RAIRO-Theor. Inf. Appl. 47 (2013) 25–68
DOI: 10.1051/ita/2012028

Available online at:
www.rairo-ita.org

STRONG FUNCTORS AND INTERLEAVING FIXPOINTS IN GAME SEMANTICS*

Pierre Clairambault[1]

Abstract. We describe a sequent calculus μLJ with primitives for inductive and coinductive datatypes and equip it with reduction rules allowing a sound translation of Gödel's system T. We introduce the notion of a *μ-closed category*, relying on a uniform interpretation of open μLJ formulas as strong functors. We show that any μ-closed category is a sound model for μLJ. We then turn to the construction of a concrete μ-closed category based on Hyland-Ong game semantics. The model relies on three main ingredients: the construction of a general class of strong functors called *open functors* acting on the category of games and strategies, the solution of recursive arena equations by exploiting *cycles* in arenas, and the adaptation of the winning conditions of parity games to build initial algebras and terminal coalgebras for many open functors. We also prove a weak completeness result for this model, yielding a normalisation proof for μLJ.

Mathematics Subject Classification. 18C50, 03F05, 68Q55, 91A40.

1. Introduction

The idea of enriching proof systems or total programming languages with syntactic constructions for induction and coinduction is nowadays very common: inductive and coinductive types and their generalisations have been studied extensively in dependent type theory [29] but were also considered in a lot of other

Keywords and phrases. Game semantics, strong functors, initial algebras, terminal coalgebras, inductive types, coinductive types.

* *Partially supported by the French ANR project CHOCO and the UK EPSRC grant EP/HO23097.*

[1] Computer Laboratory, University of Cambridge, 15 J. J. Thomson Avenue, Cambridge CB3 0FD, U.K. pierre.clairambault@cl.cam.ac.uk

Article published by EDP Sciences

© EDP Sciences 2013

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

settings, including the λ^{μ}-calculus [1] and $\mu MALL$ [4]. Motivations are multiple, but often amount to increasing the expressive power of a language without paying the price of exponential modalities (as in [4]) or impredicativity (as in [1] or [16]). Such constructions also appear in the *μ-calculus* [3], a modal logic used in verification to specify complex properties of infinite graphs. Terms in the μ-calculus can in turn be represented by *parity games*, a standard tool in the theory of automata recognising infinite objects [37]. In parity games, one can express inductive and coinductive behaviour by considering cyclic games along with winning conditions on infinite plays: in an inductive structure, it is a loss for the first player (Player, Eve, ...) if the play loops infinitely often. In a coinductive structure, it is the responsibility of the other player (Opponent, Adam, ...) to prevent this infinite looping from happening (see for instance [36]). A natural question is then: can these games serve as a basis for a model of a total programming language with induction and coinduction? Our answer is: not directly, since although one can define a category of parity games and strategies (as is implicit in [6]), this category lacks the appropriate structure to interpret the main components of a programming language, in particular it is not closed. In this paper we show that although parity games cannot be directly applied to model a total programming language with induction and coinduction, this can be done by importing the winning conditions described above in the framework of Hyland-Ong game semantics [23]. The target language of our study will be μLJ, an intuitionistic sequent calculus obtained by enriching LJ with induction and coinduction. The construction of this model however requires us to address a number of issues.

Loops and open functors. First, we need to introduce the loop construction on arenas, which in the absence of winning conditions should relate to McCusker's model of recursive types [31] in the sense that it should provide *minimal invariants* [17–19]. To formalise it, we are brought to consider *open arenas*, *i.e.* arenas with special moves called *holes* representing free variables. From open arenas we build *open functors*, which contain all the functors built using the base type constructors, and for which the loop constructions allow to define (parametrised) minimal invariants. In the presence of winning conditions, these minimal invariants split into initial algebras and terminal coalgebras for open functors.

μ-closed categories. Second, we need to spell out what is a categorical model of μLJ, since unfortunately the usual categories of games and strategies are not *algebraically complete* [17]. The idea is to require a cartesian closed category (ccc) $\mathcal{C}$ with a set of endofunctors with enough initial algebras and terminal coalgebras. In order to model the deduction rules in the presence of a non-empty context Γ these endofunctors need to be *strong* [13] (or, equivalently, *enriched* with respect to the enriched structure that $\mathcal{C}$ has over itself). These strengths can be defined by induction on types (although the case of parametrised fixpoints requires special care, see [34]). Instead, we rely here on the fact that open functors are automatically strong: we assume that the category $\mathcal{C}$ comes with a set of strong functors, and we give axioms ensuring that the strengths do correspond to those generated

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

inductively. We then define μ-closed categories as categories whose strong endofunctors have initial algebras and terminal coalgebras.

Outline. In Section 2, we introduce the logic μLJ and examine some of its proof-theoretical aspects. In Section 3 we introduce μ-closed categories, and prove they are sound models for μLJ. In Section 4, we build a concrete μ-closed category of games and strategies. We first describe the construction of open functors along with the corresponding strengths. Then we enrich the category of arenas and innocent strategies with winning conditions, and prove that the resulting category is μ-closed. Finally we study completeness properties of the resulting model.

Remark. The present paper serves as a long version for both [8] (in which we modeled the *weak* fragment of μLJ, where fixpoints and functor rules are only allowed in an empty context) and [9] (where we described the categorical structure needed to extend the model to the general rules). Let us mention some restrictions in the present work: first, loops can only be introduced in arenas when free variables do not appear in root position. Thus, representing fixpoints by loops has the effect of restricting the model to *strict* formulas, where (co)induction is guarded. Second, we were unfortunately unable to carry out the construction of open functors in the most general case and were forced to assume that holes (in open arenas) could not enable other moves. As a result, we only interpret the fragment of μLJ in which free variables never appear at the right hand side of an implication, dubbed the *pseudo-polynomial* fragment. However note that the other parts of the model are presented in full generality, thus a successful construction of all open functors would immediately yield a model of all strict formulas of μLJ.

Related work. On the game semantics side, the closest work to the present paper is the model of recursive types by McCusker [31]. The main difference is that μLJ is a total programming language so its terms are interpreted by total strategies, which has the effect of splitting recursive types into inductive and coinductive types. On the categorical side, μ-closed categories are closely related to the *strong categorical datatypes* of Cockett and Spencer [13,14], used as the basis of the programming language `Charity` [12] (with inductive and coinductive types). However μ-closed categories are more general: they include a notion of *contravariant* strong functor which allows us to support the implication connective, where strong categorical datatypes are based on (non-closed) distributive categories. Moreover we also suppose parametrised initial algebras and terminal coalgebras to be strong which allows us to support *interleaving* inductive and coinductive types, whereas `Charity` does not support mutually defined fixpoints.

2. The logic μLJ

The interest of μLJ is that it is a hybrid object, which can be seen either as a basic logic allowing reasoning by induction and coinduction or as an idealized total programming language with inductive and coinductive datatypes, as will be justified by a translation of Gödel's system T.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

$$\begin{array}{rlrl}
fv^+(S \Rightarrow T) &= fv^-(S) \cup fv^+(T) & fv^-(S \Rightarrow T) &= fv^+(S) \cup fv^-(T) \\
fv^+(S + T) &= fv^+(S) \cup fv^+(T) & fv^-(S + T) &= fv^-(S) \cup fv^-(T) \\
fv^+(S \times T) &= fv^+(S) \cup fv^+(T) & fv^-(S \times T) &= fv^-(S) \cup fv^-(T) \\
fv^+(\mu X.T) &= fv^+(T) \setminus \{X\} & fv^-(\mu X.T) &= fv^-(T) \\
fv^+(\nu X.T) &= fv^+(T) \setminus \{X\} & fv^-(\nu X.T) &= fv^-(T) \\
fv^+(X) &= \{X\} & fv^-(X) &= \emptyset \\
fv^+(1) &= \emptyset & fv^-(1) &= \emptyset \\
fv^+(0) &= \emptyset & fv^-(0) &= \emptyset
\end{array}$$

FIGURE 1. Negative and positive free variables.

2.1. DEFINITIONS

Definition 2.1 (Pre-formulas)**.** Let $\mathcal{V}$ be a countable set of variables. The set of *pre-formulas* is defined by the following grammar, with $X \in \mathcal{V}$.

$$S \ ::= \ S \Rightarrow T \mid S + T \mid S \times T \mid \mu X.T \mid \nu X.T \mid X \mid 1 \mid 0$$

On pre-formulas, we define by mutual induction the notions of free positive variables and free negative variables, as described in Figure 1. We say that F is a *formula* if for every subformula of F of the form $\mu X.F'$ or $\nu X.F'$, we have $X \notin fv^-(F')$. In other words, bound variables always appear in *positive position*. Note that unlike most works in (co)inductive types, we do not require here that the occurrences of bound variables appear only in *strictly positive* position (*i.e.*, that no occurrence of a bound variable appears at the left side of an implication). This is because our model is not set-theoretical, and hence accommodates perfectly inductive types of the form $\mu X.(X \Rightarrow 2) \Rightarrow 2$.

Definition 2.2 (Functor)**.** A formula F is a *positive functor* if $fv^-(F) = \emptyset$ and $fv^+(F)$ is a singleton[2]. We take the convention that positive functors will be denoted by P. When this causes no ambiguity, their possible free variable will be denoted by X, and we write $P(A)$ for $P[A/X]$. Similarly, F is a *negative functor* if $fv^+(F) = \emptyset$ and $fv^-(F)$ is a singleton. Negative functors will be denoted by N, and we will often write $N(A)$ instead of $N[A/X]$.

Definition 2.3 (μLJ)**.** A *sequent* is an expression $\Gamma \vdash A$, where Γ is a list of formulas and A is a formula. The deduction rules of μLJ are given in Figure 2.

2.2. DYNAMICAL BEHAVIOUR

2.2.1. Cut reduction and functor expansion

The question of which reduction rules to equip μLJ with is a delicate one. The reduction rules should hopefully always terminate, yet be powerful enough to

[2]This does not mean that F has only one occurrence of a free variable, as there may be several occurrences of the same free variable.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Identity Group

$$\frac{}{A \vdash A}\, ax \qquad \frac{\Gamma \vdash A \qquad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}\, Cut$$

Structural Group

$$\frac{\Gamma, A, A, \Delta \vdash B}{\Gamma, A, \Delta \vdash B}\, C \qquad \frac{\Gamma, \Delta \vdash B}{\Gamma, A, \Delta \vdash B}\, W \qquad \frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}\, \gamma$$

Logical Group

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} \Rightarrow_r \qquad \frac{\Gamma \vdash A \qquad \Delta, B \vdash C}{\Gamma, \Delta, A \Rightarrow B \vdash C} \Rightarrow_l$$

$$\frac{}{0 \vdash A}\, 0_l \qquad \frac{}{\vdash 1}\, 1_r$$

$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \times B} \times_r \qquad \frac{\Gamma, A, B \vdash C}{\Gamma, A \times B \vdash C} \times_l$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A + B} \overleftarrow{+_r} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A + B} \overrightarrow{+_r} \qquad \frac{\Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma, A + B \vdash C} +_l$$

Fixpoints

$$\frac{\Gamma \vdash P(\mu X.P)}{\Gamma \vdash \mu X.P}\, \mu_r \qquad \frac{\Gamma, P(A) \vdash A}{\Gamma, \mu X.P \vdash A}\, \mu_l \qquad \frac{\Gamma, P(\nu X.P) \vdash B}{\Gamma, \nu X.P \vdash B}\, \nu_l \qquad \frac{\Gamma, A \vdash P(A)}{\Gamma, A \vdash \nu X.P}\, \nu_r$$

Functors

$$\frac{\Gamma, A \vdash B}{\Gamma, P(A) \vdash P(B)}\, [P] \qquad \frac{\Gamma, A \vdash B}{\Gamma, N(B) \vdash N(A)}\, [N]$$

FIGURE 2. Deduction rules for μLJ.

allow the translation of Gödel's system T (or another programming language with induction or coinduction). Another difficulty is the fact that as a sequent calculus, the syntax of μLJ is extremely expressive; it can be thought as a programming language with explicit substitutions, weakening and contraction, and such systems must be handled with care, or they lose the normalization property [32]. Since in this paper we focus on the models of μLJ rather than on its syntax, we will not give here the (quite large) full set of reduction rules, which can be found in [10]. In this section, we just mention and illustrate the different groups of reduction rules.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

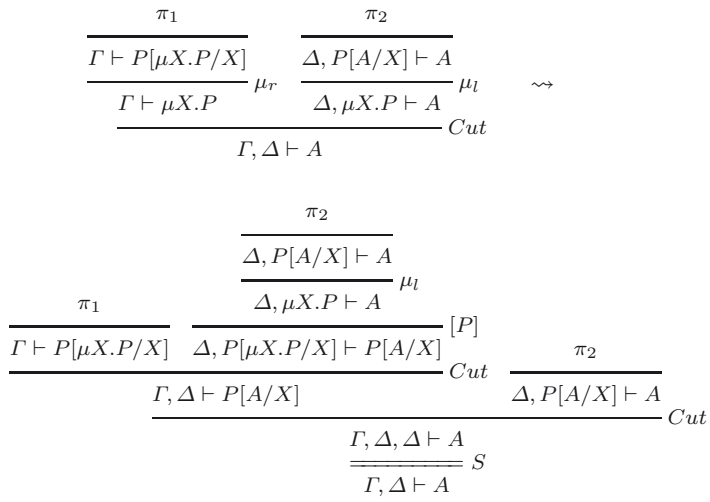


FIGURE 3. Reduction of a μ cut.

Logical cuts. This group describes the usual reduction of cuts between left and right introduction rules of the logical connectives $\times$, $+$ and $\Rightarrow$.

Fixed points. This group describes the cut reduction between left and right introduction rules of μ and ν. The case for μ is displayed in Figure 3, in which the rule S denotes a straightforward succession of structural rules. Readers familiar with the categorical models of induction may recognize here a 2-cell in the diagram of initial algebras. This reduction introduces an occurrence of the functor rule, which explains why it must be included in μLJ.

Functors. This group describes the expansion laws for functors. We give in Figure 4 the case of a functor built by taking the product of two other functors. It can be shown rather easily [10] that these expansion rules suffice to eliminate all occurrences of functor rules, hence as expected functors have no impact on provability.

Commutations. This group deals with all the other cases, ensuring all necessary commutations to get back to the above key cases.

2.2.2. Irreducible cuts

This reduction system is not complete, since some cuts cannot be eliminated. For instance, consider the following situation:

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma, T(A) \vdash A}}{\Gamma, \mu X.T \vdash A}\,\mu_l \qquad \dfrac{\pi_2}{\Delta, A \vdash B}}{\Gamma, \mu X.T, \Delta \vdash B}\,Cut$$

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\pi}{\Gamma, A \vdash B}}{\Gamma, (P_1 \times P_2)(A) \vdash (P_1 \times P_2)(B)}[P_1 \times P_2]}{\rightsquigarrow}}{\dfrac{\dfrac{\dfrac{\pi}{\Gamma, A \vdash B}}{\Gamma, P_1(A) \vdash P_1(B)}[P_1]}{\Gamma, P_1(A), P_2(A) \vdash P_1(B)}S}{\Gamma, P_1(A) \times P_2(A) \vdash P_1(B)}\times_l \quad \dfrac{\dfrac{\dfrac{\pi}{\Gamma, A \vdash B}}{\Gamma, P_2(A) \vdash P_2(B)}[P_2]}{\Gamma, P_1(A), P_2(A) \vdash P_2(B)}S}{\Gamma, P_1(A) \times P_2(A) \vdash P_2(B)}\times_l}{\times_r}}{\Gamma, P_1(A) \times P_2(A), \Gamma, P_1(A) \times P_2(A) \vdash P_1(B) \times P_2(B)}S}{\Gamma, P_1(A) \times P_2(A) \vdash P_1(B) \times P_2(B)}$$

FIGURE 4. Expansion rule for positive functor product.

There is no hope of reducing this cut while preserving termination. Indeed, π_2 should somehow commute with the μ_l rule to reach π_1, where the right introduction rule (or an axiom) for the main connective of A is found. However, this is impossible without first expanding the left premise along the lines of the diagram of initial algebras, and allowing this would yield a non-terminating system. Because of this situation, some authors [4] choose to inline this cut into the rule for μ_l thus yielding a system which does eliminate cuts, but whose cut-free proofs do not enjoy the subformula property. Hence consistency of this logic cannot be easily deduced from a cut-elimination theorem in the style of Gentzen [20], however it will be a corollary of our games model. Note that there is also the dual situation:

$$\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \quad \dfrac{\dfrac{\pi_2}{\Delta, A \vdash T(A)}}{\Delta, A \vdash \nu X.T}\nu_r}{\Gamma, \Delta \vdash B}Cut$$

Here, a useful intuition is that of *laziness*: evaluation of coinductive data must be demand-driven, not output-driven, or we have an immediate non-termination.

Let us emphasize that even if we conjecture that the reduction system on μLJ terminates we do not aim here to prove such a theorem, which for the reasons above would not yield a cut elimination result. Instead we focus on the game semantics which, as discussed in Section 4.4, will allow us to normalise μLJ proofs to potentially infinite, but cut-free, proofs.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

2.3. Translation of system T

In this section, we illustrate the dynamical behaviour of μLJ and justify the reduction rules described above by encoding Gödel's System T inside. Let us begin by defining the variant of System T that we will be using here.

Definition 2.4. The set of *types* of System T is defined by the following grammar:

$$A, B ::= \mathtt{int} \mid \mathtt{bool} \mid A \Rightarrow B \mid A \times B$$

The set of *terms* is given as follows:

$$M, N, P ::= x \mid \lambda x.M \mid MN \mid 0 \mid S\ M \mid \mathtt{tt} \mid \mathtt{ff} \mid \mathtt{if}\ M\ N\ P \mid \langle M, N\rangle \mid \pi^1\ M \mid \pi^2\ M \mid \mathtt{rec}\ M\ N\ P$$

We consider first the usual rules of simply-typed λ-calculus with products to which we add the following:

$$\frac{}{\Gamma \vdash 0 : \mathtt{int}} \qquad \frac{}{\Gamma \vdash \mathtt{tt} : \mathtt{bool}} \qquad \frac{}{\Gamma \vdash \mathtt{ff} : \mathtt{bool}}$$

$$\frac{\Gamma \vdash M : \mathtt{int}}{\Gamma \vdash S\ M : \mathtt{int}} \qquad \frac{\Gamma \vdash B : \mathtt{bool} \quad \Gamma \vdash M : A \quad \Gamma \vdash N : A}{\Gamma \vdash \mathtt{if}\ B\ M\ N : A}$$

$$\frac{\Gamma \vdash T : A \quad \Gamma \vdash M : A \Rightarrow A \quad \Gamma \vdash N : \mathtt{int}}{\Gamma \vdash \mathtt{rec}\ T\ M\ N : A}$$

These terms are then equipped with the usual reduction rules. To embed it into μLJ, we give the following translation of types:

$$\begin{aligned} \mathtt{int}^\star &= \mu X.1 + X \\ \mathtt{bool}^\star &= 1 + 1 \\ (T_1 \Rightarrow T_2)^\star &= T_1^\star \Rightarrow T_2^\star \\ (T_1 \times T_2)^\star &= T_1^\star \times T_2^\star \end{aligned}$$

The interpretation of terms is essentially straightforward. As an example, the translation of `rec` is given in Figure 5. The validity of this translation is then a rather unsurprising induction on terms, since the reductions on both sides essentially coincide. We will not say more on this subject, but the interested reader will find all necessary details in [10].

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

$$(\texttt{rec}\ T\ M\ N)^\star = \dfrac{\dfrac{N^\star}{\Gamma \vdash \texttt{nat}} \qquad \dfrac{\dfrac{\dfrac{T^\star}{\Gamma \vdash A}}{\Gamma, 1 \vdash A}\,1_l \qquad \dfrac{M^\star}{\Gamma \vdash A \Rightarrow A} \qquad \dfrac{\dfrac{}{A \vdash A}\,ax \qquad \dfrac{}{A \vdash A}\,ax}{A, A \Rightarrow A \vdash A}\,\Rightarrow_l}{\Gamma, A \vdash A}\,Cut}{\Gamma, \Gamma, 1 + A \vdash A}\,+_l \qquad \dfrac{}{\Gamma, \Gamma, \texttt{nat} \vdash A}\,\mu_l}{\Gamma, \Gamma, \Gamma \vdash A}\,Cut$$

$$\Gamma \vdash A\ S$$

FIGURE 5. Translation of the recursor.

3. μ-CLOSED CATEGORIES

In this section, we will introduce a categorical structure sufficient to interpret μLJ in a sound way. First we will describe the basic components of the model, *i.e.* a cartesian closed category $\mathcal{C}$ equipped with weak coproducts. Then, we will describe how the syntactic rules for functors and fixpoints can be dealt with by requiring the functors interpreting open types to be *strong* in an uniform way, yielding a structure of *strong types*. Finally, μ-closed categories will be defined as those categories with strong types which have enough initial algebras and terminal coalgebras.

3.1. CARTESIAN CLOSED STRUCTURE AND WEAK COPRODUCTS

If $\mathcal{C}$ is a category, let us denote by $\mathcal{C}_0$ its underlying set of objects and $\mathcal{C}(A, B)$ the set of morphisms from A to B. It seems useless here to recall the definition of a cartesian closed category, a reference book for that is [26]. The specific notations we use here are rather standard, let us recall them briefly. The composition of morphisms $f : A \to B$ and $g : B \to C$ will be denoted $f; g : A \to C$. If $f : A \to B$ and $g : A \to C$ are two morphisms, their *pairing* will be denoted by $\langle f, g \rangle$. A cartesian closed category $\mathcal{C}$ has also a terminal object, which we will denote by 1. If $f : A \times B \to C$, its *currying* will be denoted $\Lambda(f) : B \to A \Rightarrow C$. Let us turn now to the definition of the sum.

Definition 3.1. A category $\mathcal{C}$ has *functorial weak coproducts* if it has:

- For all objects A and B, an object $A + B$ and two injection morphisms $in_1 : A \to A + B$ and $in_2 : B \to A + B$
- For all morphisms $f : A \to C$ and $g : B \to C$, a copairing morphism $[f|g] : A + B \to C$
- A weak initial object 0. For every object A, we write: $0 \to A$.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Additionally, these data must satisfy the following equations:

$$in_1; [f|g] = f$$
$$in_2; [f|g] = g$$
$$[in_1|in_2] = id$$
$$[f; in_1|g; in_2]; [f'|g'] = [f; f'|g; g']$$

These equations do not ensure that we have a coproduct, only a weak coproduct. However, a direct calculation shows that they suffice to build from $+$ a bifunctor $- + - : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$.

Distributivity. As is the case with coproducts [26], the closed structure of $\mathcal{C}$ allows to build a distributivity law of the product over the sum. Thus we get a natural transformation $\gamma_{\Gamma,A,B} : \Gamma \times (A + B) \to \Gamma \times A + \Gamma \times B$

Weak coproduct. In order to interpret the sums of programming languages, it would be natural to require that the model has coproducts, over which the product distributes. However our games model of induction and coinduction will rely on the category of games used by McCusker [31], which has fixpoint operators. Therefore, as is well known, it cannot have coproducts. However it has a natural notion of weak coproducts, which will be sufficient to interpret μLJ.

3.2. Functorial extensions

Let us begin this subsection by having a close look at the strong rule for the action of a positive functor in μLJ.

$$\frac{\Gamma, A \vdash B}{\Gamma, P(A) \vdash P(B)}\,[P]$$

It is obvious that P is not acting on the base category $\mathcal{C}$. Instead, its action is on the category $\mathcal{C}_\Gamma$ over a context Γ, which can be defined as the Kleisli category over the comonad $\Gamma \times -$. Thus, the interpretation of this rule asks the question of how to compute a *functorial extension* of $T : \mathcal{C} \to \mathcal{C}$ to $\mathcal{C}_\Gamma$.

3.2.1. Covariant case

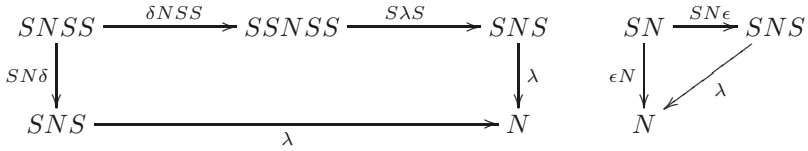
A *functorial extension* of an endofunctor T of $\mathcal{C}$ to the Kleisli category $\mathcal{C}_S$ over a comonad (S, δ, ϵ) is an endofunctor $T_S : \mathcal{C}_S \to \mathcal{C}_S$ such that for all $f : A \to B$, $T_S(\epsilon_A; f) = \epsilon_{T(A)}; T(f)$. It is folklore that the functorial extensions of an endofunctor $T : \mathcal{C} \to \mathcal{C}$ to the Kleisli category over a comonad (S, δ, ϵ) correspond to distributive laws of S over T, that is natural transformations $\lambda : ST \xrightarrow{\bullet} TS$ satisfying usual coherence conditions. The case we are concerned with in this paper is when the comonad S is $\Gamma \times -$, with the constraint that the functorial extension must be stable under substitution $\gamma : \Delta \to \Gamma$. In this case, the notion of distributive law specializes to the notion of *strength*, introduced by Kock [25], also used in a very similar way as we do here by Cockett and Spencer [13, 14].

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

3.2.2. Contravariant case

The rule $[N]$ of μLJ requires us to give as well a notion of functorial extension for a contravariant endofunctor $N : \mathcal{C}^{op} \to \mathcal{C}$. Suppose that we have a comonad (S, δ, ϵ) on $\mathcal{C}$. A functorial extension of N to the Kleisli category over S is a contravariant functor $N_S : \mathcal{C}_S^{op} \to \mathcal{C}_S$ such that for all $f : A \to B$, $N_S(\epsilon_A; f) = SN(f); \epsilon_{NA}$. To build such an extension, we propose the following contravariant cousins to distributive laws. We include the full definitions and the corresponding developments as they are, to our knowledge, new.

Definition 3.2 (Contradistributive law)**.** A *contradistributive law* for a comonad (S, δ, ϵ) over a contravariant endofunctor $N : \mathcal{C}^{op} \to \mathcal{C}$ is a natural transformation $\lambda : SNS \xrightarrow{\bullet} N$ satisfying the coherence diagrams below.



As for the case of distributive laws, contradistributive laws characterise functorial extensions of functors $N : \mathcal{C}^{op} \to \mathcal{C}$ to $\mathcal{C}_S$.

Proposition 3.3. *Let (S, δ, ϵ) a comonad on a category $\mathcal{C}$ and $N : \mathcal{C}^{op} \to \mathcal{C}$. There is an isomorphism between contradistributive laws $\lambda : SNS \xrightarrow{\bullet} N$ of S over N and functorial extensions of N on the Kleisli category $\mathcal{C}_S$.*

Proof. Let us first show how a contradistributive law induces a functorial extension. We define the functor $N_S : \mathcal{C}_S^{op} \to \mathcal{C}_S$ as follows:

- On objects, $N_S(A) = N(A)$;
- If $f : SB \to A$ is a morphism in $\mathcal{C}_S^{op}(A, B)$, we define $N_S(f)$ by:

$$SN(A) \xrightarrow{SN(f)} SNS(B) \xrightarrow{\lambda_B} N(B)$$

It is immediate to verify that this does define a functor. If $N_S : \mathcal{C}_S^{op} \to \mathcal{C}_S$ is a functorial extension of N to $\mathcal{C}_S$, we can define a family of morphisms $\lambda_A : SNSA \to NA$ by $\lambda_A = N_S(id_{SA})$. The fact that λ is natural and coherence with respect to the co-multiplication of S are easy consequences of the functoriality of N_S and the definition of composition in $\mathcal{C}_S$. Coherence with respect to the co-unit of S corresponds exactly to the equation of the definition of functorial extension, thus λ is a contradistributive law. Finally, these two constructions are inverses of one another: this boils down two the two following equations, for all $f : SB \to A$.

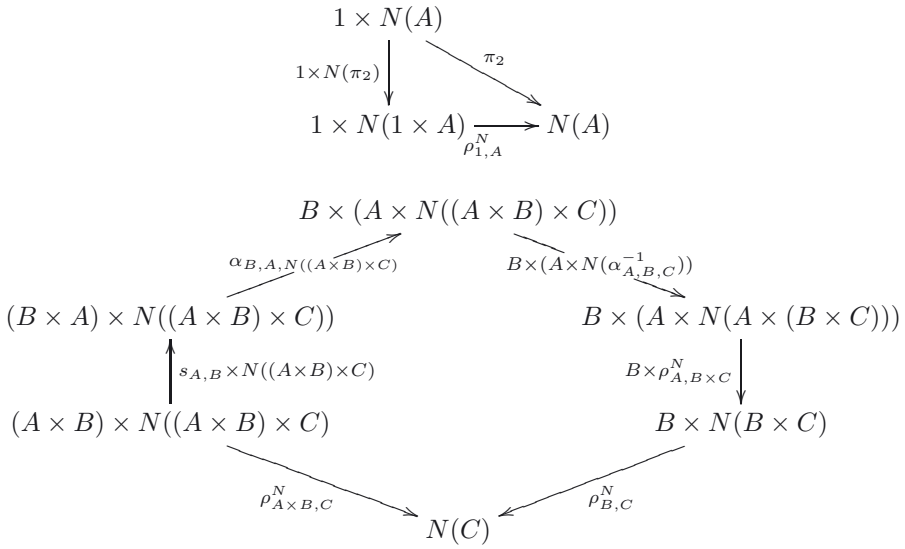
$$SN(id_{SB}); \lambda_B = \lambda_B$$
$$SN(f); N_S(id_{SB}) = N_S(f)$$

The first one is trivial, the second uses the comonad structure of S, the fact that N_S is a functorial extension of N and the definition of composition in $\mathcal{C}_S$. □

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Contrastrengths. Let us now examine the particular case of interest for us, when $S = \Gamma \times -$ with the constraint of stability under substitution $\gamma : \Delta \to \Gamma$, which corresponds to the following contravariant cousin of strengths.

Definition 3.4 (Contrastrength)**.** Suppose that $\mathcal{C}$ is cartesian and let $N : \mathcal{C}^{op} \to \mathcal{C}$. A *contrastrength* for N is a family $\rho^N_{A,B} : A \times N(A \times B) \to N(B)$ dinatural in A and natural in B, such that the following diagrams commute:



A functor $N : \mathcal{C}^{op} \to \mathcal{C}$ is *strong* when it is equipped with a contrastrength ρ.

Proposition 3.5. *Suppose that $\mathcal{C}$ is cartesian, let $N : \mathcal{C}^{op} \to \mathcal{C}$ and $\rho^N_{\Gamma,A}$ a contrastrength for N. Then for each $\Gamma \in \mathcal{C}_0$, $(\rho^N_{\Gamma,A})_A$ is a contradistributive law for $\Gamma \times -$ over N.*

Proof. It is trivial that ρ is natural in A. Coherence with respect to the co-unit $\pi_2 : \Gamma \times A \to A$ of $\Gamma \times -$ follows from the coherence of ρ with respect to the neutral element 1 for $\times$ and from the dinaturality of $\rho^N_{\Gamma,A}$ in Γ. Coherence with respect to the comultiplication follows also from dinaturality of ρ in Γ, and from the second coherence diagram of the definition of contrastrength. □

As a consequence, a contrastrength $\rho^N_{\Gamma,A}$ allows to compute a functorial extension of contravariant N over $\mathcal{C}_S$ which provides an interpretation of the rule $[N]$ of μLJ for negative functors.

3.3. Extension of fixpoints

We now know how to extend functors to $\mathcal{C}_\Gamma$, but to interpret the fixpoint rules of μLJ we also want these extensions to preserve the initial algebras and terminal coalgebras that the original functors possibly had.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Initial algebras. We shall see first that extensions preserve initial algebras, at least when the comonad (S, δ, ϵ) has a right adjoint U.

Proposition 3.6. *Let (S, δ, ϵ) be a comonad on $\mathcal{C}$, let $T : \mathcal{C} \to \mathcal{C}$ with an initial algebra $(\mu T, \chi_T)$ and let $\lambda : ST \stackrel{\bullet}{\to} TS$ be a distributive law of S over T. Suppose as well that there is $U : \mathcal{C} \to \mathcal{C}$ such that $S \dashv U$. Then T_S has also an initial algebra in $\mathcal{C}_S$.*

Proof. The initial algebra of T_S is given by the pair $(\mu T, \chi'_T)$ where $\chi'_T = \epsilon_{T(\mu T)}; \chi_T$. Let us denote by $\phi_{A,B} : \mathcal{C}(SA, B) \to \mathcal{C}(A, UB)$ the isomorphism of the adjunction $S \dashv U$. The proof is the following: for any algebra $f : STA \to A$ of T_S in $\mathcal{C}_S$, we build an algebra $\hat{f} : T(UA) \to UA$ of T in $\mathcal{C}$, such that each of the two diagrams below commutes if and only if the other commutes, for all $g : \mu T \to UA$.

$$\begin{array}{ccc} T(\mu T) & \xrightarrow{T(g)} & T(UA) \\ {\scriptstyle \chi_T}\downarrow & \mathcal{C} & \downarrow{\scriptstyle \hat{f}} \\ \mu T & \xrightarrow[g]{} & UA \end{array} \qquad \begin{array}{ccc} T_S(\mu T) & \xrightarrow{T_S(\phi^{-1}(g))} & T_S(A) \\ {\scriptstyle \chi'_T}\downarrow & \mathcal{C}_S & \downarrow{\scriptstyle f} \\ \mu T & \xrightarrow[\phi^{-1}(g)]{} & A \end{array}$$

We define $\hat{f}$ by $\phi(h)$, where h is obtained by the following composition:

$$STUA \xrightarrow{\delta_{TUA}} SSTUA \xrightarrow{S\lambda_{UA}} STSUA \xrightarrow{ST\epsilon'_A} STA \xrightarrow{f} A$$

where $\epsilon'_A : SUA \to A$ is the co-unit of the adjunction $S \dashv U$. A direct calculation then show the upper paths of these two diagrams are connected by the isomorphism ϕ, and the lower paths are connected in the same way. This proves that one diagram commutes if and only if the other commutes. But the initiality of $(\mu T, \chi_T)$ ensures uniqueness of $g : \mu T \to UA$ making the left diagram commute, hence we must as well have uniqueness of $\phi^{-1}(g)$ making the right diagram commute. □

In particular, if $\mathcal{C}$ is cartesian closed and $T : \mathcal{C} \to \mathcal{C}$ is a strong functor, then $\Gamma \times -$ has a right adjoint $\Gamma \Rightarrow -$ and any initial algebra of T extends to an initial algebra of T_Γ.

Terminal coalgebras. The situation for terminal coalgebras is slightly simpler, there is no need to require that S has a right adjoint.

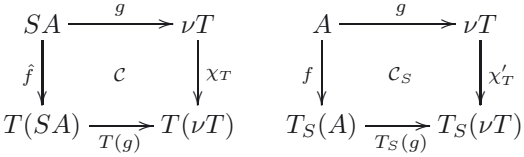
Proposition 3.7. *Let (S, δ, ϵ) be a comonad on $\mathcal{C}$, $T : \mathcal{C} \to \mathcal{C}$ with a terminal coalgebra $(\nu T, \chi_T)$ and $\lambda : ST \stackrel{\bullet}{\to} TS$ a distributive law for S over T. Then T_S also has a terminal coalgebra in $\mathcal{C}_S$.*

Proof. The terminal coalgebra of T_S is given by the pair $(\nu T, \chi'_T)$ where $\chi'_T = \epsilon_{\nu T}; \chi_T$. For any coalgebra $f : SA \to TA$ of T_S in $\mathcal{C}_S$, we build $\hat{f} : SA \to T(SA)$ by:

$$SA \xrightarrow{\delta_A} SSA \xrightarrow{Sf} STA \xrightarrow{\lambda_A} TSA$$

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Then, the definition of the extension of T allows to show that each of the two following diagrams commutes if and only if the other commutes, for each $g : SA \to \nu T$.



From this we directly deduce that $(\nu T, \chi'_T)$ is a terminal coalgebra for T_S. □

3.4. Uniformity of the extension

In the forthcoming games model open formulas are interpreted by *open functors* which, as we will show, come automatically equipped with a strength. However these strengths come from an independent construction and are in particular *not* built by induction on types, so it is not obvious that they satisfy the equations required to interpret the expansion rules for functors. We give here an axiomatic notion of a uniform set of strong functors (on a ccc with functorial weak coproducts) of which open functors will be a very natural instance, then show that these axioms suffice to ensure that strengths coincide with those generated inductively.

Definition 3.8 (Strong types)**.** A category $\mathcal{C}$ has *strong types* if $\mathcal{C}$ is cartesian closed, has functorial weak coproducts and if it is equipped with a class of functors $\mathcal{F}$ of the form $T : (\mathcal{C}^{op})^k \times \mathcal{C}^n \to \mathcal{C}$ such that

- $\mathcal{F}$ contains the identity functor, all constant functors and base functors $-_1 + -_2$, $-_1 \times -_2$ and $-_1 \Rightarrow -_2$;
- $\mathcal{F}$ is stable under composition (in the operadic sense [28]);
- $\mathcal{F}$ is stable under *contraction*: if $F(-_1, -_2, -_3) : \mathcal{C} \times \mathcal{C} \times \mathcal{D} \to \mathcal{C}$ is in $\mathcal{F}$, then $F(-_1, -_1, -_3) : \mathcal{C} \times \mathcal{D} \to \mathcal{C}$ also is; and the same condition is valid with $\mathcal{C}^{op}$ instead of $\mathcal{C}$ in the left hand side.

Moreover, unary functors in $\mathcal{F}$ are strong: they have either a strength or a contrastrength. These data must satisfy the following uniformity conditions. Let us denote by P covariant endofunctors and by N contravariant endofunctors.

- Both families $\theta^T_{\Gamma,A}$ and $\rho^T_{\Gamma,A}$ are also *natural in* T;
- Compatibility with identity and constants:

$$\theta^{Id}_{\Gamma,A} = id_{\Gamma \times A}$$
$$\theta^{B}_{\Gamma,A} = \pi_2$$
$$\rho^{B}_{\Gamma,A} = \pi_2$$

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

- Compatibility with contraction and composition:

$$\begin{aligned}
\theta^{P_1P_2}_{\Gamma,A} &= \theta^{P_1}_{\Gamma,P_2(A)}; P_1(\theta^{P_2}_{\Gamma,A}) \\
\rho^{PN}_{\Gamma,A} &= \theta^{P}_{\Gamma,N(\Gamma\times A)}; P(\rho^{N}_{\Gamma,A}) \\
\rho^{NP}_{\Gamma,A} &= \Gamma \times N(\theta^{P}_{\Gamma,A}); \rho^{N}_{\Gamma,P(A)} \\
\theta^{N_1N_2}_{\Gamma,A} &= \Gamma \times N_1(\rho^{N_2}_{\Gamma,A}); \rho^{N_1}_{\Gamma,N_2(\Gamma\times A)} \\
\theta^{P(-,-)}_{\Gamma,A} &= \langle \pi_1, \theta^{P(-,A)}_{\Gamma,A} \rangle; \theta^{P(\Gamma\times A,-)}_{\Gamma,A} \\
\rho^{N(-,-)}_{\Gamma,A} &= \langle \pi_1, \rho^{N(-,\Gamma\times A)} \rangle; \rho^{N(A,-)}_{\Gamma,A}
\end{aligned}$$

- Compatibility with the sums and the closed structure:

$$\begin{aligned}
\theta^{-+-}_{\Gamma,A} &= \gamma_{\Gamma,A,A} \\
\rho^{-\Rightarrow C}_{\Gamma,A} &= \Lambda(\langle\langle \pi_2; \pi_1, \pi_1\rangle, \pi_2; \pi_2\rangle; ev) \\
\theta^{C\Rightarrow -}_{\Gamma,A} &= \Lambda(\langle \pi_2; \pi_1, \langle \pi_1, \pi_2; \pi_2\rangle; ev\rangle)
\end{aligned}$$

Let us recall that $\gamma_{\Gamma,A,A}$ is the distributive law of product over sum.

Remark. This axiomatic definition of strengths is *complete*, in the sense that for any functor generated from the base constructors it effectively forces its strength to coincide with the inductively generated one; this will follow from the decompositions of strengths proved below. However it may not be *sound*, in the sense that there could be cartesian closed categories with functorial weak coproducts such that the set of functors generated by the base constructors does not yield a strong types structure. This is because the strengths $\theta^{T}_{\Gamma,A}$ (resp. contrastrengths $\rho^{N}_{\Gamma,A}$) are required here to be natural in T (resp. in N) and not just with respect to the natural transformations definable from the term constructors. However we do not consider that a defect: first because the model we are interested in will satisfy this property, second because our aim is to give a class of models of μLJ, not to characterise all models of μLJ. In fact the term model of μLJ itself has no chance to form a μ-closed category as its reduction rules only correspond to *weak* initial algebras and *weak* terminal coalgebras.

Lemma 3.9. *Let $\mathcal{C}$ be a category with strong types. We have the following decompositions:*

$$\begin{aligned}
\theta^{P\times P'}_{\Gamma,A} &= \langle \Gamma \times \pi_1; \theta^{P}_{\Gamma,A}, \Gamma \times \pi_2; \theta^{P'}_{\Gamma,A} \rangle \\
\theta^{P+P'}_{\Gamma,A} &= \gamma_{\Gamma,P(A),P'(A)}; \theta^{P}_{\Gamma,A} + \theta^{P'}_{\Gamma,A} \\
\theta^{N\Rightarrow P}_{\Gamma,A} &= \Lambda(\langle \pi_2; \pi_1, \langle\langle \pi_2; \pi_1, \pi_1\rangle; \rho^{N}_{\Gamma,A}, \pi_2; \pi_2\rangle; ev\rangle; \theta^{P}_{\Gamma,A}) \\
\rho^{P\Rightarrow N}_{\Gamma,A} &= \Lambda(\langle \pi_2; \pi_1, \langle\langle \pi_2; \pi_1, \pi_1\rangle; \theta^{P}_{\Gamma,A}, \pi_2; \pi_2\rangle; ev\rangle; \rho^{N}_{\Gamma,A})
\end{aligned}$$

along with identical decompositions for $\rho^{N\times N'}_{\Gamma,A}$ and $\rho^{N+N'}_{\Gamma,A}$.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Proof. Direct calculations. For the product, we use naturality of $\theta_{\Gamma,A}^{P\times P'}$ in $P \times P'$ and the naturality of the projections π_1 and π_2. For the sum, we use compatibility of strengths with composition and the distributivity law. For the arrow, we use compatibility of strengths with respect to contraction, composition and the cartesian closed structure. □

Proposition 3.10. *Let $\mathcal{C}$ a category with strong types. We have the following equations:*

$$\begin{aligned}(P \times P')_\Gamma &= P_\Gamma \times P'_\Gamma\\ (N \times N')_\Gamma &= N_\Gamma \times N'_\Gamma\\ (P + P')_\Gamma &= P_\Gamma + P'_\Gamma\\ (N + N')_\Gamma &= N_\Gamma + N'_\Gamma\\ (N \Rightarrow P)_\Gamma &= N_\Gamma \Rightarrow P_\Gamma\\ (P \Rightarrow N)_\Gamma &= P_\Gamma \Rightarrow N_\Gamma\end{aligned}$$

Proof. Obtained directly from the decompositions of Lemma 3.9. □

Higher extensions. If $\mathcal{C}$ is a category with strong types, the only functors in $\mathcal{F}$ which are natively equipped with strengths are unary ones. We show now that these suffice to build extensions of arbitrary functors in $\mathcal{F}$, which fact relies on the following lemma:

Lemma 3.11 (Commutations)**.** *Let $\mathcal{C}$ be a category with strong types and let $T \in \mathcal{F}$. We have the following commutation rules:*

(*1*) *If $T : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$:*

$$\langle \pi_1, \theta_{\Gamma,B}^{T(A,-)} \rangle ; \theta_{\Gamma,A}^{T(-,\Gamma\times B)} = \langle \pi_1, \theta_{\Gamma,A}^{T(-,B)} \rangle ; \theta_{\Gamma,B}^{T(\Gamma\times A,-)}$$

(*2*) *If $T : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C}$:*

$$\langle \pi_1, \theta_{\Gamma,B}^{T(\Gamma\times A,-)} \rangle ; \rho^{T(-,\Gamma\times B)} = \langle \pi_1, \rho_{\Gamma,A}^{T(-,B)} \rangle ; \theta_{\Gamma,B}^{T(A,-)}$$

(*3*) *If $T : \mathcal{C}^{op} \times \mathcal{C}^{op} \to \mathcal{C}$:*

$$\langle \pi_1, \rho_{\Gamma,B}^{T(\Gamma\times A,-)} \rangle ; \rho_{\Gamma,A}^{T(-,B)} = \langle \pi_1, \rho_{\Gamma,A}^{T(-,\Gamma\times B)} \rangle ; \rho_{\Gamma,B}^{T(A,-)}$$

Proof. Let us give the details of the proof of (1), the others are similar. We first remark that by functoriality of T, the family $(T(A,f))_A$ is natural in A. By naturality of θ in T, this allows us to deduce that the following diagram commutes, for all f:

$$\begin{array}{ccc} \Gamma \times T(A,B) & \xrightarrow{\Gamma\times T(A,f)} & \Gamma \times T(A,C) \\ \Big\downarrow{\scriptstyle \theta_{\Gamma,A}^{T(-,B)}} & & \Big\downarrow{\scriptstyle \theta_{\Gamma,A}^{T(-,C)}} \\ T(\Gamma \times A,B) & \xrightarrow[T(\Gamma\times A,f)]{} & T(\Gamma \times A,C) \end{array}$$

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

But this diagram exactly states that the following family is a natural transformation:

$$\left(\theta_{\Gamma,A}^{T(-,B)}\right)_B : \Gamma \times T(A,-) \xrightarrow{\bullet} T(\Gamma \times A,-)$$

Hence, another use of the naturality of θ in T gives the following equality:

$$\theta_{\Gamma,B}^{\Gamma\times T(A,-)}; \theta_{\Gamma,A}^{T(-,\Gamma\times B)} = \Gamma \times \theta_{\Gamma,A}^{T(-,B)}; \theta_{\Gamma,B}^{T(\Gamma\times A,-)}$$

from which we deduce the required equation, thanks to the decomposition of Lemma 3.9. □

Proposition 3.12. *Let us consider a functor $T : (\mathcal{C}^{op})^k \times \mathcal{C}^n \to \mathcal{C}$ in $\mathcal{F}$. Then there is an unique functor $T_\Gamma : (\mathcal{C}_\Gamma^{op})^k \times \mathcal{C}_\Gamma^n \to \mathcal{C}_\Gamma$ such that for all $i \in \{1,\dots,p,p+1,\dots,p+k\}$, for all $A_1,\dots,A_{k+p-1}$ in $\mathcal{C}_0$,*

$$(T(A_1,A_2,\dots,-_i,\dots,A_{k+n}))_\Gamma = T_\Gamma(A_1,A_2,\dots,-_i,\dots,A_{k+n})$$

Proof. The definition of T_Γ is componentwise. The fact that it does define a functor are immediate consequences of Lemma 3.11. □

3.5. μ-CLOSED CATEGORIES

3.5.1. Definition and properties

Partial interpretation. Let us suppose that the destination category $\mathcal{C}$ has strong types. If $\mathcal{X}$ is a set of free variables, let us denote by $\mathcal{C}^{\mathcal{X}}$ the cartesian product of $|\mathcal{X}|$ copies of $\mathcal{C}$. We interpret any formula F of LJ, with free variables in $\mathcal{X}$ as an endofunctor:

$$[\![F]\!] : (\mathcal{C}^{op} \times \mathcal{C})^{\mathcal{X}} \to \mathcal{C}$$

in the set $\mathcal{F}$ of strong types. This interpretation depends on the chosen set of variables $\mathcal{X}$. Formally, we should thus write $[\![F]\!]_{\mathcal{X}}$ but we will keep this annotation implicit whenever it can be recovered from the context. If F is a closed formula, we will identify the constant functor $[\![F]\!]_\emptyset$ with its image $[\![F]\!]_\emptyset() \in \mathcal{C}_0$. Let us underline that this interpretation of open formulas by functors is just an intermediate step, as we are only interested in the interpretation of closed formulas, on which the rules of μLJ are defined.

The above total interpretation of open LJ formulas extends to a partial interpretation of open μLJ formulas, as follows.

Definition 3.13 (Partial interpretation)**.** Let $\mathcal{C}$ be a category with strong types. We define by induction the interpretation of open μLJ formulas in the set $\mathcal{F}$ of strong functors for $\mathcal{C}$.

- $[\![0]\!]_{\mathcal{X}}$ is the constant functor whose image is the weak initial object 0;
- $[\![1]\!]_{\mathcal{X}}$ is the constant functor whose image is the terminal object 1;

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

- $[\![S \times T]\!]_{\mathcal{X}} = [\![S]\!]_{\mathcal{X}} \times [\![T]\!]_{\mathcal{X}}$;
- $[\![S + T]\!]_{\mathcal{X}} = [\![S]\!]_{\mathcal{X}} + [\![T]\!]_{\mathcal{X}}$;
- $[\![S \Rightarrow T]\!]_{\mathcal{X}} = [\![S]\!]_{\mathcal{X}} \Rightarrow [\![T]\!]_{\mathcal{X}}$
- $[\![\mu X.T]\!]_{\mathcal{X}\setminus\{X\}}$ is, if defined, the parametrised initial algebra of $[\![T]\!]_{\mathcal{X}}$;
- $[\![\nu X.T]\!]_{\mathcal{X}\setminus\{X\}}$ is, if defined, the parametrised terminal coalgebra of $[\![T]\!]_{\mathcal{X}}$.

Note that in these two last cases, the formula $\mu X.T$ is only valid when $X \notin fv^-(T)$, thus $[\![T]\!]_{\mathcal{X}}$ can be regarded as a functor with domain $\mathcal{C} \times (\mathcal{C}^{op} \times \mathcal{C})^{\mathcal{X}\setminus\{X\}}$. It is with respect to this domain that we take the possible parametrised initial algebra or terminal coalgebra.

Then, following the definition of *μ-bicomplete categories* [36], we say that $\mathcal{C}$ is μ-closed if it has strong types and has enough initial algebras and terminal coalgebras to interpret the rules of μLJ.

Definition 3.14 (μ-closed category)**.** A category $\mathcal{C}$ is *μ-closed* if it has strong types, and if the interpretation function $[\![-]\!]$ is total.

Example 3.15. There are a lot of μ-closed categories. For an example, McCusker's games model of the language FPC [31] forms a μ-closed category, where initial algebras and terminal coalgebras both correspond to minimal invariants [17–19]. More generally, any category of domains with minimal invariants forms a μ-closed category, but degenerate in the sense that initial algebras and terminal coalgebras coincide. It is more difficult to build a non-degenerate μ-closed category where least and greatest fixpoints no longer coincide, and this was our work in [8].

Set theory. Let us note in passing that **Set** is *not* a μ-closed category. Indeed, if we define $2 = 1 + 1$, any μ-closed category will have an interpretation of the formula $\mu X.(X \to 2) \to 2$. In **Set**, its interpretation would be a set E satisfying the isomorphism $(E \Rightarrow \{0,1\}) \Rightarrow \{0,1\} \cong E$, which means $\mathcal{P}(\mathcal{P}(E)) \cong E$: this is impossible for evident cardinality reasons. This is why numerous authors restrict to *strictly positive* formulas, *i.e.* formulas such that in $\mu X.T$, no occurrence of X can appear at the left of an arrow (even if it is in positive position).

Before going on to the interpretation of μLJ, we need to prove equations similar to Proposition 3.10 for parametrised initial algebras and terminal coalgebras.

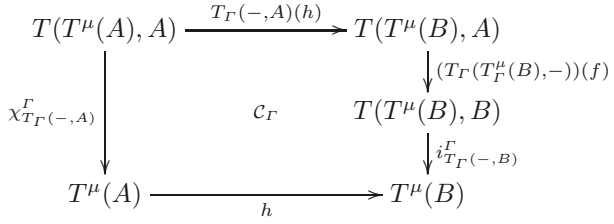
Proposition 3.16. *Suppose $\mathcal{C}$ is a μ-closed category with class $\mathcal{F}$ of strong functors. Let $\mathcal{D}$ be a category of the form $(\mathcal{C}^{op})^k \times \mathcal{C}^n$. We use the notation $\mathcal{D}_\Gamma$ to denote $(\mathcal{C}_\Gamma^{op})^k \times (\mathcal{C}_\Gamma)^n$. Let $T : \mathcal{C} \times \mathcal{D} \to \mathcal{C}$ in $\mathcal{F}$. If T has a parametrised initial algebra $T^\mu : \mathcal{D} \to \mathcal{C}$ in $\mathcal{C}$, then T_Γ also has a parametrised initial algebra $(T_\Gamma)^\mu : \mathcal{D}_\Gamma \to \mathcal{C}_\Gamma$. Moreover,*

$$(T^\mu)_\Gamma = (T_\Gamma)^\mu$$

Parametrised terminal coalgebras behave in the same way.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Proof. Let us give the proof for the case of parametrised initial algebras. The fact that T_Γ still has a parametrised initial algebra is an immediate consequence of Proposition 3.6, by definition of parametrised initial algebras. We must now show the announced equality. For the simplicity of presentation, we just detail the case when $T : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$, but the other cases are either similar or can be deduced from it. We recall that for all $f : \Gamma \times A \to B$, $(T_\Gamma)^\mu(f)$ is defined as the unique morphism h such that:



commutes. Thus to prove that $(T^\mu)_\Gamma(f) = (T_\Gamma)^\mu$, it suffices to prove that $h = (T^\mu)_\Gamma(f)$ makes this diagram commute, which is ensured by a tedious calculation using naturality of θ in T, naturality of χ and compatibility of θ with contraction and composition. □

3.5.2. Interpretation of μLJ

As usual, the interpretation of a sequent $\Gamma_1, \ldots, \Gamma_n \vdash A$ will be interpreted as a morphism $[\![\Gamma_1]\!] \times \cdots \times [\![\Gamma_n]\!] \to [\![A]\!]$ (let us say that this n-fold product is left-bracketed). For any proof π of a sequent $\Gamma \vdash A$, we define $[\![\pi]\!]$ by induction. The interpretation of most connectives is usual, and the interpretation of the rules for fixpoints and functors precisely uses the structure we introduced above, thus we will not give here the details of the interpretation nor of its correctness, however they can be found in [10]. We get the following theorem:

Theorem 3.17. *Every μ-closed category is a sound model of μLJ.*

3.5.3. Strict and pseudo-polynomial case

As announced in the introduction, limitations in the games model will require us to restrict the domain of our interpretation.

Pseudo-polynomial formulas. The first restriction comes from the fact that the construction of open functors could only be carried out when holes (free type variables) do not enable anything. Syntactically speaking this means that type variables must not appear directly at the right of an arrow, thus $A \Rightarrow X$ is not allowed, but $A \Rightarrow (0 + X)$ is. As this is not a natural fragment of μLJ we restrict instead to the (smaller) set of *pseudo-polynomial* formulas: those are formulas F such that for all subformulas $S \Rightarrow T$ of F, we always have $fv(T) = \emptyset$. This class is still sufficiently large to be interesting, it includes for example fixpoints for all polynomial functors and other exotic functors like $(X \Rightarrow 2) \Rightarrow 2$.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Strict fixpoints. The second restriction comes from the fact that we will generate fixpoints in the game semantics by a *loop construction*, which will erase the hole (*i.e.* the free variable) and replace it with links to the roots of the arena. As this operation does not seem to make any sense whenever a hole is itself initial in the arena, we will restrict to *strict* fixpoints, which means that whenever we form $\mu X.T$ or $\nu X.T$, we require that X must not appear at the *root* of T: there must be at least a $+$ or a $\Rightarrow$ in the path from the root of T to X.

It is easy to check that all previous work on μ-closed categories adapts smoothly to this strict and pseudo-polynomial setting [10]. A category $\mathcal{C}$ is μ-closed for strict and pseudo-polynomial formulas whenever it admits a total interpretation of strict and pseudo-polynomial formulas. Then, any category which is μ-closed for strict and pseudo-polynomial formulas is a model of the strict and pseudo-polynomial fragment of μLJ.

4. Game semantics

In this section, we progressively build a μ-closed category (for strict, pseudo-polynomial formulas) of games and total strategies. First of all, we will recall briefly the main definitions of the (Hyland-Ong-Nickau) category of arenas and innocent strategies. Then we turn to the description of one of the novelties of our model, the construction of strong functors from open arenas, *i.e.* arenas with free variables, and the related loop construction for recursive types. Then we will show how to enrich the games structure with winning conditions, and in particular the winning conditions on loops inspired from parity games: this will achieve the construction of a μ-closed (for pseudo-polynomial, strict formulas) category.

4.1. The base category

We briefly recall the now usual definitions of arena games, introduced in [23]. We are interested in games with two participants: Opponent (O, the *environment*) and Player (P, the *program*). Possible plays are generated by directed graphs called *arenas*, which are semantic versions of *types* or *formulas*. Hence, a play is a sequence of *moves* of the ambient arena, each of them being annotated by a *pointer* to an earlier move — these pointers being required to comply with the structure of the arena.

4.1.1. Arenas and Plays

Formally, an *arena* is a structure $A = (M_A, \lambda_A, I_A, \vdash_A)$ where:

- M_A is a set of *moves*,
- $\lambda_A : M_A \to \{O, P\} \times \{Q, A\}$ is a *labelling* function indicating whether a move is an Opponent or Player move, and whether it is a question (Q) or an answer (A). We write λ_A^{OP} for the projection of λ_A to $\{O, P\}$ and λ_A^{QA} for its projection on $\{Q, A\}$. $\overline{\lambda_A}$ will denote λ_A where the $\{O, P\}$ part has been reversed.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

- $I_A \subseteq \lambda_A^{-1}(OQ)$ is a set of *initial moves*.
- $\vdash_A \subset M_A \times M_A$ is a relation called *enabling*, satisfying:
  - If $m \vdash_A n \wedge \lambda_A^{QA}(n) = A$, then $\lambda_A^{QA}(m) = Q$;
  - If $m \vdash_A n$, then $\lambda_A^{OP}(m) \neq \lambda_A^{OP}(n)$.

In other terms, an arena is a directed bipartite graph, with a set of distinguished *initial* moves (necessarily Opponent questions) and a set of *answers* (m such that $\lambda_A^{QA} = A$) such that no answer points to another answer. We now define plays as *justified sequences* over A: these are sequences s of moves of A, each non-initial move m in s being equipped with a pointer to an earlier move n in s, satisfying $n \vdash_A m$. In other words, a justified sequence s over A is such that each reversed pointer chain $s_{\phi(0)} \leftarrow s_{\phi(1)} \leftarrow \cdots \leftarrow s_{\phi(n)}$ in s is a path on A (viewed as a directed bipartite graph).

The role of pointers is to allow *reopenings* in plays. Indeed, a path on A may be (slightly naively) understood as a linear play on A, and a justified sequence as an interleaving of paths, with possible duplications of some of them. This intuition is made precise in [21]. When writing justified sequences, we will often omit the justification information if this does not cause any ambiguity. $\sqsubseteq$ will denote the prefix ordering on justified sequences, and $s_1 \sqsubseteq^P s_2$ will mean that s_1 is a P-ending prefix of s_2. If s is a justified sequence on A, $|s|$ will denote its length.

Given a justified sequence s on A, it has two subsequences of particular interest: the P-view and O-view. The view for P (resp. O) may be understood as the subsequence of the play where P (resp. O) only sees his own duplications. In a P-view, O never points more than once to a given P-move, thus he must always point to the previous move. Concretely, P-views correspond to branches of Böhm trees [23]. Practically, the P-view $\ulcorner s \urcorner$ of s is computed by forgetting everything under Opponent's pointers, in the following recursive way:

- $\ulcorner sm \urcorner = \ulcorner s \urcorner m$ if $\lambda_A^{OP}(m) = P$;
- $\ulcorner sm \urcorner = m$ if $m \in I_A$ and m has no justification pointer;
- $\ulcorner s_1 m s_2 n \urcorner = \ulcorner s \urcorner mn$ if $\lambda_A^{OP}(n) = O$ and n points to m.

The O-view $\llcorner s \lrcorner$ of s is defined dually. The *legal sequences* over A, denoted by $\mathcal{L}_A$, are the justified sequences s on A satisfying the following conditions:

- *Alternation.* If $tmn \sqsubseteq s$, then $\lambda_A^{OP}(m) \neq \lambda_A^{OP}(n)$;
- *Bracketing.* A question q is *answered* by a if a is an answer and a points to q. A question q is *open* in s if it has not yet been answered. We require that each answer points to the *pending* question, *i.e.* the last open question.
- *Visibility.* If $tm \sqsubseteq s$ and m is not initial, then if $\lambda_A^{OP}(m) = P$ the justifier of m appears in $\ulcorner t \urcorner$, otherwise its justifier appears in $\llcorner t \lrcorner$.

Visibility and bracketing are flexible conditions, in the sense that they can either be required on plays or on strategies. It is in general considered more elegant to have them required on strategies rather than on plays, since it simplifies some of the proofs and does not unnecessarily restrict the behaviour of Opponent. However, it is sometimes convenient to have them required on plays for technical purposes, as is the case here. Anyway, both approaches build exactly the same category.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

4.1.2. The cartesian closed category of innocent strategies

A *strategy* σ on A is a non-empty even prefix closed set of even-length legal plays on A. A strategy is *deterministic* if only Opponent branches, *i.e.* $\forall smn, smn' \in \sigma,\ n = n'$. Of course, if A represents a type (or formula), there are often many more strategies on A than programs (or proofs) on this type. To address this issue we need *innocence*. An innocent strategy is a strategy σ such that

$$sab \in \sigma \wedge t \in \sigma \wedge ta \in \mathcal{L}_A \wedge \ulcorner sa \urcorner = \ulcorner ta \urcorner \implies tab \in \sigma.$$

Intuitively, an innocent strategy only takes its P-view into account to determine its next move. We now recall how arenas and innocent strategies organize themselves into a cartesian closed category. First, we build the *product* $A \times B$ of two arenas A and B:

$$\begin{aligned} M_{A\times B} &= M_A + M_B \\ \lambda_{A\times B} &= [\lambda_A, \lambda_B] \\ I_{A\times B} &= I_A + I_B \\ \vdash_{A\times B} &= \vdash_A + \vdash_B . \end{aligned}$$

We mention the empty arena $1 = (\emptyset, \emptyset, \emptyset, \emptyset)$, which will be terminal for the category of arenas and innocent strategies. We mention as well the arena $0 = (\bullet, \bullet \mapsto OQ, \{\bullet\}, \emptyset)$ with only one initial move, which will be a weak initial object. We define the *arrow* $A \Rightarrow B$ as follows:

$$\begin{aligned} M_{A\Rightarrow B} &= M_A + M_B \\ \lambda_{A\Rightarrow B} &= [\overline{\lambda_A}, \lambda_B] \\ I_{A\Rightarrow B} &= I_B \\ \vdash_{A\Rightarrow B} &= \vdash_A + \vdash_B + I_B \times I_A. \end{aligned}$$

We define composition of strategies by the usual parallel interaction plus hiding mechanism. If A, B and C are arenas, we define the set of *interactions* $I(A, B, C)$ as the set of justified sequences u over $(A \Rightarrow B) \Rightarrow C$ such that $u_{\restriction A,B} \in \mathcal{L}_{A\Rightarrow B}$, $u_{\restriction B,C} \in \mathcal{L}_{B\Rightarrow C}$ and $u_{\restriction A,C} \in \mathcal{L}_{A\Rightarrow C}$. Then, if $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$, we define parallel interaction:

$$\sigma \parallel \tau = \{u \in I(A, B, C) \mid u_{\restriction A,B} \in \sigma \wedge u_{\restriction B,C} \in \tau\}.$$

Composition is then defined as $\sigma;\tau = \{u_{\restriction A,C} \mid u \in \sigma||\tau\}$. It is associative and preserves innocence [23]. For each arena A there is an identity strategy, the copycat strategy defined by $id_A = \{s \in \mathcal{L}_{A_1\Rightarrow A_2} \mid \forall s' \sqsubseteq^P s,\ s'_{\restriction A_1} = s'_{\restriction A_2}\}$. Thus, there is a category **Inn** which has arenas as objects and innocent strategies on $A \Rightarrow B$ as morphisms from A to B. In fact, this category is cartesian closed, the cartesian structure given by the arena product above and the exponential closure given by

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

the arrow construction. This category is also equipped with a sum operation $A+B$, constructed as follows:

$$\begin{aligned} M_{A+B} &= M_A + M_B + \{q, L, R\} \\ \lambda_{A+B} &= [\lambda_A, \lambda_B, q \mapsto OQ, L \mapsto PA, R \mapsto PA] \\ I_{A+B} &= \{q\} \\ \vdash_{A+B} &= \vdash_A + \vdash_B + \{(q, L), (q, R)\} + \{L\} \times I_A + \{R\} \times I_B. \end{aligned}$$

Injections are just copycat strategies, and if $\sigma : A \to C$ and $\tau : B \to C$, the action of their copairing $[\sigma|\tau]$ can be described as follows: on any initial move on C, it immediately plays the initial move of $A+B$, then Opponent either chooses A or B. According to Opponent's choice, it either plays as σ or as τ. We can check easily that this sum defines a functorial weak coproduct, hence we have accomplished the first step in our construction of a category with strong types.

4.2. Open functors

It is very common in the literature to interpret types with free variables as functors, and often polymorphic terms as (di)natural transformations [5]. The construction of a functor from an open formula is naturally done by induction on the formula. However this approach is in conflict with the idea we defend here of building (parametrised) fixpoints by adding loops in the arenas, for which we need access to the intermediate concrete structure of open arenas. Of course, this raises the question of relating open arenas to their corresponding functors: this is the purpose of this section.

The intuition for open functors is really simple. Imagine you have an arena T with special moves (for simplicity, suppose for the moment that these are always Opponent moves) called *holes*, denoted by $\Box$. Replacing occurrences of this move in T by an arena A yields the arena $T[A]$. Suppose also that we have a strategy $\sigma : A \Rightarrow B$. We want to use it to define a strategy $T[\sigma] : T[A] \Rightarrow T[B]$, then show that this defines a functor (called the corresponding *open functor*). This is rather straightforward: it basically suffices to append plays of σ on plays of the identity on T. However in this paper we face a bigger challenge: we need to equip **Inn** with a structure of strong types, and thus to study the natural transformations between open functors. It turns out that they correspond to the similar notion of an *open natural transformation* $\eta_A : S[A] \to T[A]$, whose components are obtained by appending plays of id_A on plays of $\eta : S \Rightarrow T$.

In fact, both constructions are covered by a notion of *vertical composition*: given $\eta : S \Rightarrow T$ satisfying certain conditions and $\sigma : A \Rightarrow B$, we want to combine them into a strategy $\eta[\sigma] : S[A] \Rightarrow T[B]$. However, formalizing this construction is far more difficult than for the basic case of open functors. In fact, the restriction to pseudo-polynomial formulas comes from here: we could not carry out the construction in general, but only in the case where holes do not enable anything. Even in the pseudo-polynomial case, the details of this construction remain intricate but

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

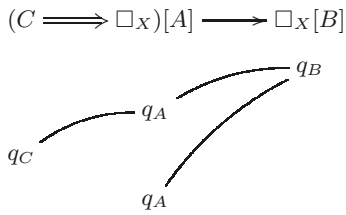


FIGURE 6. Dealing with non pseudo-polynomial open arenas.

once one has the right definitions, most of the proofs consist of long routine verifications (with the notable exception of Thm. 4.8) that seem mostly irrelevant for the purpose of this paper. Hence we will give here the main definitions and properties, but we will skip a lot of proofs. The complete details are already available in [10] and will be the subject of a future paper, hopefully overcoming the pseudo-polynomial restriction.

Before proceeding to these constructions, let us give a hint of the difficulties encountered in formalising vertical composition of strategies without the pseudo-polynomial restriction. As illustrated in Figure 6, once a play on $S[A] \Rightarrow T[B]$ reaches $A \Rightarrow B$, Opponent can interrupt it at any moment (even in a P-view) and return to $S \Rightarrow T$, causing (among other possible problems) the restriction to $A \Rightarrow B$ to be non-alternating. Dealing with this is probably possible, but requires a different approach to the construction of vertical composition.

4.2.1. Open arenas

Note that for all the arena constructions that we have considered, the new set of moves is obtained by taking the binary disjoint sum of the previous sets (except for the sum operation, where some new moves are added). So any move in an arena obtained by the interpretation of a formula must have the form $addr(a)$, where $addr$ is a sequence of *injections* (composed of in_1, in_2) and a is an *atom* in $\mathcal{A} = \{\circ, q, L, R\}$. We will restrict from now on to arenas where all moves necessarily have this shape, although we will generalise the set of atoms to allow for the definition of open arenas.

Let us fix a set $\mathcal{X}$ of variable names. An *open arena* is an arena in the usual sense, except that it is built on the set of atoms $\mathcal{A} \cup \{\Box_X \mid X \in \mathcal{X}\}$, where moves of the form $addr(\Box_X)$ are called *holes*, carrying both an *address* and a *variable name*. We add the constraint that no hole can justify anything (hence the restriction to pseudo-polynomial formulas):

$$\forall m, n \in M_A,\ m \vdash_A n \Rightarrow \forall X \in \mathcal{X}, \forall addr,\ m \neq addr(\Box_X).$$

An open arena A can have multiple holes, possibly annotated by different variables. Most of the time, we will be interested in only one free variable: we will then write $A[X]$ to emphasize that A is considered as an open arena with respect to X.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Definition 4.1 (Arena substitution)**.** Let A be an open arena with only one hole $addr(\square_X)$. If B is an arena we define $A[B/X]$ (or simply $A[B]$ when X is clear from the context) as follows.

$$M_{A[B]} = (M_A \setminus \{addr(\square_X)\}) \cup addr(M_B)$$

$$\lambda_{A[B]}(m) = \begin{cases} \lambda_A(m) & \text{if} \quad m \in M_A \\ \lambda_B(m') & \text{if} \quad m = addr(m') \text{ where } m' \in M_B \cap \lambda_B^{OP^{-1}}(\{O\}) \\ \overline{\lambda_B(m')} & \text{if} \quad m = addr(m') \quad \text{where} \quad m' \in M_B \cap \lambda_B^{OP^{-1}}(\{P\}) \end{cases}$$

$$I_{A[B]} = \begin{cases} I_A & \text{if} \quad addr(\square_X) \notin I_A \\ (I_A \setminus \{addr(\square_X)\}) \cup addr(I_B) & \text{otherwise} \end{cases}$$

$$m \vdash_{A[B]} n \Leftrightarrow \vee \begin{cases} m \vdash_A n \\ m' \vdash_B n' \wedge m = addr(m') \wedge n = addr(n') \\ m \vdash_A addr(\square_X) \wedge n = addr(n') \quad \text{where} \quad n' \in I_B. \end{cases}$$

Whenever $\square_Y$ does not appear in B we have $A[B/X][C/Y] = A[C/Y][B/X]$, hence the construction generalizes immediately to the simultaneous substitution of multiple occurrences of $\square_X$ by successive substitutions. Note that if an open arena S has both negative (Player) and positive (Opponent) occurrences of $\square_X$, we will use the notation $S[A, B]$ for the substitution of negative occurrences of $\square_X$ by A and positives occurrences by B.

4.2.2. Vertical composition

We will define both open functors and open natural transformations through the unifying notion of *vertical composition*[3]. Its definition requires us to be able to restrict plays on $S[A] \Rightarrow T[B]$ on $S \Rightarrow T$ and $A \Rightarrow B$. This restriction can be difficult to do in general because it might produce non-legal plays. However, it is well-behaved on the following set of plays.

Definition 4.2 (Functorial plays)**.** Let $C = S[B, A] \Rightarrow T[A, B]$ be an arena. A play $s \in \mathcal{L}_C$ is *functorial* for the decomposition $C = (S \Rightarrow T)[A, B]$ if it is of the form $s_1 s_2$, where s_1 only plays in S, T and s_2 starts with a move in B, then only plays in A, B. Moreover, all Player moves in s_2 which are initial in A (they cannot be initial in B) have the same justifier move. Likewise, all Opponent moves in s_2 which are initial in B (they cannot be initial in A) share the same justifier move.

Whenever it is clear from the context, we will speak of a functorial play without mentioning the decomposition for which it is functorial.

[3]Vertical composition, which as its name does not suggest, is related to *horizontal* composition in 2-categories. This mismatch follows from the fact that it seems more natural to consider usual composition of strategies to be horizontal.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Definition 4.3 (Vertical restriction)**.** Let $s \in \mathcal{L}_{S[B,A]\Rightarrow T[A,B]}$ be a functorial play. By definition it has the form $s_1 s_2$ described above. We define its *functorial restriction* $s_{\restriction S[X]\Rightarrow T[X]}$ as:

- s_1 if s_2 is empty;
- $s_1 \Box_X$ if s_2 only plays in B. Since s is functorial, all initial moves in B have the same justifier move and we can set it as be the justifier of $\Box_X$.
- $s_1 \Box_X \Box_X$ if s_2 reaches both A and B. The justifier moves of the occurrences of $\Box_X$ are defined as those of the initial moves in B and A. Note that even if s contains several moves that are initial in B, those will be replaced by only one occurrence of $\Box_X$.

Likewise, the *argument restriction* $s_{\restriction A\Rightarrow B}$ is s_2 where moves which are initial in B lose their pointers and moves which are initial in A have their pointer redirected to the unique initial move of B which is in their P-view.

Definition 4.4 (Natural strategies)**.** Let S, T be open arenas and $\eta : S \Rightarrow T$. We say that η is *natural in* X (or just *natural* if X is clear from the context) if η is innocent, and if for all $sab \in \eta$, a is an occurrence of $\Box_X$ if and only if b is an occurrence of $\Box_X$. Moreover, η *must* have a response to any occurrence of $\Box_X$.

Because of the unrestrained behaviour of Opponent, the strategies we aim to define by vertical composition can have non-functorial plays. However their P-views will always be functorial and we can exploit this to define vertical composition of innocent strategies as follows.

Definition 4.5 (Vertical composition)**.** Let $\eta : S \Rightarrow T$ be a natural strategy and $\sigma : A \Rightarrow B$ be innocent. Then we define the set of P-views of $\eta[\sigma]$ as:

$$\{s \in \mathcal{L}_{S[B,A]\Rightarrow T[A,B]} \mid s_{\restriction S[X]\Rightarrow T[X]} \in \ulcorner \eta \urcorner \wedge s_{\restriction A\Rightarrow B} \in \ulcorner \sigma \urcorner\}.$$

A direct verification ensures that this construction preserves innocence and totality. As for the case of arena substitution, if S, T has several types of holes $\Box_X$ and $\Box_Y$ and if η is natural both in X and Y, we disambiguate by specifying $\eta[\sigma/X]$ or $\eta[\sigma/Y]$.

Before moving to the actual construction of open functors (and open natural transformations), let us mention some algebraic properties of vertical composition.

Lemma 4.6. *If $\eta : S \Rightarrow T$, $\sigma : A \Rightarrow B$ and $\tau : C \Rightarrow D$, we have the following equations:*

- *Successive vertical compositions:* $(\eta[\sigma/X])[\tau/Y] = (\eta[\tau/Y])[\sigma[\tau/Y]/X]$.
- *Preservation of identities:* $id_T[id_A] = id_{T(A)}$.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

4.2.3. Open functors

The validity of both open functors and open natural transformations will rely on the following lemma:

Lemma 4.7 (Interchange law)**.** *Let S, T and U be open arenas. Let also $\eta : S \Rightarrow T$ and $\epsilon : T \Rightarrow U$ be natural strategies, and $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$ be innocent strategies. Then, if X appears only positively (Opponent move) in S, T and U:*

$$(\eta; \epsilon)[\sigma; \tau] = \eta[\sigma]; \epsilon[\tau].$$

Likewise, if X appears only negatively (Player move) in S, T and U:

$$(\eta; \epsilon)[\sigma; \tau] = \eta[\tau]; \epsilon[\sigma].$$

Proof idea. As usual when relating static properties of strategies with composition, we generalize the notion of functorial play to functorial interactions, and the notions of vertical restriction to these functorial interactions. The interchange laws then follow from the fact that vertical restrictions on functorial interactions coincide with vertical restrictions on functorial plays.

As an illustration, consider the (natural) strategy $\texttt{swap} : \square_X + \square_X \Rightarrow \square_X + \square_X$ which swaps the two components, and the negation $\texttt{neg} : \texttt{bool} \Rightarrow \texttt{bool}$. Figure 7 presents an interaction u between $\texttt{swap}[\texttt{neg}]$ and itself. However it can also be regarded as an interaction between $\texttt{swap}$ and itself appended with an interaction between $\texttt{neg}$ and itself (identifying occurrences of $\square_X$ and initial moves in $\texttt{bool}$). Hence, applying the hiding operation yields a play in $(\texttt{swap}; \texttt{swap})[\texttt{neg}; \texttt{neg}]$. Therefore, u witnesses the fact that $\texttt{swap}[\texttt{neg}]; \texttt{swap}[\texttt{neg}] = (\texttt{swap}; \texttt{swap})[\texttt{neg}; \texttt{neg}]$. □

Open functors. By the interchange law and the equations of Lemma 4.6, any positive[4] open arena S gives rise to an *open functor* $S : \mathbf{Inn} \to \mathbf{Inn}$ defined by $S(\sigma) = (id_S)[\sigma]$. Any negative open arena defines similarly an open functor $S : \mathbf{Inn}^{op} \to \mathbf{Inn}$ by the negative interchange law, and an open arena with mixed variance defines an open functor $S : \mathbf{Inn}^{op} \times \mathbf{Inn} \to \mathbf{Inn}$, defined by $S(\sigma, \tau) = (id_S)[\sigma/X^-][\tau/X^+]$. Open functors generalize then to open arenas with multiple variable names by the first equation of Lemma 4.6.

Open natural transformations. When S and T have the same fixed variance, any $\eta : S \Rightarrow T$ gives rise to an *open natural transformation* $\eta : S \overset{\bullet}{\to} T$ defined for all arena A by $\eta_A = \eta[id_A]$.

Factorisation. Open functors and open natural transformation enjoy very rich properties. In particular, we have the following powerful factorisation theorem.

Theorem 4.8 (Factorisation)**.** *Let S, T be open arenas with the same fixed variance, and let $\eta : S \overset{\bullet}{\to} T$ be a natural transformation. Then, for all A:*

$$\eta_A = \eta_{\square_X}[id_A].$$

[4]An open arena is *positive* when all occurrences of $\square_X$ are Opponent moves. Likewise, it is *negative* if all occurrences of $\square_X$ are Player moves.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

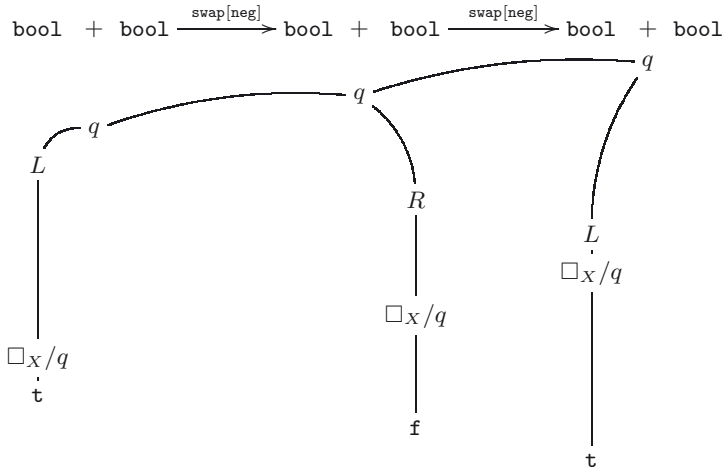


FIGURE 7. Interactions and vertical composition.

In other words, every natural transformation between open functors is an open natural transformation.

Proof idea. The proof [10] is rather intricate, and proceeds in two parts:

(1) First, we prove that any natural transformation $\iota : Id \xrightarrow{\bullet} Id$ from the identity functor to itself is the identity natural transformation. This is rather easy to prove by induction for finite, acyclic open arenas, then we generalise to infinite arenas by taking the limit and to cyclic ones by relating them to their acyclic infinite expansion.
(2) Then, given any natural transformation $\eta : S \xrightarrow{\bullet} T$ between open functors, we prove that for all A, the strategy η_A always restricts on $S \Rightarrow T$ to the same natural strategy $\eta_{\Box_X} : S \Rightarrow T$, and we apply the argument of (1) to demonstrate that its restriction to $A \Rightarrow A$ must always be the identity. □

Note that even (1) is not necessarily true in an arbitrary category: see for example the case of ribbon categories [24] where the twisting of a ribbon is a polymorphic operation from the identity to itself. Let us also mention that the proof is sufficiently generic to apply without any change to the category of games and winning strategies to be introduced in the next section.

4.2.4. Dinatural case

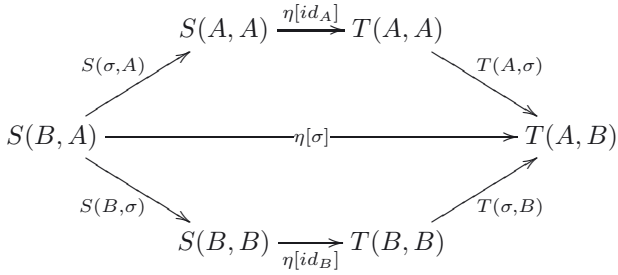
Let us now analyse the case of vertical composition where the open arenas S and T may have mixed variance, *i.e.* occurrences of $\Box_X$ both as Player and Opponent moves. In this case, we prove that natural strategies give rise to dinatural transformation. The main lemma is the following dinatural variation of the interchange law.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Lemma 4.9 (Dinatural interchange)**.** *Let $\eta : S \Rightarrow T$ be a natural strategy, and let $\sigma : A \Rightarrow B$, $\delta : B \Rightarrow C$ and $\tau : C \Rightarrow D$ innocent strategies. Then:*

$$\eta[\sigma; \delta; \tau] = S(\tau, \sigma); \eta[\delta]; T(\sigma, \tau).$$

This lemma immediately allows to deduce that natural strategies give rise to dinatural transformations. To prove that the family $(\eta[id_A])_A$ is dinatural in A, one requires two applications of the dinatural interchange law, as made explicit by the diagram below.



The fact that we get dinatural transformations is enough to provide to **Inn** a strong (pseudo-polynomial) types structure, however let us mention that unlike the general case, the composition of two open dinatural transformations is still an open dinatural transformation. Hence, it gives some hope to get an elegant model of polymorphism, especially if we manage to prove a factorisation theorem for dinatural transformations. Of course, such a model would also require us first to generalize the construction of open functors to all open arenas.

4.2.5. Strong types

Let $\mathcal{F}$ be the set of open functors in **Inn**. It is a simple verification to show that $\mathcal{F}$ contains all usual functors. The identity is generated by the arena $\square_X$, constant functors are generated by arenas without free variables, product is generated by $\square_X \times \square_Y$, sum is generated by $\square_X + \square_Y$, and exponentiation by $\square_X \Rightarrow A$ for every arena A[5]. It is also easy to check that open functors are stable under composition (substitution of open arenas) and by contraction (identify $\square_X$ and $\square_Y$).

Strengths. For each covariant open functor $T : \mathbf{Inn} \to \mathbf{Inn}$, we define a strategy $\theta^T : \square_X \times T(\square_Y) \to T(\square_X \times \square_Y)$ natural in X and Y as follows:

$$\theta^T = \{s \in \mathcal{L}_{\square_X \times T(\square_Y) \Rightarrow T(\square_X \times \square_Y)} \mid \forall s' \sqsubseteq^P s,\ s'_{\restriction L} \simeq s'_{\restriction R}\}.$$

Where by $s \simeq s'$, we mean that they are equal up to the pointers of occurrences of $\square_X$. The various laws for the algebraic manipulation of open natural transformations suffice to prove that this defines a strength.

[5]We do not have an open functor for $- \Rightarrow -$ due to the enabling restriction on holes.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Contrastrengths. For each contravariant functor $N : \mathbf{Inn}^{op} \to \mathbf{Inn}$, we define a strategy $\rho^N : \Box_X \times N(\Box_X \times \Box_Y) \to N(\Box_Y)$ by:

$$\rho^N = id_{N(\Box_Y)} \cup \{s\Box_X\Box_X \in \mathcal{L}_{\Box_X \times N(\Box_X \times \Box_Y) \to N(\Box_Y)} \mid s \in id_{N(\Box_Y)}\}.$$

In other words, ρ^N behaves as the identity on $N(\Box_Y)$ until Opponent plays $\Box_X$, in which cases it responds by the other occurrence of $\Box_X$. As above, the available algebraic manipulations on vertical composition make it easy to check that this is a contrastrength.

Uniformity. Thus open functors are always strong, it remains to show that these strengths satisfy the required uniformity conditions.

Theorem 4.10. *The category* **Inn** *has strong (pseudo-polynomial) types.*

Proof. The required equalities follow from the properties of vertical composition.

- *Naturality in T and N.* Let us detail the covariant case. Suppose we have two open functors $T_1, T_2 : \mathbf{Inn} \to \mathbf{Inn}$ and a natural transformation $\eta : T_1 \stackrel{\bullet}{\to} T_2$. The naturality of θ in T means that for all Γ, A the following diagram must commute.

$$\begin{array}{ccc} \Gamma \times T_1(A) & \xrightarrow{\theta^{T_1}_{\Gamma,A}} & T_1(\Gamma \times A) \\ {\scriptstyle \Gamma \times \eta_A}\downarrow & & \downarrow{\scriptstyle \eta_{\Gamma \times A}} \\ \Gamma \times T_2(A) & \xrightarrow[\theta^{T_2}_{\Gamma,A}]{} & T_2(\Gamma \times A) \end{array}$$

  By the factorisation theorem, we have $\eta_A = \eta[id_A]$ where $\eta = \eta_{\Box_X}$. Thus if we can prove the equality for $\Gamma = \Box_X$ and $A = \Box_Y$, the general case will follow by the algebraic manipulations on vertical composition. But this particular case is obvious by definition of the strategies θ^{T_1} and θ^{T_2}.
- *Compatibility with identity and constant functors.* Straightforward.
- *Compatibility with contraction and composition.* As above, we first prove the necessary equalities for $\Gamma = \Box_X$ and $A = \Box_Y$, then conclude by the algebraic manipulations on vertical composition.
- *Compatibility with sums.* The family $\gamma_{\Gamma,A,A} : \Gamma \times (A + A) \to \Gamma \times A + \Gamma \times A$ is natural in Γ and A, so it is an open natural transformation by the factorisation theorem. So once again the necessary equality boils down to the special case where $\Gamma = \Box_X$ and $A = \Box_Y$ and algebraic reasoning on vertical composition.
- *Compatibility with the closed structure.* Same as above, except that $\rho^{-\Rightarrow C}_{\Gamma,A}$ is dinatural (and not natural) in Γ, hence the factorisation theorem does not apply. So, we apply instead the reasoning above to $\Lambda(\rho^{-\Rightarrow C}_{\Gamma,A})$, which is natural both in Γ and A. □

4.2.6. Loops and recursive types

Before we move on to the last section of this paper, let us mention how this work allows us to revisit McCusker's model of recursive types. We recall the inclusion

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

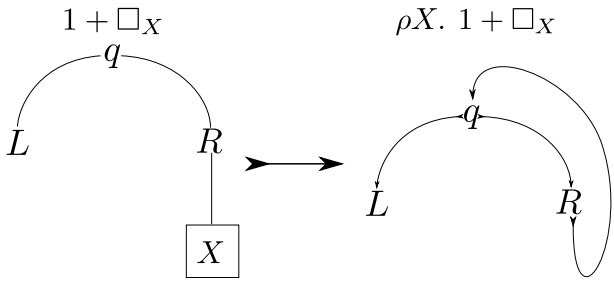


FIGURE 8. Loop construction.

order $A \trianglelefteq B$ on arenas, defined by:

$$\begin{aligned} M_A &\subseteq M_B \\ \lambda_A &= \lambda_{B \restriction M_A} \\ I_A &= I_B \cap M_A \\ \vdash_A &= \vdash_B \cap\, M_A^2. \end{aligned}$$

The set of arenas, ordered by $\trianglelefteq$, is a cpo. As for strong functors, it is easy to prove that open functors are continuous and preserve injection and projection strategies. Moreover, they are *closed* in the sense of McCusker, *i.e.* they are enriched functors w.r.t. the enrichment of **Inn** over itself given by the closed structure. Hence, for any open functor $T : \mathbf{Inn} \to \mathbf{Inn}$, McCusker's theorem [31] states that the following infinite iteration of T defines a minimal invariant [17–19] for T, hence an adequate model of recursive types.

$$T^\omega = \coprod_{i=0}^{\infty} T^i(1).$$

The novelty here is that we have access to the intermediate step of the open arena, which permits the following construction:

Definition 4.11 (Loop construction)**.** Let T be an open arena. For each name X such that T is *strict* in X (no hole $\square_X$ is initial), we define a new arena $\rho X.T$:

$$\begin{aligned} M_{\rho X.T} &= M_T \setminus \{addr(\square_X) \mid addr \in \mathcal{A}\} \\ \lambda_{\rho X.T} &= \lambda_T \restriction M_{\rho X.T} \\ m \vdash_{\rho X.T} n \Leftrightarrow &\vee \begin{cases} m \vdash_T n \\ \exists addr \in \mathcal{A},\ m \vdash_T addr(\square_X) \wedge n \in I_T \end{cases} \\ I_{\rho X.T} &= I_T. \end{aligned}$$

The loop construction has the effect of generating cycles in arenas, as illustrated in Figure 8. Even if the arenas T^ω and $\rho X.T$ are distinct, they have the same paths thus the same legal plays. More formally, there is a straightforward isomorphism

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

ϕ_T: $\mathcal{L}_{\rho X.T} \to \mathcal{L}_{T^\omega}$ which to any play in $\mathcal{L}_{\rho X.T}$ associates a play in $\mathcal{L}_{T^\omega}$ just by renaming moves. By Laurent's theorem on isomorphisms [27], ϕ_T corresponds canonically to an isomorphism in **Inn** (for which, by abuse of notation, we will keep the same notation) $\phi_T : \rho X.T \Rightarrow T^\omega$, which simply behaves as the identity. The immediate consequence is that since T^ω is a minimal invariant for T, this is also the case for $\rho X.T$, hence loops in arenas indeed model recursive types.

Recursive types are a feature of partial (*i.e.* not all programs terminate) programming languages, whereas inductive and coinductive types are a feature of total programming languages. To be able to model programs as total strategies and ban the non-terminating aspects of recursive types, we will now enrich the loops described above with *winning conditions*.

4.3. Induction and coinduction in games

Here comes the final step of the model construction, which consists of ensuring preservation of totality by composition. In the case of acyclic, either finite or well-founded arenas, a general solution of this problem can be found [11]. However when arenas are possibly infinite or cyclic, it requires the notion of *winning*. To the author's knowledge, the use of winning in game semantics originates in the early work of Blass [7]. It has subsequently been used several times for the purpose of ensuring finiteness of interactions, see [2, 22] for an introduction. The precise formulation that we use here is inspired from [22]. In this section, we use this notion of winning familiar to game semanticists and we mix it with the usual winning conditions for least and greatest fixpoints in parity games [35].

4.3.1. Winning games

For the purpose of this section, we need to temporarily allow infinite plays, hence we take the following conventions. If A is an arena, let us denote by $\mathcal{L}_A^\omega$ the set of possibly infinite legal plays over A and $\mathcal{L}_A^\infty$ the set of *infinite* legal plays. Likewise, let us denote by Th_A^ω and Th_A^∞ the sets of, respectively, possibly infinite *threads*[6] and actually infinite threads. If $s \in \mathcal{L}_A^\omega$, we will need to isolate its threads. If i is the index of an initial move in s, let us denote by $s_{\restriction i}$ the subplay of s having s_i as unique initial move: $s_{\restriction i}$ can be seen as the equivalence class of s_i for the reflexive, symmetric and transitive closure of the justification relation. We also then define the set of *threads* of $s \in \mathcal{L}_A^\omega$ by the set $[\![s]\!]$ of all plays $s_{\restriction i}$ such that s_i has no justification pointer in s.

Definition 4.12. Let A be an arena. A *winning function* on A is a function

$$G : \mathcal{L}_A^\omega \to \{W, L\}$$

stating if a play s is a Win or a Loss for Player, and satisfying the following properties:

(1) $\forall s \in \mathcal{L}_A^\omega,\ G(s) = W \Leftrightarrow \forall t \in [\![s]\!],\ G(t) = W$;

[6] Let us recall that a *thread* is a legal play with only one initial move.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

(2) If s is finite and of even length, $G(s) = W$;
(3) If s is finite and of odd length, $G(s) = L$.

A *win-game* $\mathcal{A}$ is a pair $(A, G_{\mathcal{A}})$, where A is an arena and $G_{\mathcal{A}}$ is a winning function on A.

We then extend all the arena constructions to win-games. There is only one way to extend the base arenas 0 and 1 to win-games $\mathbf{0}$ and $\mathbf{1}$, since they admit only finite plays. Let us suppose that $\mathcal{A}$ and $\mathcal{B}$ are win-games. We define $G_{\mathcal{A}\times\mathcal{B}}$, $G_{\mathcal{A}+\mathcal{B}}$ and $G_{\mathcal{A}\Rightarrow\mathcal{B}}$ by associating with them the following winning functions.

- If $s \in \mathcal{L}^{\omega}_{A\times B}$, $G_{\mathcal{A}\times\mathcal{B}}(s) = W$ if and only if $G_{\mathcal{A}}(s_{\restriction A}) = W$ and $G_{\mathcal{B}}(s_{\restriction B}) = W$.
- If $s \in \mathcal{L}^{\omega}_{A+B}$, then by well-bracketing each thread $t \in [\![s]\!]$ is either of the form qLs' where $s' \in \mathcal{L}^{\omega}_{A}$ or in qRs' where $s' \in \mathcal{L}^{\omega}_{B}$. We say that $G_{\mathcal{A}+\mathcal{B}}(s) = W$ if each of its threads is winning for $G_{\mathcal{A}}$ and $G_{\mathcal{B}}$, in the corresponding component.
- If $s \in \mathcal{L}^{\omega}_{A\Rightarrow B}$, then $G_{\mathcal{A}\Rightarrow\mathcal{B}}(s) = W$ if and only if for each thread $t \in [\![s]\!]$, if $G_{\mathcal{A}}(t_{\restriction A}) = W$, then $G_{\mathcal{B}}(t_{\restriction B}) = W$.

It is simple to check that these constructions satisfy all the requirements for a winning function. To prove that winning strategies are stable under composition, we will need the following *modus ponens* lemma:

Lemma 4.13. *Let $\mathcal{A} = (A, G_{\mathcal{A}})$ and $\mathcal{B} = (B, G_{\mathcal{B}})$ be two win-games, and $s \in \mathcal{L}^{\infty}_{A\Rightarrow B}$. Suppose that $G_{\mathcal{A}\Rightarrow\mathcal{B}}(s) = W$. Then we have: if $G_{\mathcal{A}}(s_{\restriction A}) = W$, then $G_{\mathcal{B}}(s_{\restriction B}) = W$.*

Proof. We suppose that $G_{\mathcal{A}\Rightarrow\mathcal{B}}(s) = W$ and $G_{\mathcal{A}}(s_{\restriction A}) = W$. Let us take an arbitrary $t \in [\![s_{\restriction B}]\!]$. Then there is an initial move i of $s_{\restriction B}$ such that $t = (s_{\restriction B})_{\restriction i}$. This move $(s_{\restriction B})_i$ must also be in s, with a possibly different index i', thus we have $(s_{\restriction B})_{\restriction i} = (s_{\restriction i'})_{\restriction B}$. Thus:

$$[\![(s_{\restriction i'})_{\restriction A}]\!] \subset [\![s_{\restriction A}]\!] \subset G_{\mathcal{A}}^{-1}(\{W\})$$

The last inclusion being true because s is supposed to be winning on $\mathcal{A}$. But as a thread of winning s, $s_{\restriction i'}$ must be winning, hence by definition of winning on $\mathcal{A} \Rightarrow \mathcal{B}$ we have $(s_{\restriction i'})_{\restriction B}$ winning on $\mathcal{B}$, but $(s_{\restriction i'})_{\restriction B} = (s_{\restriction B})_{\restriction i} = t$ and t is winning. □

Winning strategies. We will now define the category **Win** of win-game and total winning strategies. If $\sigma : A$ is a strategy, we define $\sigma^{\omega} = \{s \in \mathcal{L}^{\omega}_{A} \mid \forall s' \sqsubseteq^{P} s,\ s' \in \sigma\}$ the set of possibly infinite plays of σ.

Definition 4.14. Let $\mathcal{A} = (A, G_{\mathcal{A}})$ be a win-game and $\sigma : A$ a strategy. We say that σ is *winning* on $\mathcal{A}$ if for all $s \in \sigma^{\omega}$, $G_{\mathcal{A}}(s) = W$.

Proposition 4.15. *Let $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$ be win-games. If $\sigma : \mathcal{A} \Rightarrow \mathcal{B}$ and $\tau : \mathcal{B} \Rightarrow \mathcal{C}$ are winning strategies, then so is $\sigma;\tau : \mathcal{A} \Rightarrow \mathcal{C}$.*

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Proof. It suffices to prove that all threads of $(\sigma;\tau)^\omega$ are winning, so let t be such a thread. If t is finite it has even length (because $t \in \sigma$) and we must have $G_{\mathcal{A}\Rightarrow\mathcal{C}}(t) = W$ by definition of winning functions. If t is infinite, it has an infinite interaction witness $u \in \sigma \parallel \tau$. By definition of $G_{\mathcal{A}\Rightarrow\mathcal{C}}$, we have to prove that if $G_{\mathcal{A}}(t_{\uparrow A}) = W$, then $G_{\mathcal{C}}(t_{\uparrow C}) = W$. Let us suppose that $G_{\mathcal{A}}(t_{\uparrow A}) = W$, or equivalently $G_{\mathcal{A}}(u_{\uparrow A}) = W$. Since σ is winning, we can apply Lemma 4.13 and deduce that $G_{\mathcal{B}}(u_{\uparrow B}) = W$. Thus, again by Lemma 4.13 and the fact that τ is winning, we have $G_{\mathcal{C}}(u_{\uparrow C}) = W$, *i.e.* $G_{\mathcal{C}}(t_{\uparrow C}) = W$ which concludes the proof. □

Proposition 4.16. *Let $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$ be win-games. If $\sigma : \mathcal{A} \Rightarrow \mathcal{B}$ and $\tau : \mathcal{B} \Rightarrow \mathcal{C}$ are winning and total, then $\sigma;\tau : \mathcal{A} \Rightarrow \mathcal{C}$ is total.*

Proof. It is known that if total $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$ yield by composition a partial $\sigma;\tau : A \Rightarrow C$, it implies the existence of an *infinite chattering*, *i.e.* an interaction $u \in \sigma \parallel \tau$ which is finite in A and C, but infinite in B. See for example [11], where this is explained thoroughly. A simple analysis of polarities in the interaction show that we necessarily have $u_{\uparrow A}$ even-length and $u_{\uparrow C}$ odd-length, hence $G_{\mathcal{A}}(u_{\uparrow A}) = W$ and $G_{\mathcal{C}}(u_{\uparrow C}) = L$. But σ is winning, hence because of Lemma 4.13 we must have $G_{\mathcal{B}}(s_{\uparrow B}) = W$. And τ is winning, thus because of Lemma 4.13 we must have $G_{\mathcal{C}}(s_{\uparrow C}) = W$: contradiction. Thus no such infinite interaction is possible, and $\sigma;\tau$ is total. □

Thus we have a category **Win** of win-games and total winning strategies. It is straightforward to verify that the cartesian closed structure and the functorial weak coproducts of **Inn** extend immediately to **Win**, hence we still have a cartesian closed category with functorial weak coproducts. Note also that the cpo on arenas can be extended to a cpo on win-games by setting $\mathcal{A} \trianglelefteq \mathcal{B}$ if and only if $A \trianglelefteq B$ and if for all $s \in \mathcal{L}_A$, $G_{\mathcal{A}}(s) = G_{\mathcal{B}}(s)$.

4.3.2. Strong types

To prepare the construction of a μ-closed category, we must extend the notion of open functor to the category **Win**. The basic idea is that an open win-functor must be a functor $\mathcal{T} : \mathbf{Win} \to \mathbf{Win}$ whose action on arenas is given by an underlying open functor $T : \mathbf{Inn} \to \mathbf{Inn}$. We also want to ensure that $\mathcal{T}$ is strong, for which we simply require that the strength of T given by the structure of strong types of **Inn** must be winning.

Definition 4.17. An *open win-functor* is a functor

$$\mathcal{T} : (\mathbf{Win}^{op} \times \mathbf{Win})^n \to \mathbf{Win}$$

which is continuous for $\trianglelefteq$ in all its arguments and whose action on arenas is given by an open functor T. Moreover, we require that $\mathcal{T}$ is strong in each of its arguments. More precisely, for each factorisation $(\mathbf{Win}^{op} \times \mathbf{Win})^n = \mathfrak{C} \times \mathfrak{D}$ where $\mathfrak{C} = \mathbf{Win}$ or $\mathfrak{C} = \mathbf{Win}^{op}$, and where $\mathfrak{D}$ is a product of copies of **Win** and $\mathbf{Win}^{op}$, for any object D of $\mathcal{D}$, the strength $\theta_{\Gamma,A}^{T(-,D)}$ (or contrastrength $\rho_{\Gamma,A}^{T(-,D)}$,

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

if $\mathfrak{C} = \mathbf{Win}^{op}$) given by the structure of strong types in **Inn** is winning for all Γ and $\mathcal{A}$ in **Win**.

Proposition 4.18. *The category* **Win** *equipped with the class* $\mathcal{F}$ *of open win-functors has a structure of strong (pseudo-polynomial) types.*

Proof. It is easy to verify that $\mathcal{F}$ contains the necessary functors, and all equations follow from strong types in **Inn**. Naturality of $\theta^T_{\Gamma,A}$ and $\rho^N_{\Gamma,A}$ is a bit subtle to prove, because it requires us first to generalize the factorisation theorem (Thm. 4.8) to **Win**. However, it works easily because the proof only exploits commutation with morphisms of the cartesian closed structure, hence total winning strategies, see [10] for more details. □

4.3.3. Winning for loops

Let us now consider an arena $\rho X.T$ built by a loop construction, and let $s \in \mathcal{L}^\omega_{\rho X.T}$. To decide which player wins s, we must first of all look at the *depth* of s, which is the number of times the loop has been taken. Let us recall the isomorphism ϕ_T, which associates to each play on $\rho X.T$ the corresponding play $\phi_T(s)$ on T^ω. The *depth* of s is, if it exists, the smallest $n \in \mathbb{N}$ such that $\phi_T(s) \in \mathcal{L}^\omega_{T^n(1)}$. If there is no such integer we say that s has *infinite* depth.

Definition 4.19 (Least fixpoint)**.** Let $\mathcal{T} : \mathbf{Win} \to \mathbf{Win}$ be an open win-functor such that the open arena T is strict. We define a new win-game $\mu X.\mathcal{T}$ as follows. Its underlying arena is $\rho X.T$, and if $s \in \mathcal{L}^\omega_{\rho X.T}$ we define its winning function G as follows: If s has infinite depth, then $G(s) = L$. Otherwise let n be its depth, and set $G(s) = G_{\mathcal{T}^n(\mathbf{1})}(\phi_T(s))$.

Definition 4.20 (Greatest fixpoint)**.** Let $\mathcal{T} : \mathbf{Win} \to \mathbf{Win}$ be an open win-functor such that the open arena T is strict. We define a new win-game $\nu X.\mathcal{T}$ as follows. Its underlying arena is $\rho X.T$, and if $s \in \mathcal{L}^\omega_{\rho X.T}$ we define its winning function G as follows: If s has infinite depth, then $G(s) = W$. Otherwise let n be its depth, and set $G(s) = G_{\mathcal{T}^n(\mathbf{1})}(\phi_T(s))$.

Remark. In other words if a play $s \in \mathcal{L}^\omega_{\rho X.T}$ takes the external loop an infinite number of times, it is immediately losing on $\mu X.\mathcal{T}$ and winning on $\nu X.\mathcal{T}$. However if it only takes this loop a finite number of times (but may yet be infinite), the winner on both games is decided by considering it on a finite expansion $\mathcal{T}^n(1)$. Since $\mathcal{T}$ is continuous the number n does not matter, as long as it is higher than the depth of s. As hinted by the use of this finite expansion, the use of loops is in fact not mandatory: one could define $\mu X.\mathcal{T}$ and $\nu X.\mathcal{T}$ respectively as least and greatest fixpoints of $\mathcal{T}$ with respect to $\trianglelefteq$ and get isomorphic games. The interest of loops resides in the fact that they preserve finiteness of arenas, and allow us to present our model as a conservative extension of parity games. Note however that whereas in parity games the winning conditions can be expressed concisely by assigning numerical *priorities* to moves (expressing the level of imbrication of loops), the presence of branching in plays unfortunately makes this representation

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

inapplicable here. The problem of finding a finite representation of those winning conditions (and a completeness result at the level of games) remains open.

There is a strategy $\chi_T : T(\rho X.T) \Rightarrow \rho X.T$ in **Inn** generated by the obvious path isomorphism between $T(\rho X.T)$ and $\rho X.T$. This isomorphism satisfies that $\chi_{\mathcal{T}} : \mathcal{T}(\mu X.\mathcal{T}) \Rightarrow \mu X.\mathcal{T}$ and $\chi_{\mathcal{T}}^{-1} : \nu X.\mathcal{T} \Rightarrow \mathcal{T}(\nu X.\mathcal{T})$ are both winning strategies, as follows easily from the continuity hypothesis on $\mathcal{T}$.

Proposition 4.21 (Initial algebras)**.** *Let $\mathcal{T}$: **Win** $\to$ **Win** be an open winfunctor with a strict underlying open arena. Then $(\mu X.\mathcal{T}, \chi_{\mathcal{T}})$ is an initial algebra for $\mathcal{T}$.*

Proof. Let $(\mathcal{B}, \sigma)$ be another algebra for $\mathcal{T}$. We have to check that there is an unique total winning strategy $\sigma^\dagger : \mu X.\mathcal{T} \Rightarrow \mathcal{B}$ such that

$$\begin{array}{ccc} \mathcal{T}(\mu X.\mathcal{T}) & \xrightarrow{\mathcal{T}(\sigma^\dagger)} & \mathcal{T}(\mathcal{B}) \\ {\scriptstyle \chi_{\mathcal{T}}}\downarrow & & \downarrow{\scriptstyle \sigma} \\ \mu X.\mathcal{T} & \xrightarrow[\sigma^\dagger]{} & \mathcal{B} \end{array}$$

commutes. Intuitively, we only want to iterate σ:

$$\cdots \xrightarrow{\mathcal{T}^3(\sigma)} \mathcal{T}^3(\mathcal{B}) \xrightarrow{\mathcal{T}^2(\sigma)} \mathcal{T}^2(\mathcal{B}) \xrightarrow{\mathcal{T}(\sigma)} \mathcal{T}(\mathcal{B}) \xrightarrow{\sigma} \mathcal{B}$$

and somehow take the limit. However, no general property of **Win** directly ensures that this limit exists. We can however build it by using the cpo-enriched structure of **Inn**. Let us begin by defining the partial iterations of σ by:

$$\begin{aligned} \sigma^{(1)} &= \sigma \\ \sigma^{(n+1)} &= \mathcal{T}^n(\sigma); \sigma^{(n)} \end{aligned}$$

In general, we do not have $\sigma^{(n)} \subseteq \sigma^{(n+1)}$, which prevents us from concluding immediately. However, we do have for all n:

$$\sigma^{(n)}_{\restriction T^\omega \Rightarrow B} \subseteq \sigma^{(n+1)}_{\restriction T^\omega \Rightarrow B}$$

And all the strategies of this chain are (possibly partial) strategies on $T^\omega \Rightarrow B$. We can then define, by completeness of the inclusion order on strategies in **Inn**:

$$\sigma^{(\omega)} = \bigcup_{i \in \mathbb{N}_*} \sigma^{(i)}_{\restriction T^\omega \Rightarrow B}.$$

This strategy $\sigma^{(\omega)}$ is the one we are looking for, but we have to bring it back to the arena $\rho X.T \Rightarrow B$. For this purpose, note that that we have a straightforward play isomorphism $\phi' : \mathcal{L}^\omega_{\rho X.T \Rightarrow B} \to \mathcal{L}^\omega_{T^\omega \Rightarrow B}$, thus we can define $\sigma^\dagger = \phi'^{-1}(\sigma^{(\omega)})$. Clearly $\sigma^\dagger$ is innocent, since it is in direct correspondence with innocent $\sigma^{(\omega)}$.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

The strategy $\sigma^\dagger$ is also total; it suffices to prove it for $\sigma^{(\omega)}$. Let $s \in \sigma^{(\omega)}$ and $sa \in \mathcal{L}_{T^\omega \Rightarrow B}$. We want to prove that $\sigma^{(\omega)}$ has some response to a. Since s is finite, there is some $n \in \mathbb{N}$ such that $s \in \sigma^{(n)}_{\restriction T^\omega \Rightarrow B}$. As a composition of total strategies $\sigma^{(n)}$ is total, but on $T^n(B) \Rightarrow B$, and maybe the response of $\sigma^{(n)}$ to a would have been in the left copy of B. But we also have $s \in \sigma^{(n+1)}_{\restriction T^\omega \Rightarrow B}$ where $\sigma^{(n+1)}$ is also total so there must be some move b such that $sab \in \sigma^{(n+1)}$. But then $sab \in \sigma^{(n+1)}_{\restriction T^\omega \Rightarrow B}$, because T is strict and b cannot reach directly the left occurrence of B. Thus $sab \in \sigma^{(\omega)}$ and $\sigma^\dagger$ is total.

Let us also prove that $\sigma^\dagger$ is winning. Let t be a possibly infinite thread in $\sigma^\dagger$ such that $t' = t_{\restriction \rho X.T}$ is winning on $\mu X.\mathcal{T}$. By definition of winning on $\mu X.\mathcal{T}$ this means that t' has a finite depth n, and that $\phi_T(t')$ is winning on $\mathcal{T}^n(\mathbf{1})$. But in this case, we have in fact $\phi'(t) \in \sigma^{(n)}$ and $\sigma^{(n)}$ is winning as the composition of winning strategies, thus t is winning.

Finally, we have to prove that there is no other strategy $f : \mu X.\mathcal{T} \Rightarrow \mathcal{B}$ such that the above diagram commutes. Let us suppose that there is such an f, then we would have $\mathcal{T}(f); \sigma = \chi_{\mathcal{T}}; f$. In other words:

$$f = \chi_{\mathcal{T}}^{-1}; \mathcal{T}(f); \sigma$$

By abuse of notation, let us also denote by ϕ the isomorphism in $\mathbf{Inn}$ $\phi : \rho X.T \to T^\omega$ generated by the play isomorphism $\phi : \mathcal{L}^\omega_{\rho X.T} \to \mathcal{L}^\omega_{T^\omega}$. Then we have:

$$\begin{aligned} \phi^{-1}; f &= \phi^{-1}; \chi_{\mathcal{T}}^{-1}; \mathcal{T}(f); \sigma \\ &= \mathcal{T}(\phi^{-1}; f); \sigma. \end{aligned}$$

We use the equality $\phi^{-1}; \chi_{\mathcal{T}}^{-1} = \mathcal{T}(\phi^{-1})$, which is trivial by definition of these strategies. After n substitutions of this expression in itself, we get:

$$\phi^{-1}; f = \mathcal{T}^{n+1}(\phi^{-1}; f); \mathcal{T}^n(\sigma); \ldots; \mathcal{T}(\sigma); \sigma.$$

Thus for all $n \in \mathbb{N}$, we have $\mathcal{T}^{n+1}(\phi^{-1}; f); \sigma^{(n)} = \phi^{-1}; f$. Let us now take $s \in \phi^{-1}; f$ and let n be the depth of $s_{\restriction T^\omega}$. Such an n exists, because s is finite. We have $s \in \mathcal{T}^{n+1}(\phi^{-1}; f); \sigma^{(n)}$, but s has depth n so $s \in \sigma^{(n)}$, thus $s \in \sigma^{(\omega)}$. We have proved $\phi^{-1}; f \subseteq \sigma^{(\omega)}$, but since both strategies are total it suffices to prove an inclusion to get the equality [10], thus $f = \phi; \sigma^{(\omega)} = \sigma^\dagger$. □

Proposition 4.22. *Let $\mathcal{T} : \mathbf{Win} \to \mathbf{Win}$ be an open win-functor with a strict underlying open arena. Then $(\nu X.\mathcal{T}, \chi_{\mathcal{T}}^{-1})$ is a terminal coalgebra for $\mathcal{T}$.*

Proof. Let $(\mathcal{B}, \sigma)$ be another coalgebra for $\mathcal{T}$. We have to check that there is an unique total winning strategy $\sigma^\ddagger : \mathcal{B} \Rightarrow \nu X.\mathcal{T}$ such that the following diagram commutes.

$$\begin{array}{ccc} \mathcal{T}(\nu X.\mathcal{T}) & \xleftarrow{\mathcal{T}(\sigma^\ddagger)} & \mathcal{T}(\mathcal{B}) \\ {\scriptstyle \chi_{\mathcal{T}}}\uparrow & & \uparrow{\scriptstyle \sigma} \\ \nu X.\mathcal{T} & \xleftarrow[\sigma^\ddagger]{} & \mathcal{B} \end{array}$$

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

As for the previous case, we first give a direct definition of $\sigma^{(\omega)}$:

$$\begin{aligned}\sigma^{(1)} &= \sigma \\ \sigma^{(n+1)} &= \sigma^{(n)}; \mathcal{T}^n(\sigma) \\ \sigma^{(\omega)} &= \bigcup_{i \in \mathbb{N}} \sigma^{(n)}_{\restriction B \Rightarrow T^\omega}.\end{aligned}$$

Then, if we consider the obvious play isomorphism $\phi' : \mathcal{L}^\omega_{\rho X.T} \to \mathcal{L}^\omega_{T^\omega}$:

$$\sigma^\ddagger = \phi'^{-1}(\sigma^{(\omega)}).$$

For the same reasons as for initial algebras, we get that $\sigma^\ddagger$ is total and innocent. Let us detail the proof that it is also winning. Let t be a possibly infinite thread in $\sigma^\ddagger$, such that $t_{\restriction B}$ is winning on $\mathcal{B}$. Let us now consider $t' = t_{\restriction \nu X.\mathcal{T}}$. Either it has infinite depth and t is immediately winning, or it has finite depth n. Then we have $\phi(t) \in \sigma^{(n)}$ which is winning as a composition of winning strategies, so $\phi(t)$ is winning on $\mathcal{B} \Rightarrow \mathcal{T}^n(\mathbf{1})$, thus t is winning on $\mathcal{B} \Rightarrow \nu X.\mathcal{T}$ by definition of winning on $\nu X.\mathcal{T}$. The proof that $\sigma^\ddagger$ is the unique such total winning strategy is very similar to the case of initial algebras, and will not be repeated. □

As shown in Section 4.2.6, the loop construction $\rho X.T$ in **Inn** yields a minimal invariant to the corresponding open functor T; it is also both an initial algebra and a terminal coalgebra for T. The two different natural winning conditions on loops are able to separate these two roles: $\mu X.\mathcal{T}$ is the initial algebra, and $\nu X.\mathcal{T}$ is the terminal coalgebra.

At this point we almost have a μ-closed category, but we still have to prove that parametrised initial algebras and parametrised terminal coalgebras are open win-functors.

Proposition 4.23. *Let $\mathcal{T} : \mathbf{Win} \times \mathfrak{D} \to \mathbf{Win}$ be an open win-functor, where $\mathfrak{D}$ is a product of copies of* **Win** *and* $\mathbf{Win}^{op}$. *Then the parametrised initial algebra:*

$$\mathcal{T}^\mu : \mathfrak{D} \to \mathbf{Win}$$

is an open win-functor.

Proof. For simplicity, let us detail the case where $\mathfrak{D} = \mathbf{Win}$, so that $\mathcal{T}^\mu : \mathbf{Win} \to \mathbf{Win}$. The first step is to provide an open arena for the action of $\mathcal{T}^\mu$ on strategies, unsurprisingly this open arena will be $\rho X.T$, which still holes has labelled $\square_Y$. In the vocabulary we used for the proof of Proposition 4.21, we have (by definition of parametrised initial algebras) for all $\sigma : A \to B$:

$$\mathcal{T}^\mu(\sigma) = (\mathcal{T}(\mathcal{T}^\mu(B), \sigma); \chi_B)^\dagger.$$

We must prove that $\mathcal{T}^\mu(\sigma) = (\rho X.T)[\sigma/Y]$, thus let us take $s \in \mathcal{T}^\mu(\sigma)$ a P-view. Up to an isomorphism of plays ϕ, there is $n \in \mathbb{N}$ such that $s \in$

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

$(\mathcal{T}(\mathcal{T}^\mu(B),\sigma);\chi_B)^{(n)}$. By immediate induction on n, we prove that (1) s is functorial, (2) $s_{\restriction \rho X.T \Rightarrow \rho X.T} \in id_{\rho X.T}$, (3) $s_{\restriction A \Rightarrow B} \in \sigma$, hence we have an inclusion between two total strategies, which implies the equality.

It is obvious that $\mathcal{T}^\mu$ is continuous for $\trianglelefteq$. As needed, its action is given by an open arena but we still have to check that its strength $\theta^{\mu X.\mathcal{T}}_{\Gamma,\mathcal{A}} : \Gamma \times (\mu X.\mathcal{T})(\mathcal{A}) \to (\mu X.\mathcal{T})(\Gamma \times \mathcal{A})$ is winning. Let t be a possibly infinite thread of $\theta^{\mu X.\mathcal{T}}_{\Gamma,\mathcal{A}}$, and suppose that $t_{\restriction \Gamma \times (\mu X.\mathcal{T})(\mathcal{A})}$ is winning. In particular, $t_{\restriction (\mu X.\mathcal{T})(\mathcal{A})}$ is winning so t has finite depth n. But in this case we see immediately that (with an abuse of notation for the iteration of $\mathcal{T}$) $\phi(t) \in \theta^{\mathcal{T}^n(\mathbf{1},-)}_{\Gamma,\mathcal{A}}$ which is winning since $\mathcal{T}^n(\mathbf{1},-)$ is a composition of open win-functors. Hence t is winning. The cases of contravariant strengths and of functors $\mathcal{T}$ with higher arity are similar. □

Proposition 4.24. *Let $\mathcal{T} : \mathbf{Win} \times \mathfrak{D} \to \mathbf{Win}$ be an open win-functor, where $\mathfrak{D}$ is a product of copies of* **Win** *and* $\mathbf{Win}^{op}$. *Then the parametrised terminal coalgebra:*

$$\mathcal{T}^\nu : \mathfrak{D} \to \mathbf{Win}$$

is an open win-functor.

Proof. The proof that the action of $\mathcal{T}^\nu$ on strategies is that of $\rho X.T$ is identical to the case of a parametrised initial algebra. As above, we detail the case where $\mathfrak{D} = \mathbf{Win}$. Once again it is obvious that $\mathcal{T}^\nu$ is continuous for $\trianglelefteq$, thus it only remains to show that $\theta^{\nu X.\mathcal{T}}_{\Gamma,\mathcal{A}} : \Gamma \times (\nu X.\mathcal{T})(\mathcal{A}) \to (\nu X.\mathcal{T})(\Gamma \times \mathcal{A})$ is winning. Let t be a possibly infinite thread in $\theta^{\nu X.\mathcal{T}}_{\Gamma,\mathcal{A}}$. If $t_{\restriction (\nu X.\mathcal{T})(\Gamma \times \mathcal{A})}$ is winning, there is nothing to do. Otherwise it is a loss for Player, which implies in particular that t has finite depth n. As for the case of parametrised initial algebras, t is then a thread of the strength of an iteration of $\mathcal{T}$, which is winning, thus t is winning. □

Theorem 4.25. *The category* **Win** *is μ-closed (for pseudo-polynomial and strict formulas), hence is a sound model of (the pseudo-polynomial, strict fragment of) μLJ.*

An immediate corollary is that μLJ is consistent, since the pseudo-polynomial and strict formula $\mathbf{0}$ is not inhabited by any total strategy.

4.4. Completeness and faithfulness

Game semantics is known for its tight connection to syntax, hence for its tendency to provide fully complete models. In this section, we are concerned with this completeness problem: does our model characterise proofs in μLJ, or do we get additional objects? Also, does the equality in **Win** coincide with the equivalence relation on proofs generated by the reduction rules? To both questions the answer will be no, although we will try to say precisely where and why it fails.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

4.4.1. Completeness and normalisation

Let $\Gamma \vdash A$ be a sequent, then unfortunately there are in general far more total winning strategies in $[\![\Gamma]\!] \Rightarrow [\![A]\!]$ than there are μLJ proofs in $\Gamma \vdash A$. The reason for this is that total winning strategies are not required to be regular in any sense, nor even recursive. For an example, we do have a total winning strategy $\kappa : \mathtt{nat} \Rightarrow \mathtt{bool}$ which solves the halting problem for the n-th Turing machine. There can also be strategies which compute legal recursive functions, but which as a set of plays are not recursive. For an example, consider the function on $\mathtt{nat} \Rightarrow \mathtt{nat}$ which, given an input number n, calls its argument $BB(n)$ additional times (where BB is the busy beaver function) then returns the result of the final call. Extensionally it just computes the identity function, but as a set of plays it is not recursive.

However, nothing prevents us from applying the usual definability algorithm [30], although it will not always terminate. For each strategy σ, we will get a potentially infinite cut-free "proof" π such that $[\![\pi]\!] = \sigma$. Thus we will be able to eliminate cuts in μLJ, the price for this being the extension to infinite proofs.

Infinite proofs. Formulas of μLJ^ω are those of μLJ and "proofs" are potentially infinite trees made of rules from the identity group (without cut), the logical group, and rules μ_r and ν_l.

Proposition 4.26. *Let π be an infinite "proof" in μLJ^ω of a sequent $\Gamma \vdash A$, then the strategy $[\![\pi]\!]$ is innocent and total.*

Proof. All prefixes of π give rise to innocent partial strategies and $[\![\pi]\!]$ is obtained by taking the supremum. This strategy is still innocent, and it is also total, as there are no cuts in π. □

Remark. Of course, this construction does not necessarily produce winning strategies. For an example, nothing prevents us from forming the following "proof".

$$\pi \quad = \quad \dfrac{\dfrac{\dfrac{\pi}{\vdash \mathtt{nat}}}{\vdash 1 + \mathtt{nat}} \overrightarrow{+_r}}{\vdash \mathtt{nat}}\, \mu_r$$

This is the infinite "integer", which is always the successor of itself. Its interpretation is an innocent total strategy, however it clearly cannot be winning.

Definability. Let now be $\Gamma \vdash A$ a sequent, and $\sigma : [\![\Gamma]\!] \to [\![A]\!]$ a total winning strategy. For each finite[7] subset σ' of σ, we get by definability a partial proof $(\sigma')^\bullet$. If we have $\sigma' \subset \sigma''$ two finite subsets of σ, then $(\sigma')^\bullet$ is clearly a prefix of $(\sigma'')^\bullet$. Hence we can once again take the supremum, which yields a "proof" $\sigma^\bullet$ in μLJ^ω which such that $[\![\sigma^\bullet]\!] = \sigma$ by construction.

[7] Here, finite means that it has a finite number of P-views, not a finite number of plays.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

Consequences. Let us state an immediate corollary of this construction: μLJ does enjoy cut elimination modulo the extension to the infinite syntax. Let us note that this cut elimination process will not be up to reduction in the syntax, but with respect to the equivalence relation generated by the model:

$$\pi \simeq_{\mathbf{Win}} \pi' \Leftrightarrow [\![\pi]\!] = [\![\pi']\!].$$

Corollary 4.27. *Let π be a proof in μLJ of a sequent $\Gamma \vdash A$. Then, there is a possibly infinite cut-free "proof" π' of $\Gamma \vdash A$, such that $\pi \simeq_{\mathbf{Win}} \pi'$.*

Of course, if neither Γ nor A contains fixpoints, the definability operation on $[\![\pi]\!]$ yields a winning strategy on a finite arena, and these (see details in [10]) can be proved to be in fact finite. Hence in this case, this cut elimination procedure yields a finite μLJ proof. Moreover this cut elimination result is effective, since the interpretation of π by an innocent strategy amounts to the computation of a normal form of π by a notion of linear head reduction [15].

4.4.2. Faithfulness

The model is not faithful, as will be shown by the following counter-example. As the syntax of μLJ is not so intuitive, let us present it in a ML-like syntax. First consider the two terms below:

```
let rec iter f n b =        let rec iter' f n b =
   if n = 0 then b             if n = 0 then b
   else iter f (n-1) (f b)     else iter' f (n-1) (not (f (not b)))
```

It is immediate to check (by induction on n) that on every input `f`, `n` and `b`, `not (iter f n (not b))` and `iter' f n b` return the same result. In fact, these two functions (regarded as μLJ proof terms) have the same infinite unfolding in μLJ^{ω}: the unfolding of `iter'` reveals sequences of the form `not (not (-))` which immediately simplify, `not` being involutive. Thus these two terms have the same interpretation in **Win**, whereas they cannot be convertible since a `let rec` subterm would correspond to a μ_l rule, and subproofs beginning by μ_l are preserved by reduction.

Note that if the intended model of induction and coinduction is initial algebras and terminal coalgebras then the reduction rules of μLJ are incomplete since they only correspond to *weak* initial algebras and terminal coalgebras: the uniqueness condition of the universal property does not hold (however, any "complete" set of rules in this sense would give rise to an undecidable equivalence relation [33]). On the other hand, our games model should exactly implement the equivalence relation generated by the interpretation of proofs with initial algebras and terminal coalgebras: we believe (but leave it open for the time being) that for any two proofs π_1 and π_2, we have $\pi_1 \simeq_{\Gamma} \pi_2$ if and only if they have the same interpretation in any μ-closed category.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

5. Conclusion

The main contributions of this paper are a notion of μ-closed categories as categorical models of total programming languages with fixpoints, and the construction of a μ-closed (for the restricted fragment of strict and pseudo-polynomial types) category of games and total winning strategies, yielding a consistency result for an intuitionistic sequent calculus μLJ with inductive and coinductive datatypes, and a normalisation proof (for its strict and pseudo-polynomial fragment).

Overall, many technical difficulties follow from the choice of representing fixpoints with loops: one is brought to consider open functors, which unfortunately (at least for now) do not cover all types. An alternative approach would be to rely on the same category of games, but build strong functors directly by induction on types and build initial algebras and terminal coalgebras as least and greatest fixpoints for the cpo $\trianglelefteq$ on win-games. This would yield the same interpretation of proofs as strategies but would lift the pseudo-polynomial restriction and simplify parts of the presentation. However, the current approach also has its merits: making explicit the proximity to parity games might be a starting point for future work. Moreover, we think the construction of open functors is interesting in itself: they could be useful for other purposes, such as modelling polymorphism.

Let us mention two research perspectives opened by this model. First, one could study the equivalence relation on proofs generated by the model, comparing it with the equality generated by the interpretation of proofs in arbitrary μ-closed categories. One could then try to isolate subsystems of μLJ for which this equivalence relation is decidable, for instance by applying automata-theoretic tools. Another avenue of investigation is the decidability of our winning conditions: it is in general undecidable to know if a term given by general recursion (say, a term of FPC) gives rise to a winning strategy. However, isolating subsystems where this is decidable may prove worthwhile to guide the construction of termination checkers.

Acknowledgements. We would like to thank Russ Harmer and Luigi Santocanale for enlightening discussions. This work also greatly benefited from Thierry Coquand and Martin Hyland's proofreading of the author's PhD thesis, as well as from Guy McCusker's enlightening advice while preparing the final version. Finally, we would like to thank the anonymous referees for useful suggestions and comments.

References

[1] A. Abel and T. Altenkirch, A predicative strong normalisation proof for a lambda-calculus with interleaving inductive types, in *Selected Papers from Int. Workshop on Types for Proofs and Programs, TYPES '99 (Lökeberg, June 1999),* edited by T. Coquand, P. Dybjer, B. Nordström and J.M. Smith, Springer. *Lect. Notes Comput. Sci.* **1956** (2000) 21–40.

[2] S. Abramsky, Semantics of interaction: an introduction to game semantics, in *Semantics and Logics of Computation,* edited by A. Pitts and P. Dybjer. Publications of the Newton Institute, Cambridge University Press **14** (1996) 1–31.

[3] A. Arnold and D. Niwiński, *Rudiments of μ-Calculus, Studies in Logic and the Foundations of Mathematics.* North-Holland **146** (2001).

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

[4] D. Baelde and D. Miller, Least and greatest fixed points in linear logic, in *Proc. of 14th Int. Conf. on Logic for Programming, Artificial Intelligence and Reasoning, LPAR 2007 (Yerevan, Oct. 2007),* edited by N. Dershowitz and A. Voronkov, Springer. *Lect. Notes Artif. Intell.* **4790** (2007) 92–106.

[5] E.S. Bainbridge, P.J. Freyd, A. Scedrov and P.J. Scott, Functorial polymorphism. *Theoret. Comput. Sci.* **70** (1990) 35–64.

[6] W. Belkhir and L. Santocanale, The variable hierarchy for the lattice μ-calculus, in *Proc. of 15th Int. Conf. on Logic for Programming, Artificial Intelligence and Reasoning, LPAR 2008 (Doha, Nov. 2008),* edited by I. Cervesato, H. Veith and A. Voronkov, Springer. *Lect. Notes Artif. Intell.* **5330** (2008) 605–620.

[7] A. Blass, A game semantics for linear logic. *Ann. Pure Appl. Log.* **56** (1992) 183–220.

[8] P. Clairambault, Least and greatest fixpoints in game semantics, in *Proc. of 12th Int. Conf. on Foundations of Software Science and Computation Structures, FoSSaCS 2009 (York, March 2009),* edited by L. de Alfaro Springer. *Lect. Notes in Comput. Sci.* **5504** (2009) 16–31.

[9] P. Clairambault. Least and greatest fixpoints in game semantics 2: strong functors and interleaving types. Informal proceedings of the workshop on Fixed Points in Computer Science, Coimbra, Portugal, September 2009.

[10] P. Clairambault, Logique et interaction: une étude sémantique de la totalité. Ph.D. thesis, Université Paris Diderot, Paris 7 (2010).

[11] P. Clairambault and R. Harmer, Totality in arena games. *Ann. Pure Appl. Log.* **161** (2010) 673–689.

[12] J.R.B. Cockett and T. Fukushima, *About Charity*, Technical report. University of Calgary (1992).

[13] J.R.B. Cockett and D. Spencer, Strong categorical datatypes I, in *Proc. of Int. Summer Category Theory Meeting (Montréal, June 1991)*, edited by R.A.G. Seely. Canadian Math. Soc. Conference Proceedings. *Amer. Math. Soc.* **13** (1992) 141–169.

[14] J.R.B. Cockett and D. Spencer, Strong categorical datatypes II: A term logic for categorical programming. *Theoret. Comput. Sci.* **139** (1995) 69–113.

[15] V. Danos, H. Herbelin and L. Regnier, Game semantics & abstract machines, in *Proc. of 11th Ann. IEEE Symp. on Logic in Computer Science, LICS '96 (New Brunswick, NJ, July 1996).* IEEE CS Press (1996) 394–405.

[16] P. Dybjer, Inductive sets and families in Martin-Löf's type theory and their set-theoretic semantics, in Logical Frameworks, edited by G. Huet and G. Plotkin. Cambridge University Press (1991) 280–306.

[17] P.J. Freyd, Algebraically complete categories, in *Proc. of Int. Category Theory Conf., CT '90 (Como, July 1990),* edited by A. Carboni and M.C. Pedicchio and G. Rosolini, Springer. *Lect. Notes Math.* **1488** (1990) 95–104.

[18] P.J. Freyd, Recursive types reduced to inductive types, in *Proc. of 5th IEEE Ann. Symp. on Logic in Computer Science, LICS '90 (Philadelphia, PA, June 1990).* IEEE CS Press (1990) 498–507.

[19] P.J. Freyd, Remarks on algebraically compact categories, in *Proc. of LMS Symp. on Applications of Categories in Computer Science (Durham, July 1991),* edited by M.P. Fourman, P.T. Johnstone and A.M. Pitts. Cambridge University Press. *London Math. Soc. Lect. Notes Ser.* **177** (1992) 95–106.

[20] J.-Y. Girard, Y. Lafont and P. Taylor, Proofs and Types, Cambridge University Press. Cambridge Tracts in *Theoret. Comput. Sci.* **7** (1989).

[21] R. Harmer, J.M.E. Hyland and P.-A. Melliès, Categorical combinatorics for innocent strategies, in *Proc. of 22nd Ann. IEEE Symp. on Logic in Computer Science, LICS '07 (Wrocław, July 2007).* IEEE CS Press (2007) 379–388.

[22] J.M.E. Hyland, Game semantics, in *Semantics and Logics of Computation,* edited by A. Pitts and P. Dybjer. Publications of the Newton Institute, Cambridge University Press **14** (1996) 131–184.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press

[23] J.M.E. Hyland and C.H.L. Ong, On full abstraction for PCF: I, II, and III. *Inf. Comput.* **163** (2000) 285–408.

[24] A. Joyal and R. Street, Braided monoidal categories, Math. Report 860081. Macquarie University (1986).

[25] A. Kock, Monads on symmetric monoidal closed categories. *Arch. Math.* **21** (1970) 1–10.

[26] J. Lambek and P.J. Scott, Introduction to Higher Order Categorical Logic. Cambridge University Press. Cambridge Studies in *Adv. Math.* **7** (1988).

[27] O. Laurent, Classical isomorphisms of types. *Math. Struct. Comput. Sci.* **15** (2005) 969–1004.

[28] T. Leinster, Higher Operads, Higher Categories, London Math. Soc. Cambridge University Press. *Lect. Notes Ser.* **298** (2004).

[29] P. Martin-Löf, Hauptsatz for the intuitionistic theory of iterated inductive definitions, in *Proc. of 2nd Scandinavian Logic Symp. (Oslo, June 1970),* edited by J.E. Fenstad, North-Holland. *Stud. Logic Found. Math.* **63** (1971) 179–216.

[30] G. McCusker, Games and full abstraction for a functional metalanguage with recursive types, Ph.D. thesis, Imperial College (1996). Also published in Springer's Distinguished Dissertations in *Comput. Sci. ser.* (1998).

[31] G. McCusker, Games and full abstraction for FPC. *Inf. Comput.* **160** (2000) 1–61.

[32] P.-A. Melliés, Typed lambda-calculi with explicit substitutions may not terminate, in *Proc. of 2nd Int. Conf. on Typed Lambda Calculi and Applications, TLCA '95 (Edinburgh, Apr. 1995),* edited by M. Dezani-Ciancaglini and G. Plotkin, Springer. *Lect. Notes Comput. Sci.* **902** (1995) 328–334.

[33] M. Okada and P.J. Scott, A note on rewriting theory for uniqueness of iteration. *Theory. Appl. Categ.* **6** (1999) 47–64.

[34] J. Power and G. Rosolini, Fixpoint operators for domain equations. *Theoret. Comput. Sci.* **278** (2002) 323–333.

[35] L. Santocanale, Free μ-lattices. *J. Pure Appl. Algebra* **168** (2002) 227–264.

[36] L. Santocanale, μ-bicomplete categories and parity games. *Theor. Inform. Appl.* **36** (2002) 195–227.

[37] W. Thomas, Languages, Automata, and Logic, in *Handbook of Formal Languages, Beyond Words,* edited by G. Rozenberg and A. Salomaa. Springer **3** (1997) 389–455.

Communicated by R. Matthes and T. Uustalu.
Received September 28, 2012. Accepted September 28, 2012.

https://doi.org/10.1051/ita/2012028 Published online by Cambridge University Press