in SELL (Nigam et al. 2017) but not in HyLL. More precisely, since the formulas $!^w ?^w \mathbf{0} \multimap \mathbf{0}$ and $!^w ?^w \mathbf{0} \multimap !^v ?^v \mathbf{0}$ are *not* provable in SELL, it is possible to specify systems where inconsistency is local to a given space and does not propagate to the other locations.

[‖] Subexponentials are often called dummy when they are not inhabited.

[††] Clarifying some notation: if $\Delta = \{F_1@w_1, \ldots, F_n@w_n\}$, then $?^{w_i}\lfloor \Delta \rfloor = ?^{w_1}\lfloor F_1@w_1 \rfloor, \ldots, ?^{w_n}\lfloor F_n@w_n \rfloor$. Observe that, in the negative phase, such formulas will be stored at their respective contexts, that will be represented by $w_i : \lfloor \Delta \rfloor$. Similarly for $\lfloor\!\lfloor \cdot \rfloor\!\rfloor$.

In HyLL, however, it is not possible to confine inconsistency: the HyLL rule

$$\frac{}{\Gamma;\Delta,\mathbf{0}@u \vdash F@w}\;\mathbf{0}L,$$

shows that *any* formula F in *any* world w is derivable from $\mathbf{0}$ appearing in *any* world u. Observe that, even if we exchange the rule $\mathbf{0}L$ for a weaker version

$$\frac{}{\Gamma;\Delta,\mathbf{0}@w \vdash F@w}\;0'_L,$$

the rule $\mathbf{0}L$ would still be admissible

$$\dfrac{\dfrac{}{\Gamma;\mathbf{0}@w \vdash (\mathbf{0}\ \mathtt{at}\ v)@w}\;0'_L \qquad \dfrac{\dfrac{}{\Gamma;\Delta,\mathbf{0}@v \vdash F@v}\;0'_L}{\Gamma;\Delta,(\mathbf{0}\ \mathtt{at}\ v)@w \vdash F@v}\;\mathtt{at}_L}{\Gamma;\Delta,\mathbf{0}@w \vdash F@v}\;cut.$$

4. μMALL and μHyMALL

In the encodings of object systems that operate on inductive structures, such as finite automata, it will be necessary to enrich our representational logic with some mechanism for reasoning about such structures. We will use the μMALL (Baelde 2012) extension that enriches MALL with least (μ) and greatest (ν) fixed points. These fixed points are written in the form $\mu B\vec{t}$ and $\nu B\vec{t}$ where B, called the *body*, is a function of arity $|\vec{t}\,| + 1$; in effect, μB (or νB) serves the role of a *defined predicate* of arity $|\vec{t}\,|$. Since these are fixed points, we further allow for a seamless change between $\mu B\vec{t}$ and $B(\mu B)\vec{t}$—and likewise from $\nu B\vec{t}$ to $B(\nu B)\vec{t}$—which is usually called *unfolding* the fixed point. To obtain the full expressive power of fixed points, it is also essential for the logic to have a notion of intensional equality between terms that obeys the equational theory of the λ-calculus; that is, two terms s and t are considered equal, written $s = t$, if they are related by $\alpha\beta\eta$-conversion (Baelde 2012).

The final ingredient in μMALL is the ability to quantify over the complete set of unifiers (CSU) of two terms s and t that contain free eigenvariables; this set, written $csu(s,t)$, is the smallest set of unifiers of s and t such that every other unifier of s and t is an instance of some unifier in this set. For arbitrary λ-terms s and t, this set can be infinite. However, for well-behaved fragments, such as the first-order or the Lλ fragment (Miller 1992), the CSU is no larger than a singleton. Since these are all standard concepts, we refer the reader to Baelde (2012) for further details.

The proof system for μMALL is built using sequents of the form $\Sigma;\vdash \Delta$, where Σ is a context of typed *eigenvariables*, and Δ is a multi-set of μMALL formulas. As μMALL is an extension of the standard MALL proof system, we elide their standard rules here. The remaining rules cover equality, its formal negation ($\neq$), and the fixed points μ and ν. The rules for the former are as follows.

$$\frac{}{\Sigma;\vdash t = t}\;= \qquad\qquad \frac{\{(\Sigma;\vdash \Delta)\theta \;:\; \theta \in csu(s,t)\}}{\Sigma;\vdash \Delta, s \neq t}\;\neq.$$

The rule for inequality requires a bit of explanation. There is one premise for each $\theta \in csu(s,t)$. The *instance* $(\Sigma;\vdash \Delta)\theta$ of the sequent $\Sigma;\vdash \Delta$ is defined as usual: its eigenvariables are the eigenvariables in the set of terms $\{u\theta \,:\, u \in \Sigma\}$, and for each formula $F \in \Delta$ there is the formula $F\theta$ in $\Delta\theta$.

For the fixed points, there is a version of the identity rule that relates the least and greatest fixed points, an unfold rule for least fixed points, and a coinduction rule for greatest fixed points:

$$\frac{}{\Sigma; \vdash \mu B\vec{t}, \nu \bar{B}\vec{t}}\ dInit \qquad \frac{\Sigma; \vdash \Delta, B(\mu B)\vec{t}}{\Sigma; \vdash \Delta, \mu B\vec{t}}\ \mu \qquad \frac{\vec{x}; \vdash (S\vec{x}\,)^{\perp}, BS\vec{x} \quad \Sigma; \vdash \Delta, S\vec{t}}{\Sigma; \vdash \Delta, \nu B\vec{t}}\ \nu \, .$$

In the defined identity rule *dInit*, the notation $\bar{B}$ stands for $\lambda p.\, \lambda\vec{x}.\, (B\ p^{\perp}\ \vec{x}\,)^{\perp}$. In the coinduction rule (ν), the predicate S is an *invariant*. The first premise of the rule shows that it is indeed an invariant of B, while the second premise replaces the greatest fixed point νB with the invariant. Observe that if we use $B(\nu B)$ itself for the invariant S, then we obtain

$$\frac{\dfrac{\begin{array}{c}\vdots\\ \vec{x}; \vdash \bar{B}(\mu\bar{B})\vec{x}, B(B(\nu B))\vec{x}\end{array}}{\vec{x}; \vdash (B(\nu B)\vec{x}\,)^{\perp}, B(B(\nu B))\vec{x}} \quad \Sigma; \vdash \Delta, B(\nu B)\vec{t}}{\Sigma; \vdash \Delta, \nu B\vec{t}}\ \nu \, .$$

The left branch is a proof of identity where eventually the defined identity rule *dInit* is used to relate $\mu\bar{B}\vec{x}$ and $\nu B\vec{x}$. This branch will therefore always be derivable. Hence, we see that the unfold rule for ν is derivable in terms of the coinduction rule, and therefore does not need to be given explicitly. The meta-theory of μMALL, including the important cut-elimination theorem, is pretty standard and exhaustively covered in Baelde (2012).

Along the same lines as μMALL, we can extend HyMALL(the multiplicative/additive fragment of HyLL in Figure 2) to μHyMALL, by adding equality and least and greatest fixed points. In fact, for fixed point predicates built using μ and ν, we will allow the arguments to contain worlds as well; likewise, we will allow for equality to hold between worlds. However, we retain the restriction from HyMALL that all undefined predicates contain no world arguments.[‡‡] Like with μMALL sequents, μHyMALL sequents will have an explicit context of eigenvariables, so they will be of the form $\Sigma; \Delta \vdash F \ @\ w$, where Δ is as before. Since we are limiting our attention to μHyMALL, we dispense with the unbounded context Γ, which can be added to yield μHyLL. Most of the rules from Figure 2 can be directly adapted with this additional eigenvariable context. The remaining rules are given in Figure 7.

It may be worthwhile to consider if the μHyMALL rules can be encoded in μMALL by means of an extension of Definition 3.1. Indeed, we can simply extend the rules of Figure 3 with new cases for equalities and fixed points. The extension is almost entirely trivial and elided here except for the following sketch: both [μ*Init*] and [ν*Init*] will be captured by means of *dInit*; [$= R$] by means of $=$; [$= L$] by means of $\neq$; [$\mu\ L$] and [$\nu\ R$] by means of ν; and [$\mu\ R$] and [$\nu\ L$] by means of μ.

5. Computation tree logic (CTL) in linear logic

HyLL is expressive enough to encode some forms of modal operators, thus allowing for the specification of properties of transition systems. As shown in de Maria et al. (2014), it is possible to encode CTL temporal operators into HyLL considering existential (E) and bounded universal

---

[‡‡] This restriction can be lifted from HyMALL without any difficulty.

Defined identity rules

$$\frac{}{\Sigma;\, \mu B\vec{t} @ w \vdash \mu B\vec{t} @ w}\,[\mu \mathit{Init}] \qquad \frac{}{\Sigma;\, \nu B\vec{t} @ w \vdash \nu B\vec{t} @ w}\,[\nu \mathit{Init}]$$

Equality rules

$$\frac{}{\Sigma;\, \cdot \vdash t = t @ w}\,[= R] \qquad \frac{\Big\{(\Sigma; \Delta \vdash C @ w)\theta \;:\; \theta \in csu(s,t)\Big\}}{\Sigma;\, \Delta, s = t @ u \vdash C @ w}\,[= L]$$

Least fixed point rules

$$\frac{\Sigma; \Delta \vdash B(\mu B)\vec{t} @ w}{\Sigma; \Delta \vdash \mu B\vec{t} @ w}\,[\mu\ R] \qquad \frac{\vec{x}; BS\vec{x} @ u \vdash S\vec{x} @ u \quad \Sigma; \Delta, S\vec{t} @ u \vdash C @ w}{\Sigma; \Delta, \mu B\vec{t} @ u \vdash C @ w}\,[\mu\ L]$$

Greatest fixed point rules

$$\frac{\vec{x}; S\vec{x} @ w \vdash BS\vec{x} @ w \quad \Sigma; \Delta \vdash S\vec{t} @ w}{\Sigma; \Delta \vdash \nu B\vec{t} @ w}\,[\nu\ R] \qquad \frac{\Sigma; \Delta, B(\nu B)\vec{t} @ u \vdash C @ w}{\Sigma; \Delta, \nu B\vec{t} @ u \vdash C @ w}\,[\nu\ L]$$

Fig. 7. Rules specific to μHyMALL. The rules for the HyMALL connectives can be directly adapted from those in Figure 2.

(A) path quantifiers. We show in this section the limitation of such encodings and how to fully capture E and A CTL quantifiers in both propositional μMALL and first-order μHyMALL. In both cases, we follow the standard interpretation of CTL quantifiers as fixed points.

The first encoding relies on the behaviour of the LL connectives to control the use of transition rules during a proof of a CTL formula. More precisely, states in the transition system are represented as atoms (in the linear context) that are consumed and produced by the encoding of transitions. The second encoding uses HyLL's words in order to define states and quantifiers on words to specify path quantifiers. Hence, the encoding resembles the semantics of CTL.

Let us start by recalling the syntax of CTL.

Definition 5.1 (CTL connectives and path quantifiers). Given a set of atomic propositions $\mathcal{P}$, formulas in CTL are given by the following grammar:

$$F \ ::= \ p \mid \neg F \mid F \wedge F \mid F \vee F \mid \mathbf{Q}\mathsf{X}F \mid \mathbf{Q}\mathsf{F}F \mid \mathbf{Q}\mathsf{G}F \mid \mathbf{Q}[F\mathsf{U}F] \qquad p \in \mathcal{P}, \mathbf{Q} \in \{\mathsf{A}, \mathsf{E}\}.$$

The temporal connectives are: X (Next) meaning 'at the next state;' F (Future) meaning 'in some future;' G (Globally) meaning 'in all futures;' and, $F\mathsf{U}G$ (F until G) meaning 'from now, F will be true in every steps until some future point (possibly including now) where G holds.' Temporal connectives must be preceded by a path quantifier: E (Exists) meaning 'for some path'or A (All) meaning 'for all paths.' The usual dualities apply (e.g., $\neg\mathsf{EX}F = \mathsf{AX}\neg F$, $\neg\mathsf{AG}F = \mathsf{EF}\neg F$) and negation is involutive i.e., it can be restricted to atoms.

Transition Systems. Let $\mathcal{P} = \{p_1, ..., p_n\}$ be a set of atomic propositions. A Kripke structure over $\mathcal{P}$ is a tuple $\mathcal{K} = \langle S, I, R, L\rangle$ where S is a finite set of states, $I \subseteq S$ is the set of initial states, $R \subseteq S \times S$ is a transition relation and $L : S \to 2^{\mathcal{P}}$ is a labelling. We assume that given two different states $\mathtt{s}, \mathtt{s}'$, $L(\mathtt{s}) \neq L(\mathtt{s}')$. Note that this is not a loss of generality, since we can always extend $\mathcal{P}$ with atomic propositions to uniquely identify each state. We shall write $\mathtt{s} \longrightarrow \mathtt{s}'$ when $(\mathtt{s}, \mathtt{s}') \in R$. Observe that, in CTL, R must be serial, *i.e.*, every state has a successor. Finally, we write $\mathtt{s} \models^{K}_{CTL} F$ when F holds at state $\mathtt{s}$ with the standard meaning (see, e.g., Clarke and Emerson (1981)). For instance, $\mathtt{s} \models^{K}_{CTL} \mathsf{EG}F$ iff there exists a path $\pi = \langle \mathtt{s}_1 \cdot \mathtt{s}_2 \cdot \mathtt{s}_3 \cdot \ldots \rangle$ starting at $\mathtt{s}$ (*i.e.* $\mathtt{s} = \mathtt{s}_1$) such that for all $i \geqslant 1$, $\mathtt{s}_i \models^{K}_{CTL} F$.

5.1. *Transition systems and HyLL*

In order to specify reachability properties in transition systems, some modal connectives can be defined in HyLL (Despeyroux and Chaudhuri 2014):

$$\Box A \overset{\text{def}}{=} \downarrow u.\, \forall w.\, (A \;\mathtt{at}\; u.w) \qquad \Diamond A \overset{\text{def}}{=} \downarrow u.\, \exists w.\, (A \;\mathtt{at}\; u.w)$$
$$\delta_v\, A \overset{\text{def}}{=} \downarrow u.\, (A \;\mathtt{at}\; u.v).$$

$\Box A$ (resp. $\Diamond A$) represents all (resp. some) state(s) satisfying A and reachable in some path from now. The connective δ represents a form of delay: $\delta_v\, A$ stands for an *intermediate state* in a transition to A. Informally, it can be thought to be 'v before A .'

We may use such modal operators in order to encode some features of transition systems as HyLL formulas. To each $p \in \mathcal{P}$, we associate two HyLL atomic formulas: p and $p^\perp$ (abusing the notation), where by $p^\perp$ we denote the atomic HyLL proposition interpreting the CTL formula $\neg p$. Then states and transitions can be encoded as follows:

$$[\![\mathtt{s}]\!]_{\mathcal{K}} = \bigotimes_{p \in \mathcal{P}} v(\mathtt{s}, p) \qquad [\![\mathtt{s} \longrightarrow \mathtt{s}']\!]_{\mathcal{K}} = \forall w.\, \big(([\![\mathtt{s}]\!]_{\mathcal{K}} \;\mathtt{at}\; w) \multimap \delta_1([\![\mathtt{s}']\!]_{\mathcal{K}}) \;\mathtt{at}\; w\big)$$

where $v(\mathtt{s}, p) = p$ if $p \in L(s)$ and $v(\mathtt{s}, p) = p^\perp$ otherwise. Given a transition relation $R = \{r_1, ..., r_m\}$, we use $[\![R]\!]_{\mathcal{K}} @ w$ to denote the set $\{[\![r_1]\!]_{\mathcal{K}} @ w, \cdots, [\![r_m]\!]_{\mathcal{K}} @ w\}$.

We can encode in HyLL a restricted fragment of CTL, namely, formulas built using only the temporal connectives EX, EF :

$$\begin{array}{llll} [\![p]\!]_{\mathcal{K}} & = p \otimes \top & [\![\neg p]\!]_{\mathcal{K}} & = p^\perp \otimes \top \\ [\![F \wedge G]\!]_{\mathcal{K}} & = \mathbf{d}^+([\![F]\!]_{\mathcal{K}} \,\&\, [\![G]\!]_{\mathcal{K}}) & [\![F \vee G]\!]_{\mathcal{K}} & = \mathbf{d}^-([\![F]\!]_{\mathcal{K}}) \oplus \mathbf{d}^-([\![G]\!]_{\mathcal{K}}), \\ [\![\mathsf{EX}F]\!]_{\mathcal{K}} & = \mathbf{d}^+(\delta_1 [\![F]\!]_{\mathcal{K}}) & [\![\mathsf{EF}F]\!]_{\mathcal{K}} & = \Diamond [\![F]\!]_{\mathcal{K}} \end{array}$$

where $\mathbf{d}^+(F) = F \otimes 1$ and $\mathbf{d}^-(F) = 1 \multimap F$ are positive and negative delays, respectively. Observe that $\mathbf{d}^+(F) \equiv \mathbf{d}^-(F) \equiv F$. Delays are added for adequacy results.

Proposition 5.1 (Adequacy). Let $\mathcal{K} = \langle S, I, R, L \rangle$ be a Kripke structure on a set of atomic propositions $\mathcal{P}$. Let F be a CTL formula built from the CTL fragment $\wedge, \vee, \mathsf{EX}, \mathsf{EF}$. Then, $\mathtt{s} \models^{K}_{CTL} F$ iff $[\![R]\!]_{\mathcal{K}} @ 0; [\![\mathtt{s}]\!]_{\mathcal{K}} @ w \vdash [\![F]\!]_{\mathcal{K}} @ w$ is provable in HyLL.

Proof. We will reason on the focused version of HyLL and we will assume that atoms have positive bias. Assume that $\mathtt{s} \longrightarrow \mathtt{s}'$. If we decide to focus on the encoding of $(\mathtt{s}, \mathtt{s}') \in R$, we necessarily obtain a derivation of the shape

$$\frac{[\![R]\!]_{\mathcal{K}} @ 0; [\![\mathtt{s}']\!]_{\mathcal{K}} @ w.1 \vdash G}{[\![R]\!]_{\mathcal{K}} @ 0; [\![\mathtt{s}]\!]_{\mathcal{K}} @ w \vdash G} \;, \tag{1}$$

where *all* atoms from $[\![\mathtt{s}]\!]_{\mathcal{K}} @ w$ are consumed and the formula $[\![\mathtt{s}']\!]_{\mathcal{K}} @ w.1$ is added into the context. This mimics exactly the transition $\mathtt{s} \longrightarrow \mathtt{s}'$.

The (⇒) side proceeds by induction on the structure of F. For the base case, if $\mathsf{s} \models^{K}_{CTL} p$, it is easy to show that the sequent $[\![R]\!]_{\mathcal{K}}@0; [\![\mathsf{s}]\!]_{\mathcal{K}}@u \vdash (p \otimes \top)@u$ is provable in HyLL (similarly for $\neg p$). If $\mathsf{s} \models^{K}_{CTL} \mathsf{EF}\; F$, then there exists a path $\langle \mathsf{s}_1 \cdot \mathsf{s}_2 \cdot \cdots \rangle$ starting at s s.t. there exists $i \geqslant 1$ s.t. $\mathsf{s}_i \models^{K}_{CTL} F$. By repetitively applying Equation (1), we have a derivation that consumes $[\![\mathsf{s}_1]\!]_{\mathcal{K}}$ to produce $[\![\mathsf{s}_i]\!]_{\mathcal{K}}$ and the result follows by induction. The case for $\mathsf{EX}F$ follows similarly. Finally, the cases for $\wedge$ and $\vee$ follow immediately by induction.

(⇐) We shall show that each focused step corresponds exactly to a 'step' in the deduction of $\mathsf{s} \models^{K}_{CTL} F$. Consider the sequent $[\![R]\!]_{\mathcal{K}}@0; [\![\mathsf{s}]\!]_{\mathcal{K}}@w \vdash [\![F]\!]_{\mathcal{K}}@w$. We have two choices: (i) focus on $[\![\mathsf{s} \longrightarrow \mathsf{s}']\!]_{\mathcal{K}}$ and, from Equation (1), we transform the state s into the state s'; or (ii) focus on the formula on the right. In the first case, we already showed that this action mimics exactly the transition $\mathsf{s} \longrightarrow \mathsf{s}'$. In the second case, the focused formula F must be of the form

$$F \ ::= \ p \otimes \top \mid p^{\perp} \otimes \top \mid 1 \otimes (F \,\&\, F) \mid F \oplus F \mid \downarrow u\,(F \;\mathtt{at}\; u.1) \mid \downarrow u\,(\exists w.F \;\mathtt{at}\; u.w),$$

representing the encoding of atoms, conjunction, disjunction, $\mathsf{EX}F$ and $\mathsf{EF}F$, respectively. In a negative phase, the only connectives we can introduce, if any, are the hybrid ones ($\downarrow$ and $\mathtt{at}$). This is a bureaucratic step allowing us to fix the formulas at the 'current' world as in

$$\frac{\Gamma;\Delta \vdash F[x/w]@y}{\Gamma;\Delta \vdash \downarrow x(F \;\mathtt{at}\; y)@w}\ \mathtt{at}_R, \downarrow_R.$$

Hence, when focusing on F we fall in one of the following cases:

— $F = p \otimes \top$ (or $p^{\perp} \otimes \top$): the context must already have p (or $p^{\perp}$), at the right world, to prove p (or $p^{\perp}$). This corresponds to proving that the state s satisfies (or not) p.

— $F = 1 \otimes (F_1 \,\&\, F_2)$: 1 is proved with empty context and focus is lost in $F_1 \,\&\, F_2$. Hence, after a negative phase, we have a derivation proving F_1 and another proving F_2. This corresponds exactly to the step of proving a conjunction in CTL.

— $F = F_1 \oplus F_2$: chose one of the branches and focus is lost due to the negative delay in the encodings. This corresponds to proving a disjunction in CTL.

— $F = \mathbf{d}^{+}(\delta_1\, F)$: focus is lost obtaining, on the right, F fixed at the world $w + 1$. This mimics the step of proving F in the next time-unit ($\mathsf{EX}F$).

— $F = \exists w.F \;\mathtt{at}\; u.w$: a world w is chosen and focus is lost (due to $\mathtt{at}$). This corresponds in CTL to proving $\mathsf{EF}F$ by showing that there exists a future world $(u + w)$ where F holds.

□

Observe that our encoding cannot be extended to consider formulas of the shape $\mathsf{EG}F$. In fact, the natural choice would be $[\![\mathsf{EG}F]\!]_{\mathcal{K}} = \Box[\![F]\!]_{\mathcal{K}}$, but this encoding would not be adequate. Consider, for instance, a system with a unique state s and a unique (looping) transition $\mathsf{s} \longrightarrow \mathsf{s}$. Assuming that $p \in L(\mathsf{s})$, clearly s satisfies the formula $\mathsf{EG}p$. Now, consider the HyLL sequent $[\![\mathsf{s} \longrightarrow \mathsf{s}]\!]_{\mathcal{K}}@0; [\![\mathsf{s}]\!]_{\mathcal{K}}@w \vdash \Box[\![\mathsf{s}]\!]_{\mathcal{K}}@w$. Introducing the connectives on the right, we obtain

$$\frac{[\![\mathsf{s} \longrightarrow \mathsf{s}]\!]_{\mathcal{K}}@0; [\![\mathsf{s}]\!]_{\mathcal{K}}@w \vdash [\![\mathsf{s}]\!]_{\mathcal{K}}@w.v}{[\![\mathsf{s} \longrightarrow \mathsf{s}]\!]_{\mathcal{K}}@0; [\![\mathsf{s}]\!]_{\mathcal{K}}@w \vdash \Box[\![\mathsf{s}]\!]_{\mathcal{K}}@w}\ \downarrow_R, \forall_R, \mathtt{at}_R,$$

where v is fresh. Then, focusing on the encoding of $\mathsf{s} \longrightarrow \mathsf{s}'$

$$\frac{[\![\mathsf{s} \longrightarrow \mathsf{s}]\!]_{\mathcal{K}}@0; [\![\mathsf{s}]\!]_{\mathcal{K}}@(w+1) \vdash G}{[\![\mathsf{s} \longrightarrow \mathsf{s}]\!]_{\mathcal{K}}@0; [\![\mathsf{s}]\!]_{\mathcal{K}}@w \vdash G}\ \mathit{copy}, \forall_L, \multimap_L.$$

$$
\begin{array}{lcl}
\llbracket \mathsf{AX}F \rrbracket_{\mathcal{K}} & = & \mathop{\&}\limits_{(\mathtt{s},\mathtt{s}')\in R} (\mathtt{neg}(\mathtt{s}) \oplus (\mathtt{pos}(\mathtt{s}) \otimes (\llbracket \mathtt{s}' \rrbracket_{\mathcal{K}} \parr \llbracket F \rrbracket_{\mathcal{K}}))) \\
\llbracket \mathsf{EX}F \rrbracket_{\mathcal{K}} & = & \bigoplus\limits_{(\mathtt{s},\mathtt{s}')\in R} (\mathtt{pos}(\mathtt{s}) \otimes (\llbracket \mathtt{s}' \rrbracket_{\mathcal{K}} \parr \llbracket F \rrbracket_{\mathcal{K}})) \\
\llbracket \mathsf{AF}F \rrbracket_{\mathcal{K}} & = & \mu Y.\, \llbracket F \rrbracket_{\mathcal{K}} \oplus \mathop{\&}\limits_{(\mathtt{s},\mathtt{s}')\in R} (\mathtt{neg}(\mathtt{s}) \oplus (\mathtt{pos}(\mathtt{s}) \otimes (\llbracket \mathtt{s}' \rrbracket_{\mathcal{K}} \parr Y))) \\
\llbracket \mathsf{EF}F \rrbracket_{\mathcal{K}} & = & \mu Y.\, \llbracket F \rrbracket_{\mathcal{K}} \oplus \bigoplus\limits_{(\mathtt{s},\mathtt{s}')\in R} (\mathtt{pos}(\mathtt{s}) \otimes (\llbracket \mathtt{s}' \rrbracket_{\mathcal{K}} \parr Y)) \\
\llbracket \mathsf{AG}F \rrbracket_{\mathcal{K}} & = & \nu Y.\, \llbracket F \rrbracket_{\mathcal{K}} \,\&\, \mathop{\&}\limits_{(\mathtt{s},\mathtt{s}')\in R} (\mathtt{neg}(\mathtt{s}) \oplus (\mathtt{pos}(\mathtt{s}) \otimes (\llbracket \mathtt{s}' \rrbracket_{\mathcal{K}} \parr Y))) \\
\llbracket \mathsf{EG}F \rrbracket_{\mathcal{K}} & = & \nu Y.\, \llbracket F \rrbracket_{\mathcal{K}} \,\&\, \bigoplus\limits_{(\mathtt{s},\mathtt{s}')\in R} (\mathtt{pos}(\mathtt{s}) \otimes (\llbracket \mathtt{s}' \rrbracket_{\mathcal{K}} \parr Y)) \\
\llbracket \mathsf{A}[F \,\mathsf{U}\, G] \rrbracket_{\mathcal{K}} & = & \mu Y. \llbracket G \rrbracket_{\mathcal{K}} \oplus \left(\llbracket F \rrbracket_{\mathcal{K}} \,\&\, \mathop{\&}\limits_{(\mathtt{s},\mathtt{s}')\in R} (\mathtt{neg}(\mathtt{s}) \oplus (\mathtt{pos}(\mathtt{s}) \otimes (\llbracket \mathtt{s}' \rrbracket_{\mathcal{K}} \parr Y))) \right) \\
\llbracket \mathsf{E}[F \,\mathsf{U}\, G] \rrbracket_{\mathcal{K}} & = & \mu Y. \llbracket G \rrbracket_{\mathcal{K}} \oplus \left(\llbracket F \rrbracket_{\mathcal{K}} \,\&\, \bigoplus\limits_{(\mathtt{s},\mathtt{s}')\in R} (\mathtt{pos}(\mathtt{s}) \otimes (\llbracket \mathtt{s}' \rrbracket_{\mathcal{K}} \parr Y)) \right)
\end{array}
$$

Fig. 8. Encoding of CTL into propositional μMALL (see Definition 5.2).

Therefore the left and right worlds in the sequent will never match, and this sequent is not provable. In other words: the resources in the context are enough for proving the property for a (bounded) n but not for all natural numbers. For proving this, one *necessarily* needs (meta-level) induction, i.e., fixed points.

5.2. *Encoding* E *and* A *quantifiers in propositional* μ*MALL*

In order to prove (in CTL) the formula $\mathsf{AF}F$ at state $\mathtt{s}$, we have to check if $\mathtt{s}$ satisfies F. If this is not the case, we have to check whether $\mathsf{AF}F$ holds for all successors of $\mathtt{s}$. Hence, CTL quantifiers are usually characterized as fixed points (see e.g., Burch et al. (1992)).

$$
\begin{array}{lll}
\mathsf{EF}F = \mu Y.F \vee \mathsf{EX}Y & \mathsf{EG}F = \nu Y.F \wedge \mathsf{EX}Y & \mathsf{E}[F \,\mathsf{U}\, G] = \mu Y.G \vee (F \wedge \mathsf{EX}Y) \\
\mathsf{AF}F = \mu Y.F \vee \mathsf{AX}Y & \mathsf{AG}F = \nu Y.F \wedge \mathsf{AX}Y & \mathsf{A}[F \,\mathsf{U}\, G] = \mu Y.G \vee (F \wedge \mathsf{AX}Y).
\end{array}
$$

Definition 5.2 (CTL into propositional μMALL). Let $\mathcal{K} = \langle S, I, R, L \rangle$ be a Kripke structure on a set of atomic propositions $\mathcal{P}$. We define

- $\llbracket \mathtt{s} \rrbracket_{\mathcal{K}} = (\bigotimes\limits_{p \in \mathcal{P}} v(\mathtt{s}, p))^{\perp}$ where $v(\mathtt{s}, p) = p$ if $p \in L(s)$ and $v(\mathtt{s}, p) = p^{\perp}$ otherwise.
- $\mathtt{pos}(s) = \llbracket \mathtt{s} \rrbracket_{\mathcal{K}}^{\perp}$
- $\mathtt{neg}(s) = \bigoplus\limits_{p \in \mathcal{P}} (v(\mathtt{s}, p)^{\perp} \otimes \top)$.

The encodings of **Q**X, **Q**F, **Q** G *and* **Q**U, for **Q** $\in \{\mathsf{A}, \mathsf{E}\}$ are in Figure 8. The encoding of the rest of the formulas is as in the case for HyLL.

The encoding relies on the following principles. Let $r = (\mathtt{s}, \mathtt{s}') \in R$. The formula $\mathtt{pos}(\mathtt{s})$ (resp. $\mathtt{neg}(\mathtt{s})$) tests if r can (resp. cannot) be fired at the current state. If it can be fired, then the current state is transformed into the new state. Hence, the encoding of A (resp. E) test all (resp. at least one) of the fireable rules. This explains the use of $\&$ (resp. $\bigoplus$). Finally, the use of least or greatest fixed points reflects the fixed point characterization of CTL connectives given above.

Remark 5.1. Observe that, in all the clauses in Figure 8, the formula $\mathtt{pos}(\mathtt{s}) \otimes (\llbracket \mathtt{s}' \rrbracket_{\mathcal{K}} \parr B)$, is present. We could have written instead $\llbracket r \rrbracket_{\mathcal{K}} \multimap B$, which reads closer to what we expect:

‘assuming that r is fired, B holds.’ The formulas $(L \multimap R) \multimap B$ and $L \otimes (R \multimap B)$ are not logically equivalent. In fact, the first formula is equivalent to $(L \otimes R^{\perp}) \parr B$ while the second is equivalent to $L \otimes (R^{\perp} \parr B)$. The first is stronger, in the sense that B can choose the branch to move up with (L or R), while the second forces B to stick with R. We chose the second since it describes better the desired behaviour, thus easing the proof of the following adequacy result.

Theorem 5.1 (Adequacy). Let $\mathcal{K} = \langle S, I, R, L \rangle$ be a Kripke structure on a set of atomic propositions $\mathcal{P}$, $\mathtt{s} \in S$ be a state and F be a CTL formula. Then, $\mathtt{s} \models^{\mathcal{K}}_{CTL} F$ iff the sequent $\cdot; \vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, [\![F]\!]_{\mathcal{K}}$ is provable in μMALL.

Proof. As done for HyLL, we will consider the focused version of μMALL and we will assume that atoms have positive bias.
($\Rightarrow$) We proceed by induction on the structure of the formula. The base cases for atomic formulas (p and $\neg p$) are trivial and the cases for $\wedge$ and $\vee$ are easy consequences from the inductive hypothesis.
Cases AX and EX. Note that given two different states $\mathtt{s}$ and $\mathtt{s}'$ (thus $L(\mathtt{s}) \neq L(\mathtt{s}')$)

— the sequents $\vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, \mathtt{pos}(\mathtt{s})$ and $\vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, \mathtt{neg}(\mathtt{s}')$ are both provable.
— the sequents $\vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, \mathtt{neg}(\mathtt{s})$ and $\vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, \mathtt{pos}(\mathtt{s}')$ are both not provable.

This means that, in a context containing the formula $[\![\mathtt{s}]\!]_{\mathcal{K}}$, we can always prove if a given transition rule $r \in R$ is fireable or not.

Consider the case $\mathsf{AX}F$. The derivation necessarily starts with the negative phase

$$\frac{\Sigma; \vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, \mathtt{neg}(\mathtt{s}_1) \oplus (\mathtt{pos}(\mathtt{s}_1) \otimes ([\![\mathtt{s}'_1]\!]_{\mathcal{K}} \parr [\![F]\!]_{\mathcal{K}}) \quad \ldots \quad \Sigma; \vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, \mathtt{neg}(\mathtt{s}_m) \oplus (\mathtt{pos}(\mathtt{s}_m) \otimes ([\![\mathtt{s}'_m]\!]_{\mathcal{K}} \parr [\![F]\!]_{\mathcal{K}})}{\Sigma; \vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, \mathop{\&}_{(\mathtt{s},\mathtt{s}') \in R} \left(\mathtt{neg}(\mathtt{s}) \oplus (\mathtt{pos}(\mathtt{s}) \otimes \left([\![\mathtt{s}']\!]_{\mathcal{K}} \parr [\![F]\!]_{\mathcal{K}} \right) \right)} \; \& \tag{2}$$

Then, for every premise, a positive phase starts, choosing between $\mathtt{neg}(\mathtt{s}_i)$ and $\mathtt{pos}(\mathtt{s}_i)$. In the first case, if the rule is not fireable, the proof ends. In the second case, we have

$$\frac{\Sigma; \vdash [\![\mathtt{s}'_i]\!]_{\mathcal{K}}, [\![F]\!]_{\mathcal{K}}}{\Sigma; \vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, \mathtt{pos}(\mathtt{s}_i) \otimes ([\![\mathtt{s}'_i]\!]_{\mathcal{K}} \parr [\![F]\!]_{\mathcal{K}})} \; \otimes, \parr ,$$

and the positive phase ends. By inductive hypothesis, the sequent $\Sigma; \vdash [\![\mathtt{s}'_i]\!]_{\mathcal{K}}, [\![F]\!]_{\mathcal{K}}$ is provable. The case $\mathsf{EX}F$ is similar.
Cases for the least fixed point operators. If $\mathsf{AF}F$ holds in CTL at state $\mathtt{s}$, then, in all paths starting at $\mathtt{s}$, there is a reachable state $\mathtt{s}'$ such that F holds at $\mathtt{s}'$. Let $\mathtt{s} = \mathtt{s}_1 \longrightarrow \cdots \longrightarrow \mathtt{s}_n = \mathtt{s}'$ be one of such paths and consider the following derivation:

$$\frac{\Sigma; \vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, \mathtt{neg}(\mathtt{s}_1) \oplus (\mathtt{pos}(\mathtt{s}_1) \otimes ([\![\mathtt{s}'_1]\!]_{\mathcal{K}} \parr \mu B) \quad \ldots \quad \Sigma; \vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, \mathtt{neg}(\mathtt{s}_m) \oplus (\mathtt{pos}(\mathtt{s}_m) \otimes ([\![\mathtt{s}'_m]\!]_{\mathcal{K}} \parr \mu B)}{\Sigma; \vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, \mu B} \; \mu, \oplus, \& .$$

The premises correspond to proving whether a transition $r \in R$ is fireable or not. If r is fireable, we observe a derivation of the shape

$$\cfrac{\cfrac{\Sigma;\vdash \llbracket \mathtt{s}'_i \rrbracket_{\mathcal{K}}, \mu B}{\Sigma;\vdash \llbracket \mathtt{s} \rrbracket_{\mathcal{K}}, \mathtt{pos}(\mathtt{s}_i) \otimes (\llbracket \mathtt{s}'_i \rrbracket_{\mathcal{K}} \parr \mu B)}\ \otimes, \parr}{\Sigma;\vdash \llbracket \mathtt{s} \rrbracket_{\mathcal{K}}, \mathtt{neg}(\mathtt{s}_i) \oplus (\mathtt{pos}(\mathtt{s}_i) \otimes (\llbracket \mathtt{s}'_i \rrbracket_{\mathcal{K}} \parr \mu B))}\ \oplus$$

where $\mathtt{s}$ becomes $\mathtt{s}'_i$ and, from that state, μB must be proved. Hence, we can show that $\llbracket \mathtt{s}_n \rrbracket_{\mathcal{K}}$ will be eventually added to the context. By inductive hypothesis, the sequent $\Sigma;\vdash \llbracket \mathtt{s}_n \rrbracket_{\mathcal{K}}, \llbracket F \rrbracket_{\mathcal{K}}$ is provable and hence $\Sigma;\vdash \llbracket \mathtt{s}_n \rrbracket_{\mathcal{K}}, \mu B$ is provable (by unfolding and then choosing $\llbracket F \rrbracket_{\mathcal{K}}$ in the disjunction $\llbracket \mathsf{AF} F \rrbracket_{\mathcal{K}} = \mu Y.\llbracket F \rrbracket_{\mathcal{K}} \oplus \Psi$).

The other cases for least fixed point operators follow similarly.

Cases for the greatest fixed point operators. Consider now the formula $\mathsf{AG}F$. If this formula holds at $\mathtt{s}$, then $\mathtt{s}$ must satisfy F and all reachable states from $\mathtt{s}$ must also satisfy $\mathsf{AG}F$. Let

$$\mathcal{S} = \{s \in S \mid s \models^{K}_{CTL} F \text{ and, for all } s', \text{ if } s \longrightarrow s', \text{ then } s' \in S\},$$

be the greatest set of states containing $\mathtt{s}$. Note that the greatest fixed point in the (CTL) definition of AG computes exactly that set.

Let $\mathcal{S}$ above be the set $\{s_1, ..., s_n\}$ and $I = \llbracket s_1 \rrbracket_{\mathcal{K}}^{\perp} \oplus \cdots \oplus \llbracket s_n \rrbracket_{\mathcal{K}}^{\perp}$. We shall show that, for any $s \in \mathcal{S}$, the sequent $\Sigma;\vdash \llbracket s \rrbracket_{\mathcal{K}}, \llbracket \mathsf{AG}F \rrbracket_{\mathcal{K}}$ is provable using I as inductive invariant.

Once the rule ν is applied, we have to prove two premises:

1. **Premise** $\Sigma;\vdash \llbracket s \rrbracket_{\mathcal{K}}, I$. This sequent is easy by choosing $\llbracket s \rrbracket_{\mathcal{K}}^{\perp}$ from I.
2. **Premise** $\Sigma;\vdash B\,I, I^{\perp}$. The $\underset{s\in S}{\&} \llbracket s \rrbracket_{\mathcal{K}}$ formula in $I^{\perp}$ forces us to prove several cases. More precisely, for each $s \in \mathcal{S}$, we have to prove $\Sigma;\vdash BI, \llbracket s \rrbracket_{\mathcal{K}}$. Consider the following derivation

   $$\cfrac{\Sigma;\vdash \llbracket F \rrbracket_{\mathcal{K}}, \llbracket s \rrbracket_{\mathcal{K}} \quad \Sigma;\vdash R_1, \llbracket s \rrbracket_{\mathcal{K}} \quad \cdots \quad \Sigma;\vdash R_n, \llbracket s \rrbracket_{\mathcal{K}}}{\Sigma;\vdash \llbracket F \rrbracket_{\mathcal{K}} \mathbin{\&} R_1 \mathbin{\&} \cdots \mathbin{\&} R_n, \llbracket s \rrbracket_{\mathcal{K}}}\ \&,$$

   where $R_i = \mathtt{neg}(\mathtt{s}_i) \oplus (\mathtt{pos}(\mathtt{s}_i) \otimes (\llbracket \mathtt{s}'_i \rrbracket_{\mathcal{K}} \parr I))$. Again, we have several cases to prove.
   The first sequent $\Sigma;\vdash F, \llbracket s \rrbracket_{\mathcal{K}}$ follows from inductive hypothesis.
   If the rule r_i is not fireable at state s, then the sequent $\Sigma;\vdash \llbracket s \rrbracket_{\mathcal{K}}, R_i$ is provable (by choosing $\mathtt{neg}(\mathtt{s}_i)$). On the other hand, if r_i is fireable at state s, we then have

   $$\cfrac{\Sigma;\vdash \llbracket s' \rrbracket_{\mathcal{K}}, I}{\Sigma;\vdash R_i, \llbracket s \rrbracket_{\mathcal{K}}}\ \oplus, \otimes, \&.$$

   Since $\mathcal{S}$ is closed under $\longrightarrow$, it must be the case that $s' \in \mathcal{S}$ and hence the sequent $\Sigma;\vdash \llbracket s' \rrbracket_{\mathcal{K}}, I$ is provable (as in **Premise** 1 above).

The case EG is similar.

($\Leftarrow$) Due to focusing, we can show that the derivations in the $\Rightarrow$ part are the only way to proceed during a proof in (focused) μMALL. Hence, we match exactly a 'step' in the deduction of $\mathtt{s} \models^{K}_{CTL} F$. Hence, the only interesting case is the one of the greatest fixed point operator. Consider the CTL formula $\mathsf{AG}F$ and assume that we have a proof of the sequent $\Sigma;\vdash \llbracket \mathtt{s} \rrbracket_{\mathcal{K}}, \nu B$ with invariant I_x. This means that we have a proof of the sequent $\Sigma;\vdash \llbracket \mathtt{s} \rrbracket_{\mathcal{K}}, I_x$. Moreover, due to the shape of B, we must also have a proof of $\Sigma;\vdash \llbracket \mathtt{s}' \rrbracket_{\mathcal{K}}, I_x$ for any reachable state $\mathtt{s}'$. Then, we can show that there is a proof of $\Sigma;\vdash I_x, \underset{s\in\mathcal{S}}{\&} \llbracket s \rrbracket_{\mathcal{K}}$. Let I be the invariant in the proof of the $\Rightarrow$ part. Note that $I^{\perp} = \underset{s\in\mathcal{S}}{\&} \llbracket s \rrbracket_{\mathcal{K}}$ and hence $\Sigma;\vdash I_x, I^{\perp}$ (*i.e.*, $\Sigma;\vdash I \multimap I_x$) is provable. This shows that I is greater than I_x, thus we also have a proof of $\Sigma;\vdash [\![\mathtt{s}]\!]_{\mathcal{K}}, \nu B$ using I. The result follows from a derivation similar to the one used in the proof of the $\Rightarrow$ part. □

Finally, it is worth noticing that, in Definition 5.2, we do not encode the transition rules as a theory (as we did in Section 5.1). In fact, consider the following: (1) the presence of a formula of the shape $[\![\mathtt{s} \longrightarrow \mathtt{s}']\!]_{\mathcal{K}}$ in the context allows us to move from the current state to a successor one; (2) fixed points operators must be applied in order to go through paths, checking properties on them. Now, actions (1) and (2) should be coordinated, otherwise one would lose adequacy in the encodings. More precisely, by focusing on $[\![\mathtt{s} \longrightarrow \mathtt{s}']\!]_{\mathcal{K}}$, we may 'jump' a state without checking the needed property in that state. For avoiding these problems, we internalized the transition rules directly into the encoding.

5.3. *CTL in μHyMALL*

The encoding on μMALL in the previous section is heavy in two specific ways: (1) the current state of the automaton is managed by means of the `neg` and `pos` predicates, and (2) the encoding of formulas is not compositional as it is sensitive to the transition system R. These aspects limit us from even stating and proving properties of the encoding that are independent of the transition system. For instance, it is obvious from the semantics that $\mathsf{AG}F$ implies $\mathsf{EG}F$ regardless of what F or R are, and this can even be seen as a direct consequence of $(A \mathbin{\&} B) \multimap (A \oplus B)$ being true in LL, but we are prevented from writing that implication generically for any R. These issues can be addressed by means of an encoding using μHyMALL instead of μMALL.

The key difference in the encoding in μHyMALL is that we can encode the transition system directly by means of a *non-recursive* least fixed point expression, i.e., a table. We write this as the predicate `trans` that can be derived from a set of transition rules R as follows:

$$\mathtt{trans} \triangleq \mu\Big(\lambda T.\, \lambda u.\, \lambda v.\, \bigoplus_{(s,s')\in R} (s = u \otimes s' = v)\Big).$$

From the definition of `trans`, we have that, for any given s, s':
- `trans` $s\, s' \multimap$ `trans` $s\, s' \otimes$ `trans` $s\, s'$ and
- `trans` $s\, s' \multimap 1$.

These statements are easy to prove, starting with $[\multimap R]$ and then using $[\mu L]$ (with any invariant since `trans` is not recursive). Note that for any t, $t = t$ is logically equivalent to 1. Moreover, if t and t' are different terms, then $csu(t, t')$ is empty and a formula $t = t'$ on the left of the sequent finishes any derivation (using $[= L]$).

Definition 5.3 (CTL into μHyMALL). Let $\mathcal{K} = \langle S, I, R, L\rangle$ be a Kripke structure on a set of atomic propositions $\mathcal{P}$. Let `trans` be the predicated defined as above on R. The encoding $[\![\cdot]\!]$ of CTL temporal formulas, i.e., of $\mathbf{Q}\mathsf{X}$, $\mathbf{Q}\mathsf{F}$, $\mathbf{Q}\mathsf{G}$ and $\mathbf{Q}\mathsf{U}$, for $\mathbf{Q} \in \{\mathsf{A}, \mathsf{E}\}$ into μHyMALL is in Figure 9.

Theorem 5.2 (Adequacy). Let $\mathcal{K} = \langle S, I, R, L\rangle$ be a Kripke structure on a set of atomic propositions $\mathcal{P}$, $\mathtt{s} \in S$ be a state and F be a CTL formula. Then, the μHyMALL sequent: $\cdot;\cdot \vdash [\![F]\!] \mathbin{@} s$ is derivable if and only if $\mathtt{s} \models^{\mathcal{K}}_{CTL} F$.

Proof. The proof follows the same argument in the proof of Theorem 5.1. □

$$
\begin{array}{lcl}
\llbracket \mathsf{AX}F \rrbracket & = & {\downarrow} u.\, \forall w.\, \mathtt{trans}\ u\ w \otimes (\llbracket F \rrbracket\ \mathtt{at}\ w) \\
\llbracket \mathsf{EX}F \rrbracket & = & {\downarrow} u.\, \exists w.\, \mathtt{trans}\ u\ w \otimes (\llbracket F \rrbracket\ \mathtt{at}\ w) \\
\llbracket \mathsf{AF}F \rrbracket & = & \mu(\lambda R.\, \llbracket F \rrbracket \oplus {\downarrow} u.\, \forall w.\, \mathtt{trans}\ u\ w \otimes (R\ \mathtt{at}\ w)) \\
\llbracket \mathsf{EF}F \rrbracket & = & \mu(\lambda R.\, \llbracket F \rrbracket \oplus {\downarrow} u.\, \exists w.\, \mathtt{trans}\ u\ w \otimes (R\ \mathtt{at}\ w)) \\
\llbracket \mathsf{AG}F \rrbracket & = & \nu(\lambda R.\, \llbracket F \rrbracket \mathbin{\&} {\downarrow} u.\, \forall w.\, \mathtt{trans}\ u\ w \otimes (R\ \mathtt{at}\ w)) \\
\llbracket \mathsf{EG}F \rrbracket & = & \nu(\lambda R.\, \llbracket F \rrbracket \mathbin{\&} {\downarrow} u.\, \exists w.\, \mathtt{trans}\ u\ w \otimes (R\ \mathtt{at}\ w)) \\
\llbracket \mathsf{A}[F\ \mathsf{U}\ G] \rrbracket & = & \mu(\lambda R.\, \llbracket G \rrbracket \oplus (\llbracket F \rrbracket \mathbin{\&} {\downarrow} u.\, \forall w.\, \mathtt{trans}\ u\ w \otimes (R\ \mathtt{at}\ w))) \\
\llbracket \mathsf{E}[F\ \mathsf{U}\ G] \rrbracket & = & \mu(\lambda R.\, \llbracket G \rrbracket \oplus (\llbracket F \rrbracket \mathbin{\&} {\downarrow} u.\, \exists w.\, \mathtt{trans}\ u\ w \otimes (R\ \mathtt{at}\ w)))
\end{array}
$$

Fig. 9. Encoding of CTL into μHyMALL (See Definition 5.3).

Observe that in this encoding, the task of establishing the successor state is delegated to the multiplicative subformula $\mathtt{trans}\ u\ v$ in each case. The multiplicative split guarantees that it cannot consume any other linear assumptions. However, since `trans` unfolds into a disjunction of equations, there is no possible way for it to consume any linear resources in the first place. Note also that this predicate is the only one in the encoding that needs to quantify over worlds. This is typical of encodings in μHyMALL (or μHyLL): any inductive reachability relation that needs to be encoded on worlds can be represented as a least fixed point predicate.

As mentioned at the start of this section, the encoding in μHyMALL allows us to prove meta-theoretic properties of CTL such as, for any F, $\cdot ; \llbracket \mathsf{AG}F \rrbracket\ @\ s \vdash \llbracket \mathsf{AF}F \rrbracket\ @\ s$. This proof does not require examining the `trans` definition at all. In fact, all the characteristic properties of CTL given at the start of Section 5.2 can be proved as theorems in μHyMALL.

6. Concluding remarks and future work

We compared the expressiveness, as logical frameworks, of two extensions of LL. We showed that it is possible to adequately encode HyLL's logical rules into LL. In order to better analyze the meaning of worlds in HyLL, we showed that a flat subexponential structure (for worlds) suffices to encode HyLL into $\mathrm{SELL}^{\Cap}$. We also showed that information confinement cannot be specified in HyLL. Finally, with better insights about the meaning of HyLL's words, we pushed forward previous attempts of using HyLL to encode CTL. We showed that only by using meta-level induction (or fixed points inside the logic) it is possible to faithfully encode CTL path quantifiers.

There are some other logical frameworks that are extensions of LL, for example, HLF (Reed 2006). Being a logic in the LF family, HLF is based on natural deduction, hence having a complex notion of ($\beta\eta$) normal forms as well as lacking a focused system. Thus adequacy (of encodings of systems in HLF) results are often much harder to prove in HLF than in HyLL or in SELL.

While logical frameworks should be general enough for specifying and verifying properties of a large number of systems, some logical frameworks may be more suitable for dealing with specific applications than others. Hence, it makes little sense to search for 'the universal logical framework.' However, it is often salutary to establish connections between frameworks, especially when they are meant to reason about the same set of systems.

In this context, both HyLL and SELL have been used for formalizing and analysing several systems. This work indicates that SELL is a broader framework for handling such systems, since it can encode HyLL's rules and worlds naturally and directly. However, the simplicity of HyLL may be of interest for specific purposes, such as building tools for diagnosis in biomedicine. Moreover, as shown in Section 5.3, HyLL offers an elegant way of specifying transitions systems and their properties (written in CTL).

Formal proofs in HyLL were implemented in de Maria et al. (2014), in the Coq proof assistant. It would be interesting to extend the implementations of HyLL given there to μHyMALL. Such an interactive proof environment would enable both formal studies of encoded systems in μHyMALL and formal meta-theoretical study of μHyMALL itself.

Acknowledgements

We would like to thank Dale Miller for being such a good mentor and colleague. We also thank the anonymous reviewers for their valuable comments on an earlier draft of this article.

This work was supported by CAPES, Colciencias, and INRIA via the STIC AmSud project 'EPIC: EPistemic Interactive Concurrency'(Proc. No 88881.117603/2016-01). The work of Pimentel and Olarte was also supported by CNPq and the project FWF START Y544-N23.

References

Arias, J., Desainte-Catherine, M., Olarte, C. and Rueda, C. (2015). Foundations for reliable and flexible interactive multimedia scores. In: Collins, T., Meredith, D. and Volk, A. (eds.) *MCM 2015*, Lecture Notes in Computer Science, vol. 9110, Springer, 29–41.

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

Baelde, D. (2012). Least and greatest fixed points in linear logic. *ACM Transactions on Computational Logic* **13** (1) 2.

Burch, J.R., Clarke, E.M., McMillan, K.L., Dill, D.L. and Hwang, L.J. (1992). Symbolic model checking: 10^20 states and beyond. *Information and Computation* **98**(2) 142–170.

Clarke, E.M. and Emerson, E.A. (1981). Design and synthesis of synchronization skeletons using branching-time temporal logic. In: Kozen, D. (ed.) *Logics of Programs, Workshop, Yorktown Heights*, May 1981, Lecture Notes in Computer Science, vol. 131, New York: Springer, 52–71.

Chaudhuri, K. (2010). Classical and intuitionistic subexponential logics are equally expressive. In: Dawar, A. and Veith, H. (eds.) *CSL 2010*, Lecture Notes in Computer Science, vol. 6247, Springer, 185–199.

Church, A. (1940). A formulation of the simple theory of types. *The Jornal of Symbolic Logic* **5** 56–68.

Cervesato, I. and Pfenning, F. (2002). A linear logical framework. *Information and Computation* **179**(1) 19–75.

Caires, L., Pfenning, F. and Toninho, B. (2016). Linear logic propositions as session types. *Mathematical Structures in Computer Science* **26**(3) 367–423.

Chaudhuri, K. and Reis, G. (2015). An adequate compositional encoding of bigraph structure in linear logic with subexponentials. In: Davis, M., Fehnker, A., McIver, A. and Voronkov, A. (eds.) *LPAR-20 2015*, Lecture Notes in Computer Science, vol. 9450, Springer, 146–161.

Despeyroux, J. and Chaudhuri, K. (2014). A hybrid linear logic for constrained transition systems. In: *Post-Proc. of TYPES 2013*, *Leibniz Intl. Proceedings in Informatics*, vol. 26, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 150–168.

Danos, V., Joinet, J.-B. and Schellinx, H. (1993). The structure of exponentials: Uncovering the dynamics of linear logic proofs. In: Gottlob, G., Leitsch, A. and Mundici, D. (eds.) *Kurt Gödel Colloquium*, Lecture Notes in Computer Science, vol. 713, Springer, 159–171.

de Maria, E., Despeyroux, J. and Felty, A. (2014). A logical framework for systems biology. In: *Proceedings of the 1st Intl. Conference on Formal Methods in Macro-Biology (FMMB)*, Lecture Notes in Computer Science, vol. 8738, Springer, 136–155.

Despeyroux, J., Olarte, C. and Pimentel, E. (2017). Hybrid and subexponential linear logics. *Electronic Notes in Theoretical Computer Science* **332** 95–111.

Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science* **50** 1–102.

Miller, D. (1992). Unification under a mixed prefix. *Journal of Symbolic Computation* **14**(4) 321–358.

Miller, D. and Pimentel, E. (2013). A formal framework for specifying sequent calculus proof systems. *Theoretical Computer Science* **474** 98–116.

Nigam, V. (2014). A framework for linear authorization logics. *Theoretical Computer Science* **536** 21–41.

Nigam, V. and Miller, D. (2009). Algorithmic specifications in linear logic with subexponentials. In: Porto, A. and López-Fraguas, F.J. (eds.) *PPDP*, ACM, 129–140.

Nigam, V. and Miller, D. (2010). A framework for proof systems. *Journal of Automated Reasoning* **45**(2) 157–188.

Nigam, V., Olarte, C. and Pimentel, E. (2013). A general proof system for modalities in concurrent constraint programing. In: *CONCUR*, Lecture Notes in Computer Science, vol. 8052, Springer Verlag, 410–424.

Nigam, V., Olarte, C. and Pimentel, E. (2017). On subexponentials, focusing and modalities in concurrent systems. *Theoretical Computer Science* **693** 35–58.

Nigam, V., Pimentel, E. and Reis, G. (2011). Specifying proof systems in linear logic with subexponentials. *Electronic Notes in Theoretical Computer Science* **269** 109–123.

Nigam, V., Pimentel, E. and Reis, G. (2016). An extended framework for specifying and reasoning about proof systems. *Journal of Logic and Computation* **26**(2) 539–576.

Olarte, C., Chiarugi, D., Falaschi, M. and Hermith, D. (2016). A proof theoretic view of spatial and temporal dependencies in biochemical systems. *Theoretical Computer Science* **641** 25–42.

Olarte, C., Pimentel, E. and Nigam, V. (2015). Subexponential concurrent constraint programming. *Theoretical Computer Science* **606** 98–120.

Olarte, C., Pimentel, E. and Rueda, C. (2018). A concurrent constraint programming interpretation of access permissions. *TPLP* **18**(2) 252–295.

Pimentel, E., Olarte, C. and Nigam, V. (2014). A proof theoretic study of soft concurrent constraint programming. *Theory and Practice of Logic Programming* **14** 475–308.

Reed, J. (2006). Hybridizing a logical framework. In: *International Workshop on Hybrid Logic (HyLo)*, Electronic Notes in Theoretical Computer Science, Seattle, USA, Elsevier, 135–148.