

*Under consideration for publication in Math. Struct. in Comp. Science*

# Denotational semantics for guarded dependent type theory

ALEŠ BIZJAK<sup>1</sup> and RASMUS EJLERS MØGELBERG<sup>2†</sup>

<sup>1</sup> Aarhus University

<sup>2</sup> IT University of Copenhagen

Received April 14, 2020

We present a new model of Guarded Dependent Type Theory (GDTT), a type theory with guarded recursion and multiple clocks in which one can program with, and reason about coinductive types. Productivity of recursively defined coinductive programs and proofs is encoded in types using guarded recursion, and can therefore be checked modularly, unlike the syntactic checks implemented in modern proof assistants.

The model is based on a category of covariant presheaves over a category of time objects, and quantification over clocks is modelled using a presheaf of clocks. To model the *clock irrelevance axiom*, crucial for programming with coinductive types, types must be interpreted as presheaves internally right orthogonal to the object of clocks. In the case of dependent types, this translates to a lifting condition similar to the one found in homotopy theoretic models of type theory, but here with an additional requirement of uniqueness of lifts. Since the universes defined by the standard Hofmann-Streicher construction in this model do not satisfy this property, the universes in GDTT must be indexed by contexts of clock variables. We show how to model these universes in such a way that inclusions of clock contexts give rise to inclusions of universes commuting with type operations on the nose.

## 1. Introduction

Type theories with dependent types such as Martin-Löf's (1973) Type Theory or the Extended Calculus of Constructions (Luo 1994) are systems that can be simultaneously thought of as programming languages and logical systems. One reason why this is useful is that programs, their specification and the proof that a program satisfies this specification, can be expressed in the same language. In these systems, the logical interpretation of terms forces a totality requirement on the programming language, i.e., rules out general recursion, since nonterminating programs can inhabit any type, and thus be interpreted as proofs of false statements.

The lack of general recursion is a limitation both from a programming and a logical perspective. For example, when programming with coinductive types, the natural way to program and reason about these is by recursion. For example, the constant stream of zeros can be naturally described as the solution to the equation  $\mathbf{zeros} = 0 :: \mathbf{zeros}$ . To ensure logical consistency, such recursive definitions must be productive, in the sense that any finite segment of the stream can be computed

<sup>†</sup> Corresponding author. Full address: IT University of Copenhagen, Department of Computer Science, Rued Langgaards Vej 7, 2300 Copenhagen, Denmark. Email: [mogel@itu.dk](mailto:mogel@itu.dk)

in finite time. Modern proof assistants such as Coq (2004) and Agda (Norell 2007) do support coinductive types and recursive definitions such as the above but the productivity checks are based on a syntactical analysis of terms, and are not modular. This means that using these in larger applications requires sophisticated tricks (Danielsson 2010). This paper is concerned with a new technique using guarded recursion to express productivity in types.

Guarded recursion in the sense of Nakano (2000) is a safe way of adding recursion to type theory without breaking logical consistency. The idea is to guard all unfoldings of recursive equations by time steps in the form of a modal type constructor  $\blacktriangleright$ . The type  $\blacktriangleright A$  should be thought of as a type of elements of  $A$  available one time step from now. Values can be preserved by time steps using an operator `next` satisfying  $\text{next } t : \blacktriangleright A$  whenever  $t : A$ . The fixed point operator has type  $\text{fix} : (\blacktriangleright A \rightarrow A) \rightarrow A$  and computes, for any  $f$ , a fixed point for  $f \circ \text{next}$ . This is particularly useful when programming with *guarded recursive types*, i.e., recursive types where all occurrences of the type parameter appears guarded by a  $\blacktriangleright$ . For example, a guarded recursive type of streams would satisfy  $\text{Str} = \text{Nat} \times \blacktriangleright \text{Str}$  and the stream of zeros can be defined as  $\text{fix}(\lambda xs. (0, xs))$ . The type  $\blacktriangleright \text{Str} \rightarrow \text{Str}$  in fact exactly captures productive recursive stream definitions. Using universes, the type  $\text{Str}$  can itself be computed as a guarded recursive fixed point. In this paper we use universes à la Tarski, i.e., for any term  $A : \mathcal{U}$  there is a type  $\text{El}(A)$ . If we assume an operation  $\overline{\blacktriangleright} : \blacktriangleright U \rightarrow U$  satisfying  $\text{El}(\overline{\blacktriangleright}(\text{next}(A))) = \blacktriangleright \text{El}(A)$ , then the type of guarded streams can be encoded as  $\text{Str} \stackrel{\text{def}}{=} \text{El}(\text{fix}(\lambda X. \text{Nat} \times \overline{\blacktriangleright}(X)))$ .

The guarded recursive type of streams above is not the usual type of streams. In particular, a term of type  $\text{Str} \rightarrow \text{Str}$  must always be *causal* in the sense that the  $n$  first element of output only depend on the  $n$  first elements of input. Indeed, causality of maps is crucial for the encoding of productivity in types. On the other hand, a *closed* term of type  $\text{Str}$  does denote a full stream of numbers, and likewise a term of type  $\text{Str}$  in a context consisting solely of a variable  $x : \text{Nat}$  gives rise to an assignment of numbers to full streams of numbers. In general, this holds if the context is stable, i.e., consists entirely of time-independent types.

### 1.1. Guarded recursion with multiple clocks

Atkey & McBride (2013) proposed a way to program with coinductive types using this idea, expressing time-independence by indexing all  $\blacktriangleright$  operators, `next` and `fix` by clocks. For example, if  $t : A$  and  $\kappa$  is a clock,  $\text{next}^\kappa t : \blacktriangleright^\kappa A$  and the type  $\blacktriangleright^\kappa A$  is to be thought of as elements of type  $A$  available one  $\kappa$ -time step from now. Likewise the guarded recursive type of streams must be indexed with a clock and assumed to satisfy  $\text{Str}^\kappa = \text{Nat} \times \blacktriangleright^\kappa \text{Str}^\kappa$ . There are no operations on clocks, only clock variables, although we will see that a single clock constant can be useful. We refer to this as *guarded recursion with multiple clocks*, and the case of a single operator  $\blacktriangleright$  as *guarded recursion with a single clock* or sometimes simply *the single clock case*.

In turn, *clock quantification* of guarded dependent type theory allows us to define the coinductive type of streams from the guarded recursive type of streams as  $\forall \kappa. \text{Str}^\kappa$ . Clock quantification behaves similarly to the dependent product type in the sense it has analogous introduction and elimination rules; terms of this type are introduced by clock abstraction  $\Lambda \kappa. t$ , and eliminated using clock application  $t[\kappa']$ , provided  $\kappa'$  is a valid clock. However, clock quantification additionally satisfies the *clock irrelevance* property, which is crucial for showing that types such as  $\forall \kappa. \text{Str}^\kappa$  satisfy the properties expected of coinductive types, i.e., that they are final coalgebras. Using

these constructs and properties we can program with streams using guarded recursion, ensuring productivity of definitions using types.

This paper presents a model of GDTT (Bizjak et al. 2016), an extensional type theory with guarded recursion and clocks, in which one can program with, and reason about guarded recursive and coinductive types. To motivate some of the constructions of GDTT, we now take a closer look at the encoding of coinductive streams as  $\forall\kappa. \mathbf{Str}^\kappa$ . As a minimal requirement for this to work, we need an isomorphism of types  $\forall\kappa. \mathbf{Str}^\kappa \cong \mathbf{Nat} \times \forall\kappa. \mathbf{Str}^\kappa$ . This isomorphism is a composition of three isomorphisms

$$\begin{aligned} \forall\kappa. \mathbf{Str}^\kappa &= \forall\kappa. \mathbf{Nat} \times \blacktriangleright^\kappa \mathbf{Str}^\kappa \\ &\cong (\forall\kappa. \mathbf{Nat}) \times \forall\kappa. \blacktriangleright^\kappa \mathbf{Str}^\kappa \\ &\cong \mathbf{Nat} \times \forall\kappa. \blacktriangleright^\kappa \mathbf{Str}^\kappa \\ &\cong \mathbf{Nat} \times \forall\kappa. \mathbf{Str}^\kappa \end{aligned}$$

The first isomorphism follows from the fact that  $\forall\kappa. A$  behaves essentially as the dependent product type  $\prod (\kappa : \mathbf{clock}) . A$ , and thus distributes over binary products. For the second isomorphism, we need  $\mathbf{Nat} \cong \forall\kappa. \mathbf{Nat}$ . One direction of this isomorphism maps  $x : \mathbf{Nat}$  to  $\Lambda\kappa. x$ , and the opposite way evaluates an element in  $\prod (\kappa : \mathbf{clock}) . \mathbf{Nat}$  at a *clock constant*  $\kappa_0$ . The composition on  $\mathbf{Nat}$  is obviously the identity, but for the other composition to be the identity, we need to assume the  $\eta$ -axiom for  $\forall\kappa. A$ , and the *clock irrelevance axiom*, which states that whenever  $t : \forall\kappa. A$  and  $\kappa$  is not in  $A$ , then evaluating  $t$  at different clocks give the same result. One of the main contributions of this paper is that this axiom can be modelled using a notion of orthogonality. The last isomorphism requires an inverse force  $\forall\kappa. \blacktriangleright^\kappa A \rightarrow \forall\kappa. A$  to the map induced by  $\mathbf{next}^\kappa$ .

In this paper we focus on modelling GDTT, and refer the reader to (Møgelberg 2014) for a proof of correctness of the coinductive type encodings.

## 1.2. A model of guarded recursion with multiple clocks

In the single clock case guarded recursion can be modelled in the topos of trees, i.e., the category  $\mathbf{Set}^{\omega^{\text{op}}}$  of presheaves over the ordered natural numbers  $\omega$ . In this model, a closed type is modelled as a sequence of sets  $(X_n)_{n \in \mathbb{N}}$  together with restriction maps  $X_{n+1} \rightarrow X_n$ . We think of  $X_n$  as the type as it looks if we have  $n$  steps to reason about it. For example, in the guarded recursive type of streams, since the tail takes one computation step to compute, one can compute the  $n+1$  first elements of the stream in  $n$  steps. We can represent this by the object defined as  $\mathbf{Str}_n = \mathbb{N}^{n+1}$  with restriction maps as projections.

In this model  $\blacktriangleright X$  is the object given by  $(\blacktriangleright X)_0 = 1$  and  $(\blacktriangleright X)_{n+1} = X_n$ . Redefining  $\mathbf{Str}_n$  to be  $\mathbb{N}^{n+1} \times 1$  (and associating products to the right) one gets  $\mathbf{Str} = \mathbb{N} \times \blacktriangleright \mathbf{Str}$ . In the empty context a term  $t : \blacktriangleright^\kappa A \rightarrow A$  is modelled as a family of maps  $t_{n+1} : A_n \rightarrow A_{n+1}$  and  $t_0 : 1 \rightarrow A_0$ . The fixed point operator maps such a family to the global element  $\mathbf{fix}(t) : 1 \rightarrow A$  defined as  $\mathbf{fix}(t)_n = t_n \circ \dots \circ t_0$ . We refer to (Birkedal et al. 2012) for further details.

In this paper we extend this to a model of guarded recursion with multiple clocks. The model is a presheaf category over a category  $\mathbb{T}$  of time objects. In the single clock case, a time object was simply a number indicating the number of ticks left on the unique clock. In the case of multiple clocks, a time object consists of a finite set of clocks  $\mathcal{E}$ , together with a map  $\delta : \mathcal{E} \rightarrow \mathbb{N}$  indicating the number of ticks left on each clock. A morphism of time objects  $\sigma : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')$  is a

map  $\sigma : \mathcal{E} \rightarrow \mathcal{E}'$  such that  $\delta'(\sigma(\lambda)) \leq \delta(\lambda)$  for each  $\lambda \in \mathcal{E}$ . Such a morphism can rename clocks, introduce new clocks (elements of  $\mathcal{E}'$  outside the image of  $\sigma$ ) and even synchronise clocks (by mapping them to the same clock). The inequality requirement corresponds to the inequalities between numbers in the topos of trees.

We consider *covariant* presheaves on  $\mathbb{T}$ , i.e., the category of functors  $\mathbb{T} \rightarrow \text{Set}$ . In this category there is an object of clocks given by  $\mathcal{C}(\mathcal{E}, \delta) = \mathcal{E}$ , which we use to model clock variables. Clock quantification is modelled as a dependent product over  $\mathcal{C}$ . With this interpretation, for a type  $A$  in which  $\kappa$  does not appear free, the type  $\forall \kappa. A$  is modelled as a simple function type  $\mathcal{C} \rightarrow A$ . The clock irrelevance axiom mentioned above then states that the map  $A \rightarrow (\mathcal{C} \rightarrow A)$  mapping an element  $x$  in  $A$  to the constant map to  $x$  is an isomorphism. Of course, this does not hold for all presheaves  $A$ , and so we must show that this holds for the interpretation of any type. Note that it does not hold for  $A = \mathcal{C}$ , and so, although  $\forall \kappa. A$  is modelled as a dependent product over the presheaf of clocks, there is no type of clocks in the type theory. This is similar to the status of the interval in cubical type theory (Cohen et al. 2016), which is not itself a type, but still the set of types is closed under dependent products over the interval (these are path types).

For dependent types the condition becomes a unique lifting property. In a presheaf model of type theory a type depending on a context is modelled as a family  $A$  over a presheaf  $\Gamma$ . To this can be associated a projection  $p : \Gamma.A \rightarrow \Gamma$  corresponding to syntactic projection between contexts. This must satisfy the condition that for all  $Y$ , and for all commutative squares as in the outer square below (where  $\pi_Y$  is the projection), there exists a unique  $h$  such that the two triangles commute.

$$\begin{array}{ccc} Y \times \mathcal{C} & \xrightarrow{f} & \Gamma.A \\ \downarrow \pi_Y & \nearrow h & \downarrow p \\ Y & \xrightarrow{g} & \Gamma \end{array}$$

We say that such a map  $p$  is *internally right orthogonal* to  $\mathcal{C}$ . This condition is similar to the notion of fibration used in models of homotopy theoretic models of type theory (Awodey & Warren 2009, Kapulkin & Lumsdaine 2012) and cubical type theory (Bezem et al. 2013), except that here the liftings are unique. This means that it can be considered a property that must be proved for each type, rather than structure that is part of the interpretation of a type.

### 1.3. Universes

Since our model is a presheaf category, one would hope that modelling universes would follow the standard Hofmann-Streicher construction (Hofmann & Streicher 1999), restricting to the elements internally right orthogonal to  $\mathcal{C}$ . Unfortunately, this universe  $\mathcal{U}$  is not itself internally right orthogonal to  $\mathcal{C}$ . The reason is that there is a map  $\mathcal{U} \times \mathcal{C} \rightarrow \mathcal{U}$  mapping a type  $A$  and a clock  $\kappa$  to  $\blacktriangleright^{\kappa} A$ , and this map is not constant in the  $\mathcal{C}$  component. This is a new semantic manifestation of a known problem, and we follow the solution used in GDTT, which is to have a family of universes  $(\mathcal{U}_{\Delta})_{\Delta}$  in the syntax, indexed by finite sets of clock variables. Each universe  $\mathcal{U}_{\Delta}$  is to be thought of as the universe of types independent of the clocks outside of  $\Delta$ , and the type operation  $\blacktriangleright^{\kappa}$  is restricted on the universe  $\mathcal{U}_{\Delta}$  to the  $\kappa$  in  $\Delta$ .

This means that universes are indexed by a new dimension, similar to the indexing of universes by natural numbers used to avoid Russell's paradox (Martin-Löf 1973). Fortunately, there are

inclusions  $U_{\Delta} \rightarrow U_{\Delta'}$  for  $\Delta \subseteq \Delta'$ , and we prove universe polymorphism in this dimension. This means that operations on types such as dependent product can be defined on the universes in such a way that they commute with the inclusions mentioned above, not just up to isomorphism, but indeed up to identity. We hope that, as a consequence of this result, the indexing of universes by clock contexts can be suppressed in practical applications, just like the indexing by natural numbers is often suppressed.

#### 1.4. Related work

The notion of guarded recursion studied in this paper originates with Nakano (2000). Much of the recent interest in guarded recursion is due to the guarded recursive types, which can even have negative occurrences and thus, by adding  $\blacktriangleright$  operators in appropriate places, provide approximations to solutions to equations that can not be solved in set theory. These have been used to construct syntactic models and operational reasoning principles for (also combinations of) advanced programming language features including general references, recursive types, countable non-determinism and concurrency (Birkedal et al. 2012, Bizjak et al. 2014, Svendsen & Birkedal 2014). This technique can be understood as an abstract form of step-indexing (Appel & McAllester 2001), the connection to which was first discovered by Appel et al. (2007). Most of these applications have been constructed using logics with guarded recursion, such as the internal language of the topos of trees (Birkedal et al. 2012), but recently GDTT has been used to construct denotational models of programming languages like FPC (Møgelberg & Paviotti 2016), modelling the recursive types of these as guarded recursive types.

Most type theories with guarded recursion considered until now have been extensional, with the exception of *guarded cubical type theory* (Birkedal et al. 2016). This has, however, only been developed in the single clock case, although there exists an experimental version with multiple clocks.

Guarded recursion with multiple clocks was first developed in the simply typed setting by Atkey & McBride (2013). The second named author (Møgelberg 2014) extended these results to a model of dependent type theory and proved correctness of the coinductive type encodings inside a type theory with guarded recursion. These two early works used a restricted version of clock application, allowing  $t[\kappa'] : A[\kappa'/\kappa]$  for  $t : \forall\kappa.A$  only if  $\kappa'$  does not appear free in  $\forall\kappa.A$ . This condition can be thought of as disallowing the clocks  $\kappa$  and  $\kappa'$  to be synchronised in  $A$ , and was motivated by the models considered at the time. This restriction has unfortunate consequences for the syntactic metatheory. In particular, the present authors do not know how to prove type preservation for clock  $\beta$ -reductions in these systems.

This led us to suggest a different model (Bizjak & Møgelberg 2015) given by a family of presheaf categories  $\mathfrak{GR}(\Delta)$  indexed by clock contexts (finite sets of clock variables)  $\Delta$ . This model should in principle lead to a model of GDTT, but this was never done in detail, due to a problem with modelling substitution of clock variables. Such substitutions are given by maps  $\sigma : \Delta \rightarrow \Delta'$  and must correspond semantically to functors  $\mathfrak{GR}(\Delta) \rightarrow \mathfrak{GR}(\Delta')$ . While these functors can be defined in a natural way, they do not commute with dependent function types up to identity, only up to isomorphism. This problem can be thought of as a coherence problem, similar to the one arising when modelling type theory in locally cartesian closed categories (Hofmann 1994). It is very likely that Hofmann's (1994) solution to the latter problem can be adapted to construct an equivalent family of categories for which the functors preserve construction on the nose, but

we prefer the solution presented here, which organises all these categories inside one big presheaf category, thereby reducing the model construction to the known construction of modelling type theory in a presheaf category. The precise relation to the categories  $\mathfrak{GR}(\Delta)$  is discussed in Section 9.

Recently, GDTT has been refined to *clocked type theory* (CloTT) (Bahr et al. 2017), which has better operational properties, and indeed strong normalisation has been proved for clocked type theory in the setting without identity types. The principal novel feature of CloTT is the notion of ticks on a clock introduced in contexts as assumptions of the form  $\alpha : \kappa$ , for  $\kappa$  a clock. Ticks can be used to encode the delayed substitutions (see Section 6) of GDTT, and reduce most of the equalities between these to  $\beta$  and  $\eta$  equalities. Since the initial development of the research reported here, Mannaa & Møgelberg (2018) have developed a model CloTT based on the model presented here. Their paper however, does not describe how to model the clock irrelevance axiom, nor universes as presented here. Also, the presence of ticks makes the model construction for CloTT rather complicated and so we have chosen to present the model in the simpler setting of GDTT first.

In recent work on guarded computational type theory, Sterling & Harper (2018) propose a clock intersection connective to be used as a special ‘irrelevant’ quantification over clocks. Using this they encode coinductive types, while avoiding the indexing of universes by clock contexts as done here. Irrelevant clock quantification is interpreted using intersection of sets in a syntactic model, in which types are essentially indexed sets of values. This is similar to the original interpretation of clock quantification in the work of Atkey & McBride (2013). A related irrelevant quantification over sizes appears in the work of Abel et al. (2017). However, it is unclear how to give *denotational* semantics of such a constructor. We remark that the model used by Sterling & Harper (2018) is based on a category very similar to the presheaf category used in this paper and that these models were discovered independently.

One way of understanding the need for multiple clocks for encoding coinductive types is that they provide a controlled way of eliminating the  $\blacktriangleright$  modality as in the term  $\text{force} : \forall \kappa. \blacktriangleright^{\kappa} A \rightarrow \forall \kappa. A$  mentioned above. As an alternative solution to this problem, ? have suggested to use an *always* modality  $\blacksquare$  satisfying  $\blacksquare \blacktriangleright A \cong \blacksquare A$ . It is yet unclear how far this idea can be extended, in particular if it can be used for encoding nested inductive and coinductive types.

Sized types (Hughes et al. 1996) offer a different approach to the problem of encoding productivity in types. The idea is to annotate approximations of a coinductive type with the number of unfoldings that can be applied to it. The real coinductive type is then the approximation associated with an infinite ordinal. When programming with sized types, the sizes sometimes get in the way, motivating the concept of irrelevant quantification over sizes mentioned above. The syntactic theory of sized types is further developed than that of guarded recursion (Abel & Pientka 2013, Abel et al. 2017, Sacchini 2013), and sized types are also available in an experimental extension of Agda. Sized types have not been used as abstract step-indexing in the sense described above for guarded recursion, and the authors are not aware of any work on denotational semantics for sized types.

Our view is that guarded recursion should be thought of as an abstraction of sized types, providing similar benefits as the abstraction of step-indexing, in particular by hiding Kripke structure present in the model. This view is supported by work by Veltri & van der Weide (2019) in which a model of guarded recursion is constructed in Agda using sized types to model recursion. In this work, the model is restricted to a simply typed language specialised

to the case of just 0 or 1 clocks, thus avoiding the issue of clock synchronisation treated in this paper.

### 1.5. Overview

Section 2 presents a basic type theory **Core-GDTT** for guarded recursion with multiple clocks. This can be thought of as the core of **GDTT** (Bizjak et al. 2016) although we use a slightly different presentation. Section 3 then presents a basic model of **Core-GDTT** in the presheaf category  $\text{Set}^{\mathbb{T}}$ , and Section 4 shows how to model the clock irrelevance axiom. The following sections 5 and 6 then extend **Core-GDTT** with extensional identity types and delayed substitutions, a construction needed for reasoning about guarded recursive and coinductive types. Section 7 is devoted to universes and modelling universe polymorphism in the clock context dimension and Section 8 sketches how to extend Hofmann’s interpretation of dependent type theory syntax (Hofmann 1997) to interpreting **GDTT** into the model presented in this paper. Finally the relations to the categories  $\mathfrak{GR}(\Delta)$  constructed in previous work (Bizjak & Møgelberg 2015) by the authors is discussed in Section 9.

## 2. A basic type theory for guarded recursion

This section introduces **Core-GDTT** a presentational variant of a fragment of the type theory **GDTT** (Bizjak et al. 2016). The fragment is the one not mentioning universes, delayed substitutions and identity types. All these will be treated in Sections 5–7. The variation referred to above is in the treatment of clocks, which in previous work (Bizjak et al. 2016, Møgelberg 2014, Bizjak & Møgelberg 2015) had a separate context. Here we simply include them in the context as if they were ordinary variables to simplify the presentation of the denotational semantics. Section 2.1 sketches an equivalence between **Core-GDTT** and the corresponding fragment of **GDTT**.

The rules for context formation, type judgements and equalities can be found in Figure 1. Note that clock has a special status. In particular, it is not a type. Its status is similar to that of the interval type in cubical type theory (Cohen et al. 2016). Ignoring  $\blacktriangleright$  and the clock irrelevance axiom, the type theory **Core-GDTT** is in fact just a fragment of a type theory with a base type clock in which types like  $\prod (x : A) . \text{clock}$  or  $\sum (\kappa : \text{clock}) . A$  are not allowed. Under this view, the type  $\forall \kappa . A$  can be thought of as a dependent product type  $\prod (\kappa : \text{clock}) . A$ , in fact its basic behaviour is exactly like a dependent product, as can be seen from the equality rules. We make use of this view to establish soundness of the model given in Section 3. What distinguishes it from an ordinary dependent product is the clock irrelevance axiom stated at the bottom of Figure 1. The set  $\text{fv}(A)$  is the set of free variables of  $A$  defined in the usual way, and so the assumption  $\kappa \notin \text{fv}(A)$  implies that  $\forall \kappa . A$  reduces to a simple function space  $\text{clock} \rightarrow A$ . The axiom states that all maps of this type are constant. In Section 4 we explain how to model the type theory with this additional axiom.

In Figure 1 the equalities should be understood as equalities of terms in a context. For brevity we have omitted the context in most statements except the clock irrelevance axiom, which, unlike the other rules, is type directed.

The term constructor  $\text{prev } \kappa$  is a restricted elimination form for  $\blacktriangleright^{\kappa}$ , and binds  $\kappa$ . An unrestricted eliminator  $\text{prev } \kappa$  would be unsafe, because terms of the form  $\text{fix}^{\kappa} x . \text{prev } \kappa . x$  would inhabit any type. As the model presented in this paper shows, however, it is safe to eliminate a  $\blacktriangleright^{\kappa}$ , as long as

Wellformed contexts

$$\frac{}{\cdot \vdash} \quad \frac{\Gamma \vdash A \text{ type} \quad x \notin \Gamma}{\Gamma, x : A \vdash} \quad \frac{\Gamma \vdash \quad \kappa \notin \Gamma}{\Gamma, \kappa : \text{clock} \vdash}$$

Wellformed clocks

$$\frac{\kappa : \text{clock} \in \Gamma}{\Gamma \vdash \kappa : \text{clock}}$$

Type formation

$$\frac{\Gamma, x : A \vdash B \text{ type}}{\Gamma \vdash \prod (x : A). B \text{ type}} \quad \frac{\Gamma, x : A \vdash B \text{ type}}{\Gamma \vdash \sum (x : A). B \text{ type}} \quad \frac{\Gamma \vdash A \text{ type} \quad \Gamma \vdash \kappa : \text{clock}}{\Gamma \vdash \blacktriangleright^{\kappa} A \text{ type}} \\ \frac{\Gamma, \kappa : \text{clock} \vdash A \text{ type}}{\Gamma \vdash \forall \kappa. A \text{ type}}$$

Typing judgements

$$\frac{}{\Gamma, x : A, \Gamma' \vdash x : A} \quad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x. t : \prod (x : A). B} \quad \frac{\Gamma \vdash t : \prod (x : A). B \quad \Gamma \vdash u : A}{\Gamma \vdash t u : B [u/x]} \\ \frac{\Gamma \vdash t : A \quad \Gamma \vdash u : B [t/x]}{\Gamma \vdash \langle t, u \rangle : \sum (x : A). B} \quad \frac{\Gamma \vdash t : \sum (x : A). B}{\Gamma \vdash \pi_1 t : A} \quad \frac{\Gamma \vdash t : \sum (x : A). B}{\Gamma \vdash \pi_2 t : B [\pi_1 t/x]} \\ \frac{\Gamma \vdash t : A \quad \Gamma \vdash \kappa : \text{clock}}{\Gamma \vdash \text{next}^{\kappa} t : \blacktriangleright^{\kappa} A} \quad \frac{\Gamma, \kappa : \text{clock} \vdash t : \blacktriangleright^{\kappa} A}{\Gamma \vdash \text{prev } \kappa. t : \forall \kappa. A} \quad \frac{\Gamma, x : \blacktriangleright^{\kappa} A \vdash t : A}{\Gamma \vdash \text{fix}^{\kappa} x. t : A} \\ \frac{\Gamma, \kappa : \text{clock} \vdash t : A}{\Gamma \vdash \Lambda \kappa. t : \forall \kappa. A} \quad \frac{\Gamma \vdash t : \forall \kappa. A \quad \Gamma \vdash \kappa' : \text{clock}}{\Gamma \vdash t[\kappa'] : A [\kappa'/\kappa]} \quad \frac{\Gamma \vdash t : A \quad \Gamma \vdash A = B}{\Gamma \vdash t : B}$$

Equalities

$$\begin{array}{ll} (\lambda x. t)u = t [u/x] & \lambda x. tx = t \quad (\text{if } x \notin t) \\ \pi_i \langle t_1, t_2 \rangle = t_i & \langle \pi_1 t, \pi_2 t \rangle = t \\ (\Lambda \kappa. t)\kappa' = t [\kappa'/\kappa] & \Lambda \kappa. t[\kappa] = t \quad (\text{if } \kappa \notin t) \\ \text{prev } \kappa. (\text{next}^{\kappa} t) = \Lambda \kappa. t & \text{next}^{\kappa} ((\text{prev } \kappa. t)[\kappa]) = t \\ \text{fix}^{\kappa} x. t = t [\text{next}^{\kappa} (\text{fix}^{\kappa} x. t)/x] & \end{array}$$

Clock irrelevance axiom

$$\frac{\Gamma \vdash t : \forall \kappa. A \quad \kappa \notin \text{fv}(A) \quad \Gamma \vdash \kappa' : \text{clock} \quad \Gamma \vdash \kappa'' : \text{clock}}{\Gamma \vdash t[\kappa'] = t[\kappa''] : A}$$

Figure 1. Syntax of Core-GDTT, a fragment of GDTT.



$\kappa$  does not appear in the ordinary (non-clock) variables of the context. This is ensured in the rule for  $\text{prev } \kappa$  by requiring that  $\kappa$  is at the end of the context. One might have expected a simpler rule of the form

$$\frac{\Gamma, \kappa : \text{clock} \vdash t : \blacktriangleright^{\kappa} A}{\Gamma, \kappa : \text{clock} \vdash \text{prev } \kappa.t : A}$$

but this rule is not closed under substitution of clock variables. This problem is solved by binding  $\kappa$ .

*Some example terms.* We refer to Bizjak et al. (2016) for more extensive and detailed motivation and explanation of the usage of the type theory. We briefly show here some example terms on streams. The type  $\text{Str}^{\kappa}$  of *guarded streams* of natural numbers is the unique type satisfying  $\text{Str}^{\kappa} = \mathbb{N} \times \blacktriangleright^{\kappa} \text{Str}^{\kappa}$ . To understand this example it is not important how this type can be defined, only that it satisfies the stated judgemental equality. For readers familiar with guarded dependent type theory we remark that it can be defined as using the guarded fixed point on the universe  $\mathbb{U}_{\{\kappa\}}$  as outlined in the introduction of this paper. Using the mentioned judgemental equality we can type

$$\begin{array}{ll} \text{head}^{\kappa} : \text{Str}^{\kappa} \rightarrow \mathbb{N} & \text{tail}^{\kappa} : \text{Str}^{\kappa} \rightarrow \blacktriangleright^{\kappa} \text{Str}^{\kappa} \\ \text{head}^{\kappa} \stackrel{\text{def}}{=} \lambda x s. \pi_1(xs) & \text{tail}^{\kappa} \stackrel{\text{def}}{=} \lambda x s. \pi_2(xs) \end{array}$$

Notice that the  $\text{tail}^{\kappa}$  introduces a  $\blacktriangleright^{\kappa}$  modality: The tail of a guarded stream is only available later. This prevents non-productive stream definitions. However once the streams are defined we wish to use them without introducing later modalities. This can be achieved by the type  $\text{Str}$  of *streams* of natural numbers. It is defined from the type of guarded streams as  $\text{Str} \stackrel{\text{def}}{=} \forall \kappa. \text{Str}^{\kappa}$ . Thus, the tail function on *streams* is defined as

$$\begin{array}{l} \text{tail} : \text{Str} \rightarrow \text{Str} \\ \text{tail} \stackrel{\text{def}}{=} \lambda x s. \text{prev } \kappa. \text{tail}^{\kappa}(xs[\kappa]) \end{array}$$

### 2.1. Relation to previous presentations

Judgements of GDTT as presented in (Bizjak et al. 2016) have a separate context for clock variables. For example, typing judgements have the form  $\Gamma \vdash_{\Delta} t : A$  where  $\Delta$  is a clock context of the form  $\kappa_1, \dots, \kappa_n$ , and  $\Gamma$  consists exclusively of ordinary variable declarations. The two presentations are equivalent in the sense that  $\Gamma \vdash_{\Delta} t : A$  is a valid judgement in the presentation of (Bizjak et al. 2016) iff  $\kappa_1 : \text{clock}, \dots, \kappa_n : \text{clock}, \Gamma \vdash t : A$  is valid in the presentation used here.

Another minor difference is that GDTT as presented in (Bizjak et al. 2016) has a clock constant  $\kappa_0$ . The clock constant can be easily added to Core-GDTT by a precompilation adding a fresh clock variable to the left of the context in each judgement.

## 3. A presheaf model

This section defines the category  $\text{Set}^{\mathbb{T}}$  as that of covariant presheaves on the category of *time objects*  $\mathbb{T}$ . As any presheaf category,  $\text{Set}^{\mathbb{T}}$  has enough structure to model dependent type theory. The category  $\text{Set}^{\mathbb{T}}$  contains an object  $\mathcal{C}$  of *clocks* which can be used to model clock quantification

and guarded recursion. We show that  $\text{Set}^{\mathbb{T}}$  validates almost all the rules of Core-GDTT, apart from the clock irrelevance axiom, which is the topic of Section 4. The focus in this section, as in most of the paper, will be to construct the semantic structure needed for modelling the type theory, leaving the question of how to interpret syntax to Section 8.

We write **Fin** for the category of finite sets and functions whose objects are finite subsets  $\mathcal{E}$  of some given, countably infinite, set of clocks.<sup>†</sup>

**Definition 3.1.** Let  $\mathbb{T}$  be the category with objects pairs  $(\mathcal{E}, \delta)$  with  $\mathcal{E} \in \mathbf{Fin}$  and  $\delta : \mathcal{E} \rightarrow \mathbb{N}$  a function. A morphism  $(\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')$  in  $\mathbb{T}$  is a *function*  $\tau : \mathcal{E} \rightarrow \mathcal{E}'$  in **Fin** such that  $\delta' \circ \tau \leq \delta$  in the pointwise ordering.

We use  $\lambda$  to range over elements of  $\mathcal{E}$  and write  $\mathcal{E}, \lambda$  for the union of  $\mathcal{E}$  with  $\{\lambda\}$  assuming  $\lambda \notin \mathcal{E}$ . Likewise, when  $\mathcal{E}$  and  $\mathcal{E}'$  are disjoint, we write  $\mathcal{E}, \mathcal{E}'$  for their union. We use the notation  $\delta[\lambda \mapsto n]$  for both the update of  $\delta$  (when  $\lambda \in \mathcal{E}$ ) and the extension of  $\delta$  (when  $\lambda \notin \mathcal{E}$ ).

The indexing category  $\mathbb{T}$  should be thought of as a category of *time objects*. A *time object* is a finite set of semantic clocks  $\mathcal{E}$  which each have a finite number of ticks left on them as indicated by  $\delta$ . During a computation, three things can happen: time can pass on the existing clocks, as captured by a map  $\text{id}_{\mathcal{E}} : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}, \delta')$  where  $\delta' \leq \delta$ , new clocks can be introduced as captured by set inclusions  $i : (\mathcal{E}, \delta) \rightarrow ((\mathcal{E}, \lambda), \delta[\lambda \mapsto n])$ , and clocks can be synchronised as captured by a map

$$\text{id}_{\mathcal{E}}[\lambda \mapsto \lambda'', \lambda' \mapsto \lambda''] : ((\mathcal{E}, \lambda, \lambda'), \delta[\lambda \mapsto n, \lambda' \mapsto m]) \rightarrow ((\mathcal{E}, \lambda''), \delta[\lambda'' \mapsto \min(n, m)]).$$

Finally, clocks can be renamed, e.g., via an isomorphism  $\sigma : \mathcal{E} \cong \mathcal{E}'$  inducing an isomorphism  $\sigma : (\mathcal{E}, \delta \circ \sigma) \rightarrow (\mathcal{E}', \delta)$ . Any map in the indexing category  $\mathbb{T}$  can be written as a composition of these three kinds of maps.

Variables of the form  $\kappa : \text{clock}$  will be modelled as the *object of clocks*  $\mathcal{C}$ , which is simply the first projection

$$\mathcal{C}(\mathcal{E}, \delta) = \mathcal{E}.$$

**Lemma 3.1.** Let  $\lambda$  be a clock. There is an isomorphism of objects of  $\text{Set}^{\mathbb{T}}$

$$\mathcal{C} \cong \varinjlim_{n \in \mathbb{N}} y(\{\lambda\}, n)$$

where  $y : \mathbb{T}^{\text{op}} \rightarrow \text{Set}^{\mathbb{T}}$  is the (co)Yoneda embedding, and we write  $(\{\lambda\}, n)$  for the  $\mathbb{T}$  object  $(\{\lambda\}, [\lambda \mapsto n])$ , i.e., the first component is the singleton containing  $\lambda$ , and the second component is the map which maps  $\lambda$  to  $n$ .

*Proof.* The objects of the diagram are

$$\begin{aligned} y(\{\lambda\}, n)(\mathcal{E}, \delta) &= \text{Hom}_{\mathbb{T}}((\{\lambda\}, n), (\mathcal{E}, \delta)) \\ &\cong \{\lambda' \in \mathcal{E} \mid \delta(\lambda') \leq n\} \end{aligned}$$

and up to this isomorphism, the arrows are inclusions of sets. Since colimits are computed pointwise in presheaf categories, the isomorphism follows.  $\square$

<sup>†</sup> The assumption that the objects are subset of a fixed set, as opposed to arbitrary finite sets keeps the category **Fin**, and thus also  $\mathbb{T}$ , small, thus simplifying definitions of, e.g., dependent products.

When describing objects and morphisms of  $\text{Set}^{\mathbb{T}}$  we will use the following notation: An object  $\Gamma$  is a family of sets  $\Gamma_{(\mathcal{E}, \delta)}$  indexed by  $(\mathcal{E}, \delta) \in \mathbb{T}$  together with maps

$$\sigma \cdot - : \Gamma_{(\mathcal{E}, \delta)} \rightarrow \Gamma_{(\mathcal{E}', \delta')}$$

for each  $\sigma : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')$  in  $\mathbb{T}$ , satisfying the following two functoriality properties

$$\text{id} \cdot x = x \tag{1}$$

$$(\sigma \circ \tau) \cdot x = \sigma \cdot (\tau \cdot x). \tag{2}$$

A morphism  $\rho : \Gamma \rightarrow \Gamma'$  is a family of maps  $\rho_{(\mathcal{E}, \delta)} : \Gamma_{(\mathcal{E}, \delta)} \rightarrow \Gamma'_{(\mathcal{E}, \delta)}$  such that  $\sigma \cdot (\rho_{(\mathcal{E}, \delta)}(\gamma)) = \rho_{(\mathcal{E}', \delta')}(\sigma \cdot \gamma)$  for any  $\sigma : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')$  in  $\mathbb{T}$  and any  $\gamma \in \Gamma_{(\mathcal{E}, \delta)}$ .

### 3.1. Interpreting type theory in categories of presheaves

We now recall the notion of category with families (CwF) (Dybjer 1995), which is a standard notion of model of dependent type theory. We also recall how  $\text{Set}^{\mathbb{T}}$  gives rise to a CwF modelling  $\Pi$ -, and  $\Sigma$ -types through a standard construction (Hofmann 1997) that works for all presheaf categories.

**Definition 3.2.** A category with families comprises

- A category  $\mathcal{C}$  with a distinguished terminal object
- For each object  $\Gamma$  of  $\mathcal{C}$  a set  $\mathcal{C}(\Gamma)$  of *families* over  $\Gamma$ .
- For each  $\Gamma$  in  $\mathcal{C}$  and each family  $A$  in  $\mathcal{C}(\Gamma)$  a set  $\mathcal{C}(\Gamma \vdash A)$  of *elements* of  $A$ .
- For each morphism  $\gamma : \Delta \rightarrow \Gamma$  in  $\mathcal{C}$  *reindexing* operations mapping  $A$  in  $\mathcal{C}(\Gamma)$  to  $A[\gamma]$  in  $\mathcal{C}(\Delta)$  and  $t$  in  $\mathcal{C}(\Gamma \vdash A)$  to  $t[\gamma]$  in  $\mathcal{C}(\Delta \vdash A[\gamma])$ . These must satisfy the equations  $A[\text{id}] = A$ ,  $t[\text{id}] = t$ ,  $A[\gamma \circ \delta] = A[\gamma][\delta]$  and  $t[\gamma \circ \delta] = t[\gamma][\delta]$  for all morphisms  $\delta$  with codomain  $\Delta$ .
- A *comprehension* operation associating to each family  $A$  in  $\mathcal{C}(\Gamma)$  the following: An object  $\Gamma.A$  in  $\mathcal{C}$ , a morphism  $p_A : \Gamma.A \rightarrow \Gamma$  and an element  $q_A$  in  $\mathcal{C}(\Gamma.A \vdash A[p_A])$ , such that for every  $\gamma : \Delta \rightarrow \Gamma$ , and  $t$  in  $\mathcal{C}(\Delta \vdash A[\gamma])$  there exists a unique morphism  $\langle \gamma, t \rangle : \Delta \rightarrow \Gamma.A$  such that  $p_A \circ \langle \gamma, t \rangle = \gamma$  and  $q_A[\langle \gamma, t \rangle] = t$ .

Note that uniqueness implies that  $\langle \gamma, t \rangle \circ \rho = \langle \gamma \circ \rho, t[\rho] \rangle$ .

We will often refer to a CwF simply by its underlying category  $\mathcal{C}$  leaving the rest of the structure implicit. Categories with families provide models of dependent type theories in which contexts are interpreted as objects in the underlying category, types are interpreted as families and terms as elements. The category  $\text{Set}^{\mathbb{T}}$  is the underlying category of a CwF whose families above an object  $\Gamma$  are families of sets  $A_{(\mathcal{E}, \delta)}(\gamma)$  indexed over  $(\mathcal{E}, \delta)$  in  $\mathbb{T}$  and  $\gamma \in \Gamma_{(\mathcal{E}, \delta)}$ , together with restriction maps

$$\sigma \cdot (-) : A_{(\mathcal{E}, \delta)}(\gamma) \rightarrow A_{(\mathcal{E}', \delta')}(\sigma \cdot \gamma)$$

indexed by  $\sigma : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')$  in  $\mathbb{T}$  and  $\gamma \in \Gamma_{(\mathcal{E}, \delta)}$  and satisfying the functoriality properties (1) and (2). Note that the notation  $\sigma \cdot x$  is overloaded both for a restriction of objects as well as families.

An element  $t \in \text{Set}^{\mathbb{T}}(\Gamma \vdash A)$  is a family of elements  $t_{(\mathcal{E}, \delta)}(\gamma) \in A_{(\mathcal{E}, \delta)}(\gamma)$  indexed over  $(\mathcal{E}, \delta)$  in  $\mathbb{T}$  and  $\gamma \in \Gamma_{(\mathcal{E}, \delta)}$  satisfying  $\sigma \cdot (t_{(\mathcal{E}, \delta)}(\gamma)) = t_{(\mathcal{E}', \delta')}(\sigma \cdot \gamma)$  for every  $\sigma : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')$ . Reindexing of terms and types along morphisms  $\rho : \Gamma' \rightarrow \Gamma$  is defined as  $A[\rho]_{(\mathcal{E}, \delta)}(\gamma) = A_{(\mathcal{E}, \delta)}(\rho_{(\mathcal{E}, \delta)}(\gamma))$  and

$t[\rho]_{(\mathcal{E}, \delta)}(\gamma) = t_{(\mathcal{E}, \delta)}(\rho_{(\mathcal{E}, \delta)}(\gamma))$ . We will often omit the subscripts  $(\mathcal{E}, \delta)$  when they can be inferred from the context.

Comprehension is defined as

$$(\Gamma.A)_{(\mathcal{E}, \delta)} = \{(\gamma, a) \mid \gamma \in \Gamma_{(\mathcal{E}, \delta)}, a \in A_{(\mathcal{E}, \delta)}(\gamma)\}$$

with presheaf action defined as  $\sigma \cdot (\gamma, x) = (\sigma \cdot \gamma, \sigma \cdot x)$ .

Recall the following standard result (Hofmann 1997).

**Lemma 3.2.** The CwF structure on  $\text{Set}^{\mathbb{T}}$  models  $\Pi$ - and  $\Sigma$ -types.

These are constructed as follows, for  $A \in \text{Set}^{\mathbb{T}}(\Gamma)$ ,  $B \in \text{Set}^{\mathbb{T}}(\Gamma.A)$  and  $\gamma \in \Gamma_{(\mathcal{E}, \delta)}$

$$\Pi(A, B)_{(\mathcal{E}, \delta)}(\gamma) = \left\{ (f_{\sigma})_{\sigma: (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')} \left| \begin{array}{l} \forall \mathcal{E}', \delta', \sigma : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta'), \forall a \in A_{(\mathcal{E}', \delta')}(\sigma \cdot \gamma), \\ f_{\sigma}(a) \in B_{(\mathcal{E}', \delta')}(\sigma \cdot \gamma, a) \text{ such that} \\ \tau \cdot f_{\sigma}(a) = f_{\tau \circ \sigma}(\tau \cdot a) \text{ for composable } \tau, \sigma \end{array} \right. \right\} \quad (3)$$

$$\Sigma(A, B)_{(\mathcal{E}, \delta)}(\gamma) = \{(a, b) \mid a \in A_{(\mathcal{E}, \delta)}(\gamma), b \in B_{(\mathcal{E}, \delta)}(\gamma, a)\} \quad (4)$$

with presheaf action on  $\Pi(A, B)$  defined by precomposition, i.e., if  $\tau : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')$  then

$$\tau \cdot ((f_{\sigma})_{\sigma: (\mathcal{E}, \delta) \rightarrow (\mathcal{E}'', \delta'')}) = (f_{\sigma\tau})_{\sigma: (\mathcal{E}', \delta') \rightarrow (\mathcal{E}'', \delta'')}$$

Recall also that evaluation mapping an element  $f \in \text{Set}^{\mathbb{T}}(\Gamma \vdash \Pi(A, B))$  and  $t \in \text{Set}^{\mathbb{T}}(\Gamma \vdash A)$  to  $\text{ev}(f, t) \in \text{Set}^{\mathbb{T}}(\Gamma \vdash B[\langle \text{id}_{\Gamma}, t \rangle])$  is defined as

$$\text{ev}(f, t)_{(\mathcal{E}, \delta)}(\gamma) = (f_{(\mathcal{E}, \delta)}(\gamma))_{\text{id}_{(\mathcal{E}, \delta)}}(t_{(\mathcal{E}, \delta)}(\gamma))$$

When  $A, B \in \text{Set}^{\mathbb{T}}(\Gamma)$  we write  $A \rightarrow B$  for  $\Pi(A, B[\text{p}])$ . When  $t \in \text{Set}^{\mathbb{T}}(\Gamma.A \vdash B)$  we write  $\lambda(t)$  for the corresponding abstracted element in  $\text{Set}^{\mathbb{T}}(\Gamma \vdash \Pi(A, B))$ . The semantic  $\beta$ -rule states that  $\text{ev}(\lambda t, u) = t[\langle \text{id}, u \rangle]$ . Finally, recall the substitution property  $\Pi(A, B)[\rho] = \Pi(A[\rho], B[\langle \text{p}\rho, \text{q} \rangle])$ , and similarly for  $\Sigma$ -types.

### 3.2. Modelling $\blacktriangleright$ and guarded recursion

We now explain how to model the  $\blacktriangleright$ -modality and fixed points. First note that there is a family  $\text{clock} \in \text{Set}^{\mathbb{T}}(1)$  defined as the object  $\mathcal{C}$ , since families in context 1 correspond to objects of  $\text{Set}^{\mathbb{T}}$ , and so, for any  $\Gamma$ , there is a family  $\text{clock}[\!_{\Gamma}] \in \text{Set}^{\mathbb{T}}(\Gamma)$ , where  $!_{\Gamma} : \Gamma \rightarrow 1$  is the unique map.

**Lemma 3.3.** If  $A \in \text{Set}^{\mathbb{T}}(\Gamma)$  and  $\kappa \in \text{Set}^{\mathbb{T}}(\Gamma \vdash \text{clock}[\!_{\Gamma}])$  there is a family  $\blacktriangleright^{\kappa} A \in \text{Set}^{\mathbb{T}}(\Gamma)$  and a mapping associating to each element  $t \in \text{Set}^{\mathbb{T}}(\Gamma \vdash A)$  an element  $\text{next}^{\kappa}(t) \in \text{Set}^{\mathbb{T}}(\Gamma \vdash \blacktriangleright^{\kappa} A)$  both commuting with reindexing, such that for every  $f \in \text{Set}^{\mathbb{T}}(\Gamma \vdash \blacktriangleright^{\kappa} A \rightarrow A)$  there is a unique  $\text{fix}^{\kappa}(f) \in \text{Set}^{\mathbb{T}}(\Gamma \vdash A)$  satisfying  $\text{ev}(f, \text{next}^{\kappa}(\text{fix}^{\kappa}(f))) = \text{fix}^{\kappa}(f)$ .

Note that the uniqueness here implies that the construction  $\text{fix}^{\kappa}(f)$  commutes with reindexing: Since

$$\begin{aligned} \text{ev}(f[\rho], \text{next}^{\kappa[\rho]}((\text{fix}^{\kappa}(f))[\rho])) &= \text{ev}(f[\rho], (\text{next}^{\kappa}(\text{fix}^{\kappa}(f)))[\rho]) \\ &= \text{ev}(f, \text{next}^{\kappa}(\text{fix}^{\kappa}(f)))[\rho] \\ &= (\text{fix}^{\kappa}(f))[\rho] \end{aligned}$$

uniqueness implies  $(\text{fix}^{\kappa}(f))[\rho] = \text{fix}^{\kappa[\rho]}(f[\rho])$ .

*Proof.* If  $(\mathcal{E}, \delta)$  is an object of  $\mathbb{T}$  and  $\lambda \in \mathcal{E}$  such that  $\delta(\lambda) > 0$  we write  $\delta^{-\lambda}$  for the function which agrees with  $\delta$  everywhere except on  $\lambda$  where  $\delta^{-\lambda}(\lambda) = \delta(\lambda) - 1$ . It is elementary that the identity function defines a morphism

$$\text{tick}^\lambda : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}, \delta^{-\lambda})$$

in  $\mathbb{T}$ .

With this notation we can define  $\blacktriangleright^\kappa A$  as follows, omitting the subscript on  $\kappa$

$$(\blacktriangleright^\kappa A)_{(\mathcal{E}, \delta)}(\gamma) = \begin{cases} \{\star\} & \text{if } \delta(\kappa(\gamma)) = 0 \\ A_{(\mathcal{E}, \delta^{-\kappa(\gamma)})}(\text{tick}^{\kappa(\gamma)} \cdot \gamma) & \text{otherwise} \end{cases}$$

Let  $\sigma : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')$ . The map  $\sigma \cdot (-) : (\blacktriangleright^\kappa A)_{(\mathcal{E}, \delta)}(\gamma) \rightarrow (\blacktriangleright^\kappa A)_{(\mathcal{E}', \delta')}(\sigma \cdot \gamma)$ , can be defined in the case that  $\delta'(\kappa(\sigma \cdot \gamma)) = 0$  as  $\sigma \cdot x = \star$ . If  $\delta'(\kappa(\sigma \cdot \gamma)) > 0$  also  $\delta(\kappa(\gamma)) > 0$  because

$$\delta'(\kappa(\sigma \cdot \gamma)) = \delta'(\sigma \cdot \kappa(\gamma)) = \delta'(\sigma(\kappa(\gamma))) \leq \delta(\kappa(\gamma))$$

and so  $\sigma$  induces a map  $\sigma^{-\kappa(\gamma)} : (\mathcal{E}, \delta^{-\kappa(\gamma)}) \rightarrow (\mathcal{E}', \delta'^{-\sigma(\kappa(\gamma))})$ , satisfying  $\sigma^{-\kappa(\gamma)} \circ \text{tick}^{\kappa(\gamma)} = \text{tick}^{\sigma(\kappa(\gamma))} \circ \sigma$ . In this case, we can thus define  $\sigma \cdot (-)$  to be the map

$$\sigma^{-\kappa(\gamma)} \cdot (-) : A_{(\mathcal{E}, \delta^{-\kappa(\gamma)})}(\text{tick}^{\kappa(\gamma)} \cdot \gamma) \rightarrow A_{(\mathcal{E}', \delta'^{-\sigma(\kappa(\gamma))})}(\text{tick}^{\sigma(\kappa(\gamma))} \cdot \sigma \cdot \gamma)$$

The construction  $\blacktriangleright^\kappa$  commutes with reindexing, since

$$\begin{aligned} (\blacktriangleright^{\kappa[\rho]} A[\rho]) &= \begin{cases} \{\star\} & \text{if } \delta(\kappa[\rho](\gamma)) = 0 \\ A_{(\mathcal{E}, \delta^{-\kappa[\rho](\gamma)})}(\rho(\text{tick}^{\kappa[\rho](\gamma)} \cdot \gamma)) & \text{otherwise} \end{cases} \\ &= \begin{cases} \{\star\} & \text{if } \delta(\kappa(\rho(\gamma))) = 0 \\ A_{(\mathcal{E}, \delta^{-\kappa(\rho(\gamma))})}(\text{tick}^{\kappa(\rho(\gamma))} \cdot \rho(\gamma)) & \text{otherwise} \end{cases} \end{aligned}$$

and writing out  $((\blacktriangleright^{\kappa[\rho]} A[\rho])(\gamma)) = ((\blacktriangleright^\kappa A)(\rho(\gamma)))$  gives the exact same expression.

Analogously, the element  $\text{next}^\kappa(t)$  is defined as

$$(\text{next}^\kappa(t))_{(\mathcal{E}, \delta)}(\gamma) = \begin{cases} \star & \text{if } \delta(\kappa(\gamma)) = 0 \\ t_{(\mathcal{E}, \delta^{-\kappa(\gamma)})}(\text{tick}^{\kappa(\gamma)} \cdot \gamma) & \text{otherwise} \end{cases}$$

To define  $\text{fix}^\kappa(f)$ , note that by the above definitions

$$\text{ev}(f, \text{next}^\kappa(\text{fix}^\kappa(f)))_{(\mathcal{E}, \delta)}(\gamma) = \begin{cases} (f(\rho))_{\text{id}}(\star) & \text{if } \delta(\kappa(\rho(\gamma))) = 0 \\ (f(\rho))_{\text{id}}((\text{fix}^\kappa(f))(\text{tick}^{\kappa(\gamma)} \cdot \gamma)) & \text{otherwise} \end{cases}$$

and thus the  $\text{fix}^\kappa(f)$  can be defined by induction on  $\kappa(\rho(\gamma))$ .  $\square$

### 3.3. Modelling previous

As noted in Section 2, universal quantification over clocks is simply a special case of a dependent function space, and so can be modelled in the CwF  $\text{Set}^{\mathbb{T}}$  using  $\Pi$ -types. No special construction is needed for this. However, in order to model  $\text{prev}$  we now give an alternative description of  $\Pi$  types with domain clock in the model as a limit over a family of objects indexed by natural numbers. Universal quantification over clocks is modelled similarly in the models of (Atkey & McBride 2013, Møgelberg 2014, Bizjak & Møgelberg 2015).

In the following, we will assume a choice of fresh clock names  $\mathcal{E} \mapsto \lambda_{\mathcal{E}}$ , such that  $\lambda_{\mathcal{E}} \notin \mathcal{E}$  and write  $\iota^n : (\mathcal{E}, \delta) \rightarrow ((\mathcal{E}, \lambda_{\mathcal{E}}), \delta[\lambda_{\mathcal{E}} \mapsto n])$  for the inclusion for  $n \in \mathbb{N}$ . Note that  $\text{tick}^\lambda \circ \iota^{n+1} = \iota^n$ .

**Lemma 3.4.** Let  $A \in \text{Set}^{\mathbb{T}}(\Gamma.\text{clock}[\Gamma])$ ,  $(\mathcal{E}, \delta) \in \mathbb{T}$  and  $\gamma \in \Gamma_{(\mathcal{E}, \delta)}$ . The set  $\Pi(\text{clock}[\Gamma], A)_{(\mathcal{E}, \delta)}(\gamma)$  is the limit of the diagram

$$A(\iota^0 \cdot \gamma, \lambda_{\mathcal{E}}) \xleftarrow{\text{tick}^{\lambda_{\mathcal{E}}} \cdot (-)} A(\iota^1 \cdot \gamma, \lambda_{\mathcal{E}}) \xleftarrow{\text{tick}^{\lambda_{\mathcal{E}}} \cdot (-)} A(\iota^2 \cdot \gamma, \lambda_{\mathcal{E}}) \xleftarrow{\text{tick}^{\lambda_{\mathcal{E}}} \cdot (-)} \dots$$

*Proof.* Let  $y : \mathbb{T}^{\text{op}} \rightarrow \text{Set}^{\mathbb{T}}$  be the (co)Yoneda embedding. Uncurrying the definition in (3) we see that elements of  $\Pi(\text{clock}[\Gamma], A)_{(\mathcal{E}, \delta)}(\gamma)$  correspond to maps mapping objects  $(\mathcal{E}', \delta')$  of  $\mathbb{T}$  and elements  $(\sigma, \lambda) \in (y(\mathcal{E}, \delta) \times \mathcal{C})(\mathcal{E}', \delta')$  to elements in  $A(\sigma \cdot \gamma, \lambda)$  naturally in  $(\mathcal{E}', \delta')$ . By Lemma 3.1, the presheaf  $y(\mathcal{E}, \delta) \times \mathcal{C}$  is isomorphic to the colimit over  $n$  of the diagram given by objects  $y(\mathcal{E}, \delta) \times y(\{\lambda_{\mathcal{E}}\}, n)$ . Thus  $\Pi(\text{clock}[\Gamma], A)_{(\mathcal{E}, \delta)}(\gamma)$  is isomorphic to the limit of a diagram of the form

$$X_0 \longleftarrow X_1 \longleftarrow X_2 \longleftarrow X_3 \longleftarrow \dots$$

where  $X_n$  is the set of maps as above defined just for  $(\sigma, \lambda) \in (y(\mathcal{E}, \delta) \times y(\{\lambda_{\mathcal{E}}\}, n))(\mathcal{E}', \delta')$ , and the maps are given by restriction. It remains to show the isomorphism of the above diagram with that of the lemma.

The object  $((\mathcal{E}, \lambda_{\mathcal{E}}), \delta[\lambda_{\mathcal{E}} \mapsto n])$  is the coproduct in  $\mathbb{T}$  of  $(\mathcal{E}, \delta)$  and  $(\{\lambda_{\mathcal{E}}\}, n)$  with inclusions given by inclusions of sets. Since the yoneda embedding preserves products,  $y(\mathcal{E}, \delta) \times y(\{\lambda_{\mathcal{E}}\}, n) \cong y((\mathcal{E}, \lambda_{\mathcal{E}}), \delta[\lambda_{\mathcal{E}} \mapsto n])$ . Up to this correspondence, the restriction of an element in the family  $\Pi(\text{clock}[\Gamma], A)_{(\mathcal{E}, \delta)}(\gamma)$  to  $y(\mathcal{E}, \delta) \times y(\{\lambda_{\mathcal{E}}\}, n)$  corresponds to a mapping associating to  $(\mathcal{E}', \delta')$  of  $\mathbb{T}$  and elements  $(\sigma, \lambda) \in y((\mathcal{E}, \lambda_{\mathcal{E}}), \delta[\lambda_{\mathcal{E}} \mapsto n])(\mathcal{E}', \delta')$  elements in  $A(\sigma \cdot (\iota^n \cdot \gamma), \sigma(\lambda_{\mathcal{E}}))$  naturally in  $(\mathcal{E}', \delta')$ . By a yoneda style argument such mappings are determined by their action on the identity on  $((\mathcal{E}, \lambda_{\mathcal{E}}), \delta[\lambda_{\mathcal{E}} \mapsto n])$  and thus we arrive at the diagram of the lemma.  $\square$

Rather than modeling `prev` directly, we model the construct

$$\frac{\Gamma \vdash t : \forall \kappa. \blacktriangleright A}{\Gamma \vdash \text{force } t : \forall \kappa. A}$$

Using this, one can define `prev` as

$$\text{prev } \kappa.t \stackrel{\text{def}}{=} \text{force}(\Lambda \kappa.t) \tag{5}$$

To satisfy the equalities of Figure 1, the term  $\lambda x. \text{force}(x)$  should be an inverse to  $\lambda x. \Lambda \kappa. \text{next}^\kappa(x[\kappa])$ . Using this, one can prove the first equality for `force` in Figure 1 as follows

$$\begin{aligned} \text{prev } \kappa. (\text{next}^\kappa t) &= \text{force}(\Lambda \kappa. (\text{next}^\kappa t)) \\ &= \text{force}(\Lambda \kappa. (\text{next}^\kappa (\Lambda \kappa.t)[\kappa])) \\ &= \Lambda \kappa.t \end{aligned}$$

The other equality is proved similarly.

We now show that the semantic correspondent to  $\lambda x. \Lambda \kappa. \text{next}^\kappa(x[\kappa])$  is an isomorphism.

**Lemma 3.5.** Suppose  $A \in \text{Set}^{\mathbb{T}}(\Gamma.\text{clock}[\Gamma])$ . The mapping of elements  $t \in \text{Set}^{\mathbb{T}}(\Gamma \vdash \Pi(\text{clock}[\Gamma], A))$  to  $\lambda(\text{next}^q(\text{ev}(t[p], q)))$  in  $\text{Set}^{\mathbb{T}}(\Gamma \vdash \Pi(\text{clock}[\Gamma], \blacktriangleright A))$  is an isomorphism.

Before proving this, we argue that the mapping referred to is welltyped. By the assumption on  $t$ ,  $t[p] \in \text{Set}^{\mathbb{T}}(\Gamma.\text{clock}[\Gamma] \vdash (\Pi(\text{clock}[\Gamma], A))[p])$  and since  $(\Pi(\text{clock}[\Gamma], A))[p]$  equals  $\Pi(\text{clock}[\Gamma.\text{clock}[\Gamma]], A[\langle p \circ p, q \rangle])$  also  $\text{ev}(t[p], q)$  is an element in  $A[\langle p \circ p, q \rangle][[\text{id}, q]] = A[\langle p, q \rangle] = A$ . Since  $q \in \text{Set}^{\mathbb{T}}(\Gamma.\text{clock}[\Gamma] \vdash)$  also  $\text{next}^q(\text{ev}(t[p], q))$  is an element in  $\mathbf{q}A$ , and therefore  $\lambda(\text{next}^q(\text{ev}(t[p], q)))$  is in  $\text{Set}^{\mathbb{T}}(\Gamma \vdash \Pi(\text{clock}[\Gamma], \mathbf{q}A))$ . Note that the inverse of this map necessary must commute with reindexing, since the construction of the map does.

*Proof.* Unfolding definitions, we see that the construction of lemma at  $\gamma \in \Gamma_{(\mathcal{E}, \delta)}$  is the map induced by the map of diagrams below.

$$\begin{array}{ccccccc}
 A(\iota^0 \cdot \gamma, \lambda_{\mathcal{E}}) & \xleftarrow{\text{tick}^{\lambda_{\mathcal{E}} \cdot (-)}} & A(\iota^1 \cdot \gamma, \lambda_{\mathcal{E}}) & \xleftarrow{\text{tick}^{\lambda_{\mathcal{E}} \cdot (-)}} & A(\iota^2 \cdot \gamma, \lambda_{\mathcal{E}}) & \xleftarrow{\text{tick}^{\lambda_{\mathcal{E}} \cdot (-)}} & \dots \\
 \downarrow ! & & \downarrow \text{tick}^{\lambda_{\mathcal{E}} \cdot (-)} & & \downarrow \text{tick}^{\lambda_{\mathcal{E}} \cdot (-)} & & \\
 1 & \longleftarrow & A(\iota^0 \cdot \gamma, \lambda_{\mathcal{E}}) & \xleftarrow{\text{tick}^{\lambda_{\mathcal{E}} \cdot (-)}} & A(\iota^1 \cdot \gamma, \lambda_{\mathcal{E}}) & \xleftarrow{\text{tick}^{\lambda_{\mathcal{E}} \cdot (-)}} & \dots
 \end{array}$$

The map induced between the limits is therefore an isomorphism.  $\square$

With these definitions we can extend the interpretation to the whole of **Core-GDTT**. However the interpretation only validates the basic axioms, i.e.,  $\beta$  and  $\eta$  laws. It does not validate the clock irrelevance axiom. To soundly interpret **Core-GDTT** we need to require that the families are suitably constant. This is the subject of the next section.

#### 4. Modelling clock irrelevance using orthogonality

In the interpretation above  $\forall \kappa. A$  is interpreted as an ordinary dependent product  $\prod (\kappa : \text{clock}) . A$ . Under this interpretation, the clock irrelevance axiom concerns functions  $f$  of type  $\mathcal{C} \rightarrow A$  and states that each such function must be constant. To model this, we restrict attention in the model to those families satisfying this property, and show that the collection of these is closed under the type constructions of **Core-GDTT**. To capture clock irrelevance semantically, we start by recalling the category theoretic concept of orthogonality.

A morphism  $e : A \rightarrow B$  is *left-orthogonal* to  $m : C \rightarrow D$  (and  $m$  is right-orthogonal to  $e$ ) if all commutative squares as below have a *unique* filler  $h$ .

$$\begin{array}{ccc}
 A & \xrightarrow{f} & C \\
 \downarrow e & \nearrow h & \downarrow m \\
 B & \xrightarrow{g} & D
 \end{array}$$

Often we will simply refer to this as  $e$  being orthogonal to  $m$ . If  $B$  is the terminal object, we may also refer to this as the object  $A$  being left-orthogonal to  $m$  and similarly for the case of  $D$  being terminal. We shall need the slightly stronger notion of *internal orthogonality* (?), which can be understood by rephrasing the above lifting property as the requirement that the following

diagram of hom-sets is a pullback

$$\begin{array}{ccc} \mathrm{Hom}_{\mathbb{C}}(B, C) & \xrightarrow{m \circ (-)} & \mathrm{Hom}_{\mathbb{C}}(B, D) \\ \downarrow (-) \circ e & & \downarrow (-) \circ e \\ \mathrm{Hom}_{\mathbb{C}}(A, C) & \xrightarrow{m \circ (-)} & \mathrm{Hom}_{\mathbb{C}}(A, D) \end{array}$$

The idea of internal orthogonality is to replace the external hom-sets above with exponentials in a cartesian closed category. The resulting condition is equivalent to the following, which can be stated also in categories that are not cartesian closed.

**Definition 4.1.** Let  $\mathbb{C}$  be a category with finite products. Say a morphism  $p : A \rightarrow B$  is *internally right orthogonal to an object  $X$*  if for any  $Y$  and any  $f, g$  making the outer square below commute, there exists a *unique*  $h : Y \rightarrow A$  such that the diagram

$$\begin{array}{ccc} Y \times X & \xrightarrow{f} & A \\ \downarrow \pi_Y & \searrow h & \downarrow p \\ Y & \xrightarrow{g} & B \end{array}$$

commutes.

A map  $p : A \rightarrow B$  in  $\mathrm{Set}^{\mathbb{T}}$  is *invariant under clock introduction* if it is internally right orthogonal to any object of the form  $y(\{\lambda\}, n)$ .

**Definition 4.2.** A family  $A \in \mathrm{Set}^{\mathbb{T}}(\Gamma)$  is *invariant under clock introduction* if  $p : \Gamma.A \rightarrow \Gamma$  is invariant under clock introduction in the sense of Definition 4.1.

The terminology of being invariant under clock introduction is justified by the following lemma, the proof of which is on page 17 after preliminary Lemmas 4.2 and 4.3.

**Lemma 4.1.** A morphism  $p : A \rightarrow B$  in  $\mathrm{Set}^{\mathbb{T}}$  is invariant under clock introduction if and only if for all  $(\mathcal{E}, \delta) \in \mathbb{T}$ , and any (equivalently all)  $\lambda \notin \mathcal{E}$  and any  $n$  the square

$$\begin{array}{ccc} A(\mathcal{E}, \delta) & \xrightarrow{A(\iota^n)} & A((\mathcal{E}, \lambda), \delta[\lambda \mapsto n]) \\ \downarrow p_{(\mathcal{E}, \delta)} & \lrcorner & \downarrow p_{((\mathcal{E}, \lambda), \delta[\lambda \mapsto n])} \\ B(\mathcal{E}, \delta) & \xrightarrow{B(\iota^n)} & B((\mathcal{E}, \lambda), \delta[\lambda \mapsto n]) \end{array}$$

is a pullback, where  $\iota^n : (\mathcal{E}, \delta) \rightarrow ((\mathcal{E}, \lambda), \delta[\lambda \mapsto n])$  is the inclusion.

In particular, for any presheaf  $A$ , the unique map  $A \rightarrow 1$  is invariant under clock introduction iff  $A$  is a constant presheaf. It will be an invariant of the interpretation defined here that the interpretation of any type is invariant under clock introduction.

Lemma 4.1 can be restated in the following way for interpretations of types.

**Corollary 4.1.** A family  $A \in \mathrm{Set}^{\mathbb{T}}(\Gamma)$  is invariant under clock introduction if and only if for any  $\mathcal{E}$ , any  $\lambda \notin \mathcal{E}$ , any inclusion  $\iota^n : (\mathcal{E}, \delta) \rightarrow ((\mathcal{E}, \lambda), \delta[\lambda \mapsto n])$ , and any  $\gamma \in \Gamma_{(\mathcal{E}, \delta)}$ , the action

$$\iota^n \cdot (-) : A_{(\mathcal{E}, \delta)}(\gamma) \rightarrow A_{((\mathcal{E}, \lambda), \delta[\lambda \mapsto n])}(\iota^n \cdot \gamma) \quad (6)$$

is an isomorphism.



The proof of Lemma 4.1 uses the characterisation of internal orthogonality in Lemma 4.2 together with the characterisation of exponentiation with certain representable functors in Lemma 4.3.

The following lemma is proved by a straightforward diagram chase.

**Lemma 4.2.** Suppose  $\mathbb{C}$  is cartesian closed,  $X$  is an object of  $\mathbb{C}$  and  $p : A \rightarrow B$  a morphism. Then  $p$  is internally right orthogonal to  $X$  if and only if

$$\begin{array}{ccc} A & \xrightarrow{c_A} & A^X \\ \downarrow p & \lrcorner & \downarrow p^X \\ B & \xrightarrow{c_B} & B^X \end{array}$$

is a pullback. Here  $c_A$  and  $c_B$  are exponential transposes of projections  $A \times X \rightarrow A$  and  $B \times X \rightarrow B$  and  $p^X$  is postcomposition with  $p$ .

By the pullback lemma (MacLane 1998, Exercise III.4.8), we derive the following corollary.

**Corollary 4.2.** If the morphisms  $p \circ q$  and  $p$  are internally right orthogonal to  $X$  then so is  $q$ .

**Lemma 4.3.** Let  $A$  be an object of  $\text{Set}^{\mathbb{T}}$ . Let  $\lambda$  be a clock and  $n \in \mathbb{N}$ . As in Lemma 3.1 we write simply  $n$  for the map  $\{\lambda\} \rightarrow \mathbb{N}$  mapping  $\lambda$  to  $n$ . Then

$$A^{y(\{\lambda\}, n)}(\mathcal{E}, \delta) \cong A((\mathcal{E}, \lambda_{\mathcal{E}}), \delta[\lambda_{\mathcal{E}} \mapsto n])$$

and up to this isomorphism,  $c_A = A(\iota)$ , where  $\iota : (\mathcal{E}, \delta) \rightarrow ((\mathcal{E}, \lambda_{\mathcal{E}}), \delta[\lambda_{\mathcal{E}} \mapsto n])$  is the inclusion.

*Proof.* In  $\mathbb{T}$ , the object  $((\mathcal{E}, \lambda_{\mathcal{E}}), \delta[\lambda_{\mathcal{E}} \mapsto n])$  is a coproduct of  $(\mathcal{E}, \delta)$  and  $(\{\lambda\}, n)$  with coproduct inclusions given by set inclusions (mapping  $\lambda$  to  $\lambda_{\mathcal{E}}$ ). Since  $y : \mathbb{T}^{\text{op}} \rightarrow \text{Set}^{\mathbb{T}}$  preserves products, we get the following series of isomorphisms using the Yoneda lemma and standard definitions of exponentials in presheaf categories:

$$\begin{aligned} A^{y(\{\lambda\}, n)}(\mathcal{E}, \delta) &= \text{Hom}(y(\mathcal{E}, \delta) \times y(\{\lambda\}, n), A) \\ &\cong \text{Hom}(y((\mathcal{E}, \lambda_{\mathcal{E}}), \delta[\lambda_{\mathcal{E}} \mapsto n]), A) \\ &\cong A((\mathcal{E}, \lambda_{\mathcal{E}}), \delta[\lambda_{\mathcal{E}} \mapsto n]) \end{aligned}$$

The morphism  $c_A$  maps  $x \in A(\mathcal{E}, \delta)$  to the natural transformation given by the composition of the first projection  $y(\mathcal{E}, \delta) \times y(\{\lambda\}, n) \rightarrow y(\mathcal{E}, \delta)$  and the morphism  $y(\mathcal{E}, \delta) \rightarrow A$  corresponding to  $x$  under the Yoneda lemma. Since the projection corresponds to composition with  $\iota$ , the second statement of the lemma follows.  $\square$

*Proof of Lemma 4.1* Follows from Lemma 4.2 and Lemma 4.3.  $\square$

When interpreting syntax dependent types will be interpreted as families invariant under clock introduction. This will be used to prove soundness of the clock irrelevance axiom. In fact, just to prove that, it would be enough that the interpretation of every type is internally right orthogonal to  $\mathcal{C}$ . This is a slightly weaker statement than being invariant under clock introduction, as the next lemma states. We have chosen to work with the latter because of the natural characterisation of Lemma 4.1.

**Lemma 4.4.** Let  $\mathbb{C}$  be a cartesian closed category  $\mathbb{C}$  and let  $X = \varinjlim_i X_i$  be a connected colimit. If  $p : A \rightarrow B$  is internally right orthogonal to all  $X_i$ , then it is also internally right orthogonal

to  $X$ . As a consequence, if  $p$  is invariant under clock introduction, it is also internally right orthogonal to  $\mathcal{C}$ .

The second statement of the lemma follows from the first by Lemma 3.1.

The notion of internal orthogonality can be shown to be equivalent to the one used by Hyland et al. (1990), and the next lemma follows from (Hyland et al. 1990, Proposition 2.1). Rather than proving this equivalence, we give here a direct proof.

**Proposition 4.1.** Suppose  $\mathbb{C}$  is a locally cartesian closed category and  $X$  is an object in  $\mathbb{C}$ . The notion of being internally right orthogonal to  $X$  is then closed under composition, pullback (along *arbitrary* maps), dependent products (along *arbitrary* maps) and all isomorphisms are internally right orthogonal to  $X$ .

*Proof.* Closure under composition and the fact that isomorphisms are internally right orthogonal to  $X$  follow straightforwardly from Lemma 4.2.

To prove the statement for pullbacks, suppose  $p : B \rightarrow D$  is internally right orthogonal to  $X$  and  $q : A \rightarrow C$  is the pullback of  $p$  along some map  $g$  not assumed to be internally right orthogonal to  $X$ . By the pullback pasting lemma then the outer square below is a pullback.

$$\begin{array}{ccccc} A & \xrightarrow{f} & B & \xrightarrow{c_B} & B^X \\ q \downarrow & \lrcorner & \downarrow p & \lrcorner & \downarrow p^X \\ C & \xrightarrow{g} & D & \xrightarrow{c_D} & D^X \end{array}$$

By naturality of  $c$ , the below outer square is equal to the one above, and thus also a pullback.

$$\begin{array}{ccccc} A & \xrightarrow{c_A} & A^X & \xrightarrow{f^X} & B^X \\ q \downarrow & & \downarrow q^X & & \downarrow p^X \\ C & \xrightarrow{c_C} & C^X & \xrightarrow{g^X} & D^X \end{array} \quad (7)$$

Since  $-^X$  has a left adjoint it preserves pullbacks and so right square of (7) is a pullback. By the pullback lemma, also the left square is a pullback, and thus  $q$  is internally right orthogonal to  $X$ .

For dependent products, suppose  $p : A \rightarrow B$  is internally right orthogonal to  $X$ , and  $f : B \rightarrow C$ . We must show that  $\Pi_f(p)$  is internally right orthogonal to  $X$ , where  $\Pi_f : \mathbb{C}/B \rightarrow \mathbb{C}/C$  is the right adjoint to pullback along  $f$ . We write  $f^*(h) : B \times_C Y \rightarrow B$  for the result of applying the pullback functor to an object  $h : Y \rightarrow C$  of  $\mathbb{C}/C$  and use the notation

$$\widehat{(-)} : \text{Hom}_{\mathbb{C}/C}(h, \Pi_f(p)) \rightarrow \text{Hom}_{\mathbb{C}/B}(f^*(h), p)$$

for the isomorphism of hom-sets, given  $h : Y \rightarrow C$ .

Given  $Y, h, k$  as in the outer square on the left below, by naturality, the isomorphism  $\widehat{(-)}$  extends to a bijective correspondence of diagonal fillers in the following two diagrams.

$$\begin{array}{ccc} Y \times X & \xrightarrow{k} & \Pi_B A \\ \pi_Y \downarrow & \nearrow & \downarrow \Pi_f(p) \\ Y & \xrightarrow{h} & C \end{array} \quad \begin{array}{ccc} B \times_C (Y \times X) & \xrightarrow{\widehat{k}} & A \\ B \times_C \pi_Y \downarrow & \nearrow & \downarrow p \\ B \times_C Y & \xrightarrow{f^*(h)} & B \end{array} \quad (8)$$

where  $B \times_C \pi_Y$  is the pullback functor applied to the morphism  $\pi_Y : (h \circ \pi_Y) \rightarrow h$  in  $\mathbb{C}/C$ .

By the pullback pasting lemma, the following outer diagram is a pullback

$$\begin{array}{ccc}
 (B \times_C Y) \times X & \xrightarrow{\pi_Y^C \times \text{id}_X} & Y \times X \\
 \pi_{(B \times_C Y)} \downarrow & \lrcorner & \downarrow \pi_Y \\
 B \times_C Y & \xrightarrow{\pi_Y^C} & Y \\
 f^*(h) \downarrow & \lrcorner & \downarrow h \\
 B & \xrightarrow{f} & C
 \end{array}$$

From this we conclude that there is an isomorphism  $\phi : (B \times_C Y) \times X \cong B \times_C (Y \times X)$ . An easy diagram chase verifies (using the universal property of the lower diagram above) that

$$(B \times_C \pi_Y) \circ \phi = \pi_{B \times_C Y} : (B \times_C Y) \times X \rightarrow (B \times_C Y).$$

Thus, the fillers of (8) are in bijective correspondence with the fillers of

$$\begin{array}{ccc}
 (B \times_C Y) \times X & \xrightarrow{\widehat{k} \circ \phi} & A \\
 \pi_{B \times_C Y} \downarrow & \searrow \text{dotted} & \downarrow p \\
 B \times_C Y & \xrightarrow{f^*(h)} & B
 \end{array}$$

Since  $p$  is assumed to be internally right orthogonal to  $X$ , there is a unique filler of the diagram above, and thus a unique filler of the left diagram of (8). This proves that  $\Pi_f(p)$  is internally right orthogonal to  $X$  as desired.  $\square$

**Corollary 4.3.** In the CwF structure of  $\text{Set}^{\mathbb{T}}$ , the collection of families invariant under clock introduction is closed under the operations for taking  $\Pi$ -, and  $\Sigma$ -types as well as reindexing.

**Lemma 4.5.** If  $\kappa \in \text{Set}^{\mathbb{T}}(\Gamma \vdash \text{clock}[!_{\Gamma}])$  and the family  $A \in \text{Set}^{\mathbb{T}}(\Gamma)$  is invariant under clock introduction then  $\blacktriangleright^{\kappa} A$  is invariant under clock introduction.

*Proof.* The map  $\iota \cdot (-)$  is defined to be the identity on  $\{\star\}$  in the case of  $\delta(\kappa(\gamma)) = 0$ . In the case of  $\delta(\kappa(\gamma)) > 0$  it is defined as the action  $\sigma^{-\kappa(\gamma)}(-)$  on  $A$ . By the assumption the latter is always an isomorphism and thus so is  $\iota \cdot (-)$ .  $\square$

We now show that invariance under clock introduction implies the soundness of the clock irrelevance axiom.

**Lemma 4.6.** Suppose  $A$  in  $\text{Set}^{\mathbb{T}}(\Gamma)$  is invariant under clock introduction and that  $t$  is in  $\text{Set}^{\mathbb{T}}(\Gamma \vdash \Pi(\text{clock}[!_{\Gamma}], A[p]))$  and  $\kappa, \kappa' \in \text{Set}^{\mathbb{T}}(\Gamma \vdash \text{clock}[!_{\Gamma}])$ . Then  $\text{ev}(t, \kappa) = \text{ev}(t, \kappa')$ .

*Proof.* Note first that  $\text{ev}(t[p], q) \in \text{Set}^{\mathbb{T}}(\Gamma.\text{clock}[!_{\Gamma}] \vdash A[p])$ . Since  $\Gamma.\text{clock}[!_{\Gamma}] = \Gamma \times C$ , this gives

us the commutative outer diagram below.

$$\begin{array}{ccc}
 \Gamma \times \mathcal{C} & \xrightarrow{\langle p, \text{ev}(t[p], q) \rangle} & \Gamma.A \\
 \downarrow p & \searrow u & \downarrow p \\
 \Gamma & \xrightarrow{\text{id}} & \Gamma
 \end{array}$$

Since  $p : \Gamma.A \rightarrow \Gamma$  is internally right orthogonal to  $\mathcal{C}$  by Lemma 4.4, there is a unique lifting  $u$  as indicated in the diagram.

Now,

$$\begin{aligned}
 \langle p, \text{ev}(t[p], q) \rangle \circ \langle \text{id}, \kappa \rangle &= \langle p \circ \langle \text{id}, \kappa \rangle, \text{ev}(t[p], q)[\langle \text{id}, \kappa \rangle] \rangle \\
 &= \langle \text{id}, \text{ev}(t[p][\langle \text{id}, \kappa \rangle], q[\langle \text{id}, \kappa \rangle]) \rangle \\
 &= \langle \text{id}, \text{ev}(t, \kappa) \rangle
 \end{aligned}$$

Since  $\langle p, \text{ev}(t[p], q) \rangle = u \circ p$  this implies

$$\langle \text{id}, \text{ev}(t, \kappa) \rangle = u \circ p \circ \langle \text{id}, \kappa \rangle = u$$

Likewise we can prove that  $\langle \text{id}, \text{ev}(t, \kappa') \rangle = u$  and so  $\text{ev}(t, \kappa) = \text{ev}(t, \kappa')$ .  $\square$

## 5. Identity types

Since  $\text{Set}^{\mathbb{T}}$  is a presheaf category it models *extensional* identity types, i.e., identity types with the identity reflection axiom. Recall that the rules for these are

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash u : A}{\Gamma \vdash \text{Id}_A(t, u) \text{ type}} \quad \frac{\Gamma \vdash t : A}{\Gamma \vdash \text{refl}_A(t) : \text{Id}_A(t, t)} \quad \frac{\Gamma \vdash p : \text{Id}_A(t, u)}{\Gamma \vdash t = u}$$

In the CwF structure, this structure is defined, for  $t, u \in \text{Set}^{\mathbb{T}}(\Gamma \vdash A)$  as

$$\text{Id}_A(t, u)_{(\mathcal{E}, \delta)}(\gamma) = \{\star \mid t(\gamma) = u(\gamma)\}$$

**Lemma 5.1.** Let  $A \in \text{Set}^{\mathbb{T}}(\Gamma)$  and  $t, u \in \text{Set}^{\mathbb{T}}(\Gamma \vdash A)$ . If  $A$  is invariant under clock introduction, so is  $\text{Id}_A(t, u)$ .

*Proof.* We must show that if  $\iota : (\mathcal{E}, \delta) \rightarrow ((\mathcal{E}, \lambda), \delta[\lambda \mapsto n])$  is given by the inclusion, then

$$\iota \cdot (-) : \{\star \mid t(\gamma) = u(\gamma)\} \rightarrow \{\star \mid t(\iota \cdot \gamma) = u(\iota \cdot \gamma)\}$$

is an isomorphism. First recall that since  $t$  and  $u$  are elements,  $t(\iota \cdot \gamma) = \iota \cdot t(\gamma)$  and likewise for  $u$ . Since  $A$  is invariant under clock introduction,  $\iota \cdot (-)$  is an isomorphism on  $A$ , and so  $t(\gamma) = u(\gamma)$  if and only if  $\iota \cdot t(\gamma) = \iota \cdot u(\gamma)$ . This implies that  $\iota \cdot (-)$  on  $\text{Id}_A(t, u)$  is also an isomorphism as required.  $\square$

## 6. Delayed substitutions

In the simply typed setting the applicative functor (McBride & Paterson 2008) structure of the later modality is essential. For instance, it allows us to apply a term  $f$  of type  $\blacktriangleright^{\kappa}(A \rightarrow B)$  to a term  $t$  of type  $\blacktriangleright^{\kappa}A$  to get a term  $f \circledast^{\kappa} t$  of type  $\blacktriangleright^{\kappa}B$ ; that is, if we have a function after one

$\kappa$ -step and if after one  $\kappa$ -step we have an argument, we can apply the function at the time, and get the result after one  $\kappa$ -step.

In GDTT the function types can be dependent, and thus to be able to use the later modality to its fullest, the applicative functor structure needs to be generalised, so that we can apply a term  $f$  of type  $\blacktriangleright^{\kappa}(\prod(x : A).B)$  to a term  $t$  of type  $\blacktriangleright^{\kappa}A$ . In GDTT the type of the delayed application  $f \otimes^{\kappa} t$  becomes  $\blacktriangleright^{\kappa}[x \leftarrow t].B$  where  $[x \leftarrow t]$  is a *delayed substitution*, and  $x$  is bound in  $\blacktriangleright^{\kappa}[x \leftarrow t].B$ . If at some point we learn that  $t$  is of the form  $\text{next}^{\kappa} t'$  for some  $t'$  we can actually perform the substitution and get the type  $\blacktriangleright^{\kappa}B[t'/x]$ . This process can be iterated, e.g., if  $B$  is also a dependent product  $\prod(y : C).D$  and  $s$  is a term of type  $\blacktriangleright^{\kappa}[x \leftarrow t].C$  then the delayed application  $f \otimes^{\kappa} t \otimes^{\kappa} s$  is well-typed with type  $\blacktriangleright^{\kappa}[x \leftarrow t, y \leftarrow s].D$ .

Delayed substitutions satisfy convenient judgemental equalities (listed in Figure 3) which ensure that delayed substitutions can be manipulated in an intuitive way. For example, if the type  $A$  is well-formed without  $x$  then the delayed substitution in  $\blacktriangleright^{\kappa}[x \leftarrow t]A$  is redundant, and thus  $\blacktriangleright^{\kappa}[x \leftarrow t]A = \blacktriangleright^{\kappa}A$ . Further, as explained above, if the term  $t$  is of type  $\text{next}^{\kappa} t'$  then we can perform an actual substitution, and thus  $\blacktriangleright^{\kappa}[x \leftarrow \text{next}^{\kappa} t'].B = \blacktriangleright^{\kappa}B[t'/x]$ . Finally, the order of bindings in  $\blacktriangleright^{\kappa}[x \leftarrow t, y \leftarrow s].D$  matters only in as much as it usually does in dependent type theory. That is,  $x \leftarrow t$  and  $y \leftarrow s$  can be exchanged provided  $x$  does not appear in the type of  $y$ .

To conclude this introduction to delayed substitutions we remark that they can be attached to the term former  $\text{next}^{\kappa} t$  as well and they enjoy analogous rules. As shown in previous work (Bizjak et al. 2016) a calculus with just these generalised  $\text{next}^{\kappa}$  and  $\blacktriangleright^{\kappa}$  can express the delayed application construct which was primitive in simply typed calculi with guarded recursion. We refer to (Bizjak et al. 2016) for extensive examples of how to use delayed substitutions for reasoning about guarded recursive and coinductive terms.

The typing rules for delayed substitutions and related constructs are recalled in Figure 2 and the equality rules are recalled in Figure 3. We write  $\xi : \Gamma \rightarrow^{\kappa} \Gamma'$  for the delayed substitution  $\xi$  from  $\Gamma$  to  $\Gamma'$ . Note that  $\Gamma'$  is not a context, but a telescope such that  $\Gamma, \Gamma'$  is a well-formed context. The delayed substitution  $\xi$  is a list of pairs written as  $x \leftarrow t$ , which are successively well-typed in context  $\Gamma$  of types derived from  $\Gamma'$ , as stated in the formation rule in Figure 2.

### Delayed substitutions

$$\frac{\Gamma \vdash \quad \Gamma \vdash \kappa : \text{clock}}{\cdot : \Gamma \rightarrow^{\kappa} \cdot} \quad \frac{\xi : \Gamma \rightarrow^{\kappa} \Gamma' \quad \Gamma, \Gamma' \vdash A \text{ type} \quad \Gamma \vdash t : \blacktriangleright^{\kappa} \xi.A}{\xi[x \leftarrow t] : \Gamma \rightarrow^{\kappa} \Gamma', x : A}$$

### Well-formed types

$$\frac{\Gamma, \Gamma' \vdash A \text{ type} \quad \xi : \Gamma \rightarrow^{\kappa} \Gamma'}{\Gamma \vdash \blacktriangleright^{\kappa} \xi.A \text{ type}}$$

### Well-typed terms

$$\frac{\Gamma, \Gamma' \vdash t : A \quad \xi : \Gamma \rightarrow^{\kappa} \Gamma'}{\Gamma \vdash \text{next}^{\kappa} \xi.t : \blacktriangleright^{\kappa} \xi.A}$$

Figure 2. Typing rules involving delayed substitutions.

**Type equality**

$$\begin{array}{c}
\frac{\xi[x \leftarrow t] : \Gamma \rightarrow^{\kappa} \Gamma', x : B \quad \Gamma, \Gamma' \vdash A \text{ type}}{\Gamma \vdash \blacktriangleright^{\kappa} \xi[x \leftarrow t].A = \blacktriangleright^{\kappa} \xi.A} \\
\\
\frac{\xi[x \leftarrow t, y \leftarrow u] \xi' : \Gamma \rightarrow^{\kappa} \Gamma', x : B, y : C, \Gamma'' \quad \Gamma, \Gamma' \vdash C \text{ type} \quad \Gamma, \Gamma', x : B, y : C, \Gamma'' \vdash A \text{ type}}{\Gamma \vdash \blacktriangleright^{\kappa} \xi[x \leftarrow t, y \leftarrow u] \xi'.A = \blacktriangleright^{\kappa} \xi[y \leftarrow u, x \leftarrow t] \xi'.A} \\
\\
\frac{\xi : \Gamma \rightarrow^{\kappa} \Gamma' \quad \Gamma, \Gamma', x : B \vdash A \text{ type} \quad \Gamma, \Gamma' \vdash t : B}{\Gamma \vdash \blacktriangleright^{\kappa} \xi[x \leftarrow \text{next}^{\kappa} \xi.t].A = \blacktriangleright^{\kappa} \xi.A[t/x]} \\
\\
\frac{\Gamma, \Gamma', \Gamma'' \vdash A \text{ type} \quad \xi : \Gamma \rightarrow^{\kappa} \Gamma' \quad \xi' : \Gamma \rightarrow^{\kappa} \Gamma''}{\Gamma \vdash \blacktriangleright^{\kappa} \xi. \blacktriangleright^{\kappa} \xi'.A = \blacktriangleright^{\kappa} \xi'. \blacktriangleright^{\kappa} \xi.A} \\
\\
\frac{\xi : \Gamma \rightarrow^{\kappa} \Gamma' \quad \Gamma, \Gamma' \vdash t : A \quad \Gamma, \Gamma' \vdash s : A}{\Gamma \vdash \text{ld}_{\blacktriangleright^{\kappa} \xi.A}(\text{next}^{\kappa} \xi.t, \text{next}^{\kappa} \xi.s) = \blacktriangleright^{\kappa} \xi.\text{ld}_A(t, s)}
\end{array}$$

**Term equality**

$$\begin{array}{c}
\frac{\xi[x \leftarrow t] : \Gamma \rightarrow^{\kappa} \Gamma', x : B \quad \Gamma, \Gamma' \vdash u : A}{\Gamma \vdash \text{next}^{\kappa} \xi[x \leftarrow t].u = \text{next}^{\kappa} \xi.u : \blacktriangleright^{\kappa} \xi.A} \\
\\
\frac{\xi[x \leftarrow t, y \leftarrow u] \xi' : \Gamma \rightarrow^{\kappa} \Gamma', x : B, y : C, \Gamma'' \quad \Gamma, \Gamma' \vdash C \text{ type} \quad \Gamma, \Gamma', x : B, y : C, \Gamma'' \vdash v : A}{\Gamma \vdash \text{next}^{\kappa} \xi[x \leftarrow t, y \leftarrow u] \xi'.v = \text{next}^{\kappa} \xi[y \leftarrow u, x \leftarrow t] \xi'.v : \blacktriangleright^{\kappa} \xi[x \leftarrow t, y \leftarrow u] \xi'.A} \\
\\
\frac{\xi : \Gamma \rightarrow^{\kappa} \Gamma' \quad \Gamma, \Gamma', x : B \vdash u : A \quad \Gamma, \Gamma' \vdash t : B}{\Gamma \vdash \text{next}^{\kappa} \xi[x \leftarrow \text{next}^{\kappa} \xi.t].u = \text{next}^{\kappa} \xi.u[t/x] : \blacktriangleright^{\kappa} \xi.A[t/x]} \quad \frac{\Gamma \vdash t : \blacktriangleright^{\kappa} \xi.A}{\Gamma \vdash \text{next}^{\kappa} \xi[x \leftarrow t].x = t : \blacktriangleright^{\kappa} \xi.A} \\
\\
\frac{\Gamma, \Gamma', \Gamma'' \vdash A \text{ type} \quad \Gamma, \Gamma', \Gamma'' \vdash u : A \quad \xi : \Gamma \rightarrow^{\kappa} \Gamma' \quad \xi' : \Gamma \rightarrow^{\kappa} \Gamma''}{\Gamma \vdash \text{next}^{\kappa} \xi. \text{next}^{\kappa} \xi'.u = \text{next}^{\kappa} \xi'. \text{next}^{\kappa} \xi.u : \blacktriangleright^{\kappa} \xi. \blacktriangleright^{\kappa} \xi'.A}
\end{array}$$

Figure 3. Equality rules involving delayed substitutions.

The typing rule for `prev` is generalised in (Bizjak et al. 2016) to allow elimination also of  $\blacktriangleright$  with attached delayed substitutions. We now recall that rule and show that it is admissible.

**Proposition 6.1.** For any delayed substitution  $\xi : \Gamma, \kappa \rightarrow^{\kappa} \Gamma'$ , there is a substitution

$$\text{adv}^{\kappa} \xi : \Gamma, \kappa : \text{clock} \rightarrow \Gamma, \kappa : \text{clock}, \Gamma'$$

defined as

$$\begin{aligned}
\text{adv}^{\kappa}(\cdot) &= \text{id}_{\Gamma, \kappa} \\
\text{adv}^{\kappa}(\xi'[x \mapsto s]) &= (\text{adv}^{\kappa} \xi')[x \mapsto (\text{prev } \kappa.s)[\kappa]]
\end{aligned}$$

such that whenever  $\Gamma, \kappa, \Gamma' \vdash A \text{ type}$  and  $\Gamma, \kappa : \text{clock} \vdash t : \blacktriangleright^{\kappa} \xi.A$  also  $\Gamma \vdash \text{prev } \kappa.t : \forall \kappa. A(\text{adv}^{\kappa} \xi)$ .

Moreover, the following equality rule holds

$$\frac{\Gamma, \kappa, \Gamma' \vdash u : A \quad \xi : (\Gamma, \kappa : \text{clock}) \rightarrow^\kappa \Gamma'}{\Gamma \vdash \text{prev } \kappa. \text{next}^\kappa \xi. u = \Lambda \kappa. u(\text{adv}^\kappa(\xi)) : \forall \kappa. A(\text{adv}^\kappa(\xi))}$$

Note that in the definition of  $\text{adv}^\kappa(\xi'[x \mapsto s])$ , the typing assumption on  $s$  is

$$\Gamma \vdash s : \blacktriangleright^\kappa \xi'. A$$

and so the typing of  $\text{adv}$  relies on the second statement of the proposition. Thus the statements of welltypedness of  $\text{adv}^\kappa(\xi)$  and of  $\text{prev } \kappa. t$  must be proved by simultaneous induction over the length of  $\xi$ .

*Proof.* We first define the concept of applying  $\text{next}^\kappa$  to a substitution obtaining a delayed substitution. This should be thought of as an inverse operation to advancing a delayed substitution. Let  $\sigma : \Gamma \rightarrow \Gamma, \Gamma'$  be a substitution which restricted to the context  $\Gamma$  is the identity. Define

$$\text{next}^\kappa(\sigma) : \Gamma \rightarrow^\kappa \Gamma'$$

by induction on the size of  $\Gamma'$  by

$$\text{next}^\kappa(\sigma[x \mapsto u]) = \text{next}^\kappa(\sigma)[x \leftarrow \text{next}^\kappa(u)]$$

This is welltyped, since by assumption  $\Gamma \vdash u : A\sigma$  and so  $\Gamma \vdash \text{next}^\kappa(u) : \blacktriangleright^\kappa(A\sigma)$  and

$$\blacktriangleright^\kappa(A\sigma) = \blacktriangleright^\kappa \text{next}^\kappa(\sigma). A.$$

Since  $\text{next}^\kappa(\text{adv}^\kappa(\xi)) = \xi$  by the  $\eta$  rule for  $\text{prev}$  the assumed type of  $t$  in the statement of the proposition is

$$\begin{aligned} \blacktriangleright^\kappa \xi. A &= \blacktriangleright^\kappa \text{next}^\kappa(\text{adv}^\kappa(\xi)). A \\ &= \blacktriangleright^\kappa A(\text{adv}^\kappa(\xi)) \end{aligned}$$

by repeated application of the first and third rule of Figure 3. Thus,  $\Gamma \vdash \text{prev } \kappa. t : \forall \kappa. A(\text{adv}^\kappa \xi)$  as desired. The equality rule stated at the end of the proposition follows analogously as

$$\begin{aligned} \text{prev } \kappa. \text{next}^\kappa \xi. u &= \text{prev } \kappa. \text{next}^\kappa [\text{next}^\kappa(\text{adv}^\kappa(\xi))] . u \\ &= \text{prev } \kappa. \text{next}^\kappa (u(\text{adv}^\kappa(\xi))) \\ &= \Lambda \kappa. u(\text{adv}^\kappa(\xi)) \end{aligned}$$

where the last equality is the  $\beta$  rule for  $\text{prev}$  from Figure 1.  $\square$

### 6.1. Semantics of delayed substitutions.

Let  $\Gamma$  be an object of  $\text{Set}^\mathbb{T}$ . A telescope over  $\Gamma$  is a sequence of families  $(A_1, \dots, A_n)$ , such that  $A_{i+1} \in \text{Set}^\mathbb{T}(\Gamma. A_1 \dots A_i)$  for each  $i$ . Let  $\kappa \in \text{Set}^\mathbb{T}(\Gamma \vdash \text{clock}[\Gamma])$ . We define the sets of *delayed sequence of elements*  $\text{Set}^\mathbb{T}(\Gamma \rightarrow^\kappa \Gamma')$  to be the set of mappings  $\xi$ , associating to each  $(\mathcal{E}, \delta)$  and  $\gamma \in \Gamma_{(\mathcal{E}, \delta)}$  such that  $\delta(\kappa(\gamma)) > 0$  a sequence  $(\xi_1, \dots, \xi_n)$  such that

$$\xi_{i+1}(\gamma) \in A_{i+1}(\text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi_1(\gamma), \dots, \xi_i(\gamma))$$

and such that for every  $\sigma : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')$  such that  $\delta'(\kappa(\sigma \cdot \gamma)) > 0$

$$\sigma^{-\kappa(\gamma)} \cdot (\xi_{i+1}(\gamma)) = \xi_{i+1}(\sigma \cdot \gamma)$$

Given a telescope  $(A_1, \dots, A_{n+1})$  over  $\Gamma$  and  $\xi \in \text{Set}^{\mathbb{T}}(\Gamma \rightarrow^{\kappa} (A_1, \dots, A_n))$ , define the family  $\blacktriangleright^{\kappa} \xi.A_{n+1}$  over  $\Gamma$  as

$$(\blacktriangleright^{\kappa} \xi.A_{n+1})(\gamma) = \begin{cases} \{\star\} & \text{if } \delta(\kappa(\gamma)) = 0 \\ A_{n+1}(\text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi_1(\gamma), \dots, \xi_n(\gamma)) & \text{otherwise} \end{cases}$$

with action  $\sigma \cdot (-)$  defined using  $\sigma^{-\kappa(\gamma)} \cdot (-)$  on  $A_{n+1}$ . Note that this implies that if  $(\xi_1, \dots, \xi_n) \in \text{Set}^{\mathbb{T}}(\Gamma \rightarrow^{\kappa} (A_1, \dots, A_n))$  and  $\xi_{n+1}$  is an element in  $\blacktriangleright^{\kappa} \xi.A_{n+1}$  then  $(\xi_1, \dots, \xi_{n+1})$  is in the set  $\text{Set}^{\mathbb{T}}(\Gamma \rightarrow^{\kappa} (A_1, \dots, A_{n+1}))$ .

If  $t$  is an element in  $A_{n+1}$  define  $\text{next}^{\kappa} \xi.t$  as an element of  $\blacktriangleright^{\kappa} \xi.A_{n+1}$  by

$$(\text{next}^{\kappa} \xi.t)(\gamma) = \begin{cases} \star & \text{if } \delta(\kappa(\gamma)) = 0 \\ t(\text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi_1(\gamma), \dots, \xi_n(\gamma)) & \text{otherwise} \end{cases}$$

If  $\Gamma'$  is a telescope over  $\Gamma$  and  $\rho : \Delta \rightarrow \Gamma$ , there is a telescope  $\Gamma'[\rho]$  over  $\Delta$  and if further  $\xi \in \text{Set}^{\mathbb{T}}(\Gamma \rightarrow^{\kappa} \Gamma')$  we can define the reindexing  $\xi[\rho] \in \text{Set}^{\mathbb{T}}(\Delta \rightarrow^{\kappa[\rho]} \Gamma'[\rho])$  as  $(\xi[\rho])_i(\gamma) = \xi_i(\rho(\gamma))$ . The two above constructions commute with reindexing in the sense that  $(\blacktriangleright^{\kappa} \xi.A_{n+1})[\rho] = \blacktriangleright^{\kappa[\rho]} \xi[\rho].(A_{n+1}[\langle \rho p^n, q[p^{n-1}], \dots, q \rangle])$  and likewise for  $\text{next}$ .

There are semantic correspondences to all of the syntactic equalities of Figure 3, but we only state and prove a few of these. We use notation similar to the syntax for delayed substitutions, e.g., if  $\Gamma' = (A_1, \dots, A_n)$  is a telescope over  $\Gamma$  we write  $\Gamma.\Gamma'$  for  $\Gamma.A_1 \dots A_n$ . If  $(\xi_1, \dots, \xi_n) \in \text{Set}^{\mathbb{T}}(\Gamma \rightarrow^{\kappa} (A_1, \dots, A_n))$  and  $\xi_{n+1}$  is an element in  $\blacktriangleright^{\kappa} \xi.A_{n+1}$  we write  $\xi[\xi_{n+1}]$  for  $(\xi_1, \dots, \xi_{n+1})$ .

**Theorem 6.1.** Let  $\Gamma'$  be a telescope over  $\Gamma$ ,  $\kappa$  an element of  $\text{clock}[\Gamma]$  and  $\xi \in \text{Set}^{\mathbb{T}}(\Gamma \rightarrow^{\kappa} \Gamma')$

- 1 If  $t \in \text{Set}^{\mathbb{T}}(\Gamma.\Gamma' \vdash B)$  and  $A \in \text{Set}^{\mathbb{T}}(\Gamma.\Gamma'.B)$ . Then

$$\blacktriangleright^{\kappa} \xi[\text{next}^{\kappa} \xi.t].A = \blacktriangleright^{\kappa} \xi.(A[\langle \text{id}, t \rangle])$$

- 2 If also  $\Gamma''$  is a telescope over  $\Gamma$ ,  $\xi' \in \text{Set}^{\mathbb{T}}(\Gamma \rightarrow^{\kappa} \Gamma'')$  and  $A$  is in  $\text{Set}^{\mathbb{T}}(\Gamma.\Gamma'.\Gamma''[p])$  where  $p : \Gamma.\Gamma' \rightarrow \Gamma$  then

$$\blacktriangleright^{\kappa} \xi.(\blacktriangleright^{\kappa} (\xi'[p]).A) = \blacktriangleright^{\kappa} \xi'.(\blacktriangleright^{\kappa} (\xi[p]).A[\text{swap}])$$

where  $\text{swap} : \Gamma.\Gamma'.\Gamma''[p] \rightarrow \Gamma.\Gamma''.\Gamma'[p]$  is the obvious map.

- 3 If  $A$  is in  $\text{Set}^{\mathbb{T}}(\Gamma.\Gamma')$  and  $t, u \in \text{Set}^{\mathbb{T}}(\Gamma.\Gamma' \vdash A)$ . Then

$$\text{Id}_{\blacktriangleright^{\kappa} \xi.A}(\text{next}^{\kappa} \xi.t, \text{next}^{\kappa} \xi.u) = \blacktriangleright^{\kappa} \xi.(\text{Id}_A(t, u))$$

*Proof.* Write  $\xi = (\xi_1, \dots, \xi_n)$ . For the first one in the case of  $\delta(\kappa(\gamma)) > 0$  we get

$$\begin{aligned} \blacktriangleright^{\kappa} \xi[\text{next}^{\kappa} \xi.t].A(\gamma) &= A(\text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi_1(\gamma), \dots, \xi_n(\gamma), \text{next}^{\kappa} \xi.t(\gamma)) \\ &= A(\text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi_1(\gamma), \dots, \xi_n(\gamma), t(\text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi_1(\gamma), \dots, \xi_n(\gamma))) \\ &= A[\langle \text{id}, t \rangle](\text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi_1(\gamma), \dots, \xi_n(\gamma)) \\ &= \blacktriangleright^{\kappa} \xi.(A[\langle \text{id}, t \rangle])(\gamma) \end{aligned}$$

In the second one if  $\delta(\kappa(\gamma)) < 2$  both sides are  $\{\star\}$ . Otherwise, writing  $\xi(\gamma)$  for  $(\xi_1(\gamma), \dots, \xi_n(\gamma))$



and likewise for  $\xi'$  we get

$$\begin{aligned}
\blacktriangleright^{\kappa} \xi. (\blacktriangleright^{\kappa} (\xi' [p]). A) (\gamma) &= (\blacktriangleright^{\kappa} (\xi' [p]). A) (\text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi(\gamma)) \\
&= A(\text{tick}^{\kappa(\gamma)} \cdot \text{tick}^{\kappa(\gamma)} \cdot \gamma, \text{tick}^{\kappa(\gamma)} \cdot \xi(\gamma), \xi'(\text{tick}^{\kappa(\gamma)} \cdot \gamma)) \\
&= A(\text{tick}^{\kappa(\gamma)} \cdot \text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi(\text{tick}^{\kappa(\gamma)} \cdot \gamma), \text{tick}^{\kappa(\gamma)} \cdot \xi'(\gamma)) \\
&= A[\text{swap}](\text{tick}^{\kappa(\gamma)} \cdot \text{tick}^{\kappa(\gamma)} \cdot \gamma, \text{tick}^{\kappa(\gamma)} \cdot \xi'(\gamma), \xi(\text{tick}^{\kappa(\gamma)} \cdot \gamma)) \\
&= \blacktriangleright^{\kappa} (\xi [p]). A[\text{swap}](\text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi(\gamma)) \\
&= \blacktriangleright^{\kappa} \xi'. (\blacktriangleright^{\kappa} (\xi [p]). A[\text{swap}])(\gamma)
\end{aligned}$$

In the last statement, if  $\delta(\kappa(\gamma)) = 0$  both sides are  $\{\star\}$ . Otherwise

$$\begin{aligned}
\text{Id}_{\blacktriangleright^{\kappa} \xi.A} (\text{next}^{\kappa} \xi.t, \text{next}^{\kappa} \xi.u) (\gamma) &= \{\star \mid (\text{next}^{\kappa} \xi.t)(\gamma) = (\text{next}^{\kappa} \xi.u)(\gamma)\} \\
&= \{\star \mid t(\text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi(\gamma)) = u(\text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi(\gamma))\} \\
&= \text{Id}_A(t, u)(\text{tick}^{\kappa(\gamma)} \cdot \gamma, \xi(\gamma)) \\
&= \blacktriangleright^{\kappa} \xi. (\text{Id}_A(t, u))(\gamma)
\end{aligned}$$

□

Finally we note that the collection of families invariant under clock introduction is closed under  $\blacktriangleright$ .

**Proposition 6.2.** If  $A$  is invariant under clock introduction so is  $\blacktriangleright^{\kappa} \xi.A$ .

*Proof.* The conclusion follows directly from the hypothesis since  $\iota \cdot (-)$  on  $\blacktriangleright^{\kappa} \xi.A$  is defined to be  $\iota^{-\kappa(\gamma)} \cdot (-)$  as defined on  $A$  when  $\delta(\kappa(\gamma)) > 0$  and the identity when  $\delta(\kappa(\gamma)) = 0$ . □

## 7. Universes

We now assume we are given a set theoretic universe with its induced notion of small sets. Being a presheaf category,  $\text{Set}^{\mathbb{T}}$  has a universe object  $\mathcal{U}$  and a dependent type  $\mathcal{E}l$  of elements defined as in (Hofmann & Streicher 1999), as we now recall. If  $(\mathcal{E}, \delta)$  is a time object, then the set  $\mathcal{U}(\mathcal{E}, \delta)$  is the set of small families over  $y(\mathcal{E}, \delta)$ . Concretely, an element  $X$  in  $\mathcal{U}(\mathcal{E}, \delta)$  assigns to each  $\sigma : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')$  a small set  $X_{\sigma}$  and to each  $\tau : (\mathcal{E}', \delta') \rightarrow (\mathcal{E}'', \delta'')$  a map  $\tau \cdot (-) : X_{\sigma} \rightarrow X_{\tau\sigma}$  in a functorial way. The action  $\sigma \cdot (-) : \mathcal{U}(\mathcal{E}, \delta) \rightarrow \mathcal{U}(\mathcal{E}', \delta')$  maps an  $X$  to the family  $(X_{\tau\sigma})_{\tau}$ . The family  $\mathcal{E}l$  is defined as  $\mathcal{E}l_{(\mathcal{E}, \delta)}(X) = X_{\text{id}}$  with action  $\sigma \cdot (-) : \mathcal{E}l_{(\mathcal{E}, \delta)}(X) \rightarrow \mathcal{E}l_{(\mathcal{E}', \delta')}(\sigma \cdot X)$  defined as  $\sigma \cdot (-) : X_{\text{id}} \rightarrow X_{\sigma}$ .

One might hope that this universe could be used to model an extension of Core-GD TT with one universe. However,  $\mathcal{U}$  is not a constant presheaf, and therefore not invariant under clock introduction. Another way to see this is that the map

$$\blacktriangleright : \mathcal{C} \times \mathcal{U} \rightarrow \mathcal{U}$$

defined, at  $(\mathcal{E}, \delta) \in \text{Set}^{\mathbb{T}}$ , as

$$(\blacktriangleright(\lambda, X))_{\sigma : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')} = \begin{cases} 1 & \text{if } \delta'(\sigma(\lambda)) = 0 \\ X_{\text{tick}^{\sigma(\lambda)} \circ \sigma} & \text{else} \end{cases}$$

does not factor through the second projection. One can restrict the universe  $\mathcal{U}$  to the families invariant under clock introduction, i.e., those  $X$  such that  $\iota \cdot (-) : X_\sigma \rightarrow X_{\iota\sigma}$  is an isomorphism for  $\iota$  of the relevant form, but this does not rule out the problematic map, and so does not eliminate the problem. Note that  $\blacktriangleright$  above does indeed encode the constructor  $\blacktriangleright$  since if  $A \in \text{Set}^\top(\Gamma \vdash \mathcal{U}[\Gamma])$  and  $\kappa \in \text{Set}^\top(\Gamma \vdash \text{clock}[\Gamma])$  then, if  $\delta(\kappa(\gamma)) > 0$ ,

$$\begin{aligned} \mathcal{E}l[(\blacktriangleright(\kappa, A))](\gamma) &= (\blacktriangleright(\kappa(\gamma), A(\gamma)))_{\text{id}} \\ &= (A(\gamma))_{\text{tick}^{\kappa(\gamma)}} \\ &= (A(\text{tick}^{\kappa(\gamma)} \cdot \gamma))_{\text{id}} \\ &= (\mathcal{E}l[A])(\text{tick}^{\kappa(\gamma)} \cdot \gamma) \\ &= \blacktriangleright^{\kappa}(\mathcal{E}l[A])(\gamma) \end{aligned}$$

To avoid this problem we follow the approach of GDTT and introduce, for each finite set of clock variables  $\Delta$ , a universe of types depending on the clocks in  $\Delta$ . An element in this universe is to be thought of as being constant in the dimensions outside  $\Delta$ , and the operation  $\blacktriangleright^{\kappa}$  is only defined on the universe for  $\kappa \in \Delta$ . This rules out the more general  $\blacktriangleright$  operation mentioned above. It also means that universes are now indexed over a new dimension (clock contexts). We show that the operations on types are polymorphic in this dimension.

### 7.1. Universes in GDTT

We first describe the syntax of Tarski style universes in GDTT. The basic rules are listed in Figure 4 and the rules for type operations on the universes are listed in Figure 5. The type  $\mathbf{U}_\Delta$  can be formed in a context  $\Gamma$ , whenever  $\Delta$  is a sequence of clocks in that context, but the equality rules say that the universes formed by two lists are equal if the lists contain the same elements. Inclusions between sets of clocks induce inclusions between universes and these commute with taking types of elements as well as with all type operations. This is the notion of universe polymorphism in the clock dimension referred to above. The universe  $\mathbf{U}_\Delta$  is closed under  $\blacktriangleright^{\kappa}$ , but only for  $\kappa \in \Delta$  thus avoiding the problem described above. The choice of domain type  $\blacktriangleright^{\kappa} \mathbf{U}_\Delta$  for  $\overline{\blacktriangleright}^{\kappa}$  ensures that guarded recursive types can be defined by guarded recursion. For example, if  $B : \mathbf{U}_\Delta$  and  $\kappa \in \Delta$  we can define a type of guarded recursive streams over  $B$  as

$$\overline{\text{Str}}^\kappa(B) \stackrel{\text{def}}{=} \text{fix}^\kappa A. B \overline{\blacktriangleright}^{\kappa} A : \mathbf{U}_\Delta$$

where  $\overline{\blacktriangleright}$  is encoded using  $\Sigma$ -types in the usual way. Then

$$\begin{aligned} \text{El}_\Delta(\overline{\text{Str}}^\kappa(B)) &= \text{El}_\Delta(B \overline{\blacktriangleright}^{\kappa} (\text{next}^\kappa(\overline{\text{Str}}^\kappa(B)))) \\ &= \text{El}_\Delta(B) \times \text{El}_\Delta(\overline{\blacktriangleright}^{\kappa} (\text{next}^\kappa(\overline{\text{Str}}^\kappa(B)))) \\ &= \text{El}_\Delta(B) \times \blacktriangleright^{\kappa}(\text{El}_\Delta(\overline{\text{Str}}^\kappa(B))) \end{aligned}$$

thus satisfying the expected type equality for guarded recursive streams over  $\text{El}_\Delta(B)$ . In the last equality of Figure 5, the typing assumption on  $A$  is  $\Gamma \vdash A : \blacktriangleright^{\kappa} \mathbf{U}_\Delta$  and since  $\Gamma, x : \mathbf{U}_\Delta \vdash \text{in}_{\Delta, \Delta'}(x) : \mathbf{U}_{\Delta'}$  also  $\Gamma \vdash \text{next}^\kappa[x \leftarrow A] \cdot \text{in}_{\Delta, \Delta'}(x) : \blacktriangleright^{\kappa} \mathbf{U}_{\Delta'}$

Formation and typing rules

$$\frac{\Gamma \vdash \kappa_1 : \text{clock} \quad \dots \quad \Gamma \vdash \kappa_n : \text{clock}}{\Gamma \vdash \mathbf{U}_{\kappa_1, \dots, \kappa_n} \text{ type}} \qquad \frac{\Gamma \vdash t : \mathbf{U}_\Delta}{\Gamma \vdash \text{El}_\Delta(t) \text{ type}}$$

$$\frac{\Gamma \vdash t : \mathbf{U}_\Delta \quad \Gamma \vdash \mathbf{U}_{\Delta'} \text{ type} \quad \Delta \subseteq \Delta'}{\Gamma \vdash \text{in}_{\Delta, \Delta'}(t) : \mathbf{U}_{\Delta'}}$$

Equations

$$\begin{aligned} \mathbf{U}_\Delta &= \mathbf{U}_{\Delta'} && \text{if } \Delta = \Delta' \text{ as sets} \\ \text{El}_{\Delta'}(\text{in}_{\Delta, \Delta'}(t)) &= \text{El}_\Delta(t) \\ \text{in}_{\Delta', \Delta''}(\text{in}_{\Delta, \Delta'}(t)) &= \text{in}_{\Delta, \Delta''}(t) \end{aligned}$$

Figure 4. Universes in GDTT.

Formation and typing rules

$$\frac{\Gamma \vdash A : \mathbf{U}_\Delta \quad \Gamma, x : \text{El}_\Delta(A) \vdash B : \mathbf{U}_\Delta}{\Gamma \vdash \overline{\prod}_\Delta (x : A) . B : \mathbf{U}_\Delta} \qquad \frac{\Gamma \vdash A : \mathbf{U}_\Delta \quad \Gamma, x : \text{El}_\Delta(A) \vdash B : \mathbf{U}_\Delta}{\Gamma \vdash \overline{\sum}_\Delta (x : A) . B : \mathbf{U}_\Delta}$$

$$\frac{\Gamma, \kappa : \text{clock} \vdash A : \mathbf{U}_{\Delta, \kappa} \quad \kappa \notin \Delta}{\Gamma \vdash \overline{\forall} \kappa . A : \mathbf{U}_\Delta} \qquad \frac{\kappa \in \Delta \quad \Gamma \vdash A : \blacktriangleright^\kappa \mathbf{U}_\Delta}{\Gamma \vdash \blacktriangleright^\kappa A : \mathbf{U}_\Delta}$$

Equations

$$\begin{aligned} \text{El}_\Delta(\overline{\prod}_\Delta (x : A) . B) &= \prod (x : \text{El}_\Delta A) . \text{El}_\Delta B \\ \text{El}_\Delta(\overline{\sum}_\Delta (x : A) . B) &= \sum (x : \text{El}_\Delta A) . \text{El}_\Delta B \\ \text{El}_\Delta(\overline{\forall} \kappa . A) &= \forall \kappa . \text{El}_{\Delta, \kappa}(A) \\ \text{El}_\Delta(\blacktriangleright^\kappa \text{next}^\kappa \xi . A) &= \blacktriangleright^\kappa \xi . (\text{El}_\Delta A) \\ \overline{\prod}_{\Delta'} (x : \text{in}_{\Delta, \Delta'}(A)) . \text{in}_{\Delta, \Delta'}(B) &= \text{in}_{\Delta, \Delta'}(\overline{\prod}_\Delta (x : A) . B) \\ \overline{\sum}_{\Delta'} (x : \text{in}_{\Delta, \Delta'}(A)) . \text{in}_{\Delta, \Delta'}(B) &= \text{in}_{\Delta, \Delta'}(\overline{\sum}_\Delta (x : A) . B) \\ \text{in}_{\Delta, \Delta'}(\overline{\forall} \kappa . A) &= \overline{\forall} \kappa . \text{in}_{(\Delta, \kappa), (\Delta', \kappa)}(A) \\ \text{in}_{\Delta, \Delta'}(\blacktriangleright^\kappa A) &= \blacktriangleright^\kappa (\text{next}^\kappa [x \leftarrow A] . \text{in}_{\Delta, \Delta'}(x)) \end{aligned}$$

Figure 5. Syntax for codes for basic operations on types

## 7.2. A family of semantic universes

To model the universe  $\mathbf{U}_\Delta$ , we must be in a context where  $\Delta$  is defined, and the smallest syntactic context where this happens is the one with  $|\Delta|$ -many clock variables. This is modelled by the  $\mathbf{Set}^\mathbb{T}$  object  $\mathcal{C}^\Delta$  defined as  $\mathcal{C}^\Delta(\mathcal{E}, \delta) = \mathcal{E}^\Delta$ . Note here that we treat  $\Delta$  as a set, and so the exponential  $\mathcal{E}^\Delta$  is the ordinary set-theoretic one. The universe  $\mathbf{U}_\Delta$  will be modelled as a family  $\mathcal{U}^\Delta$  over  $\mathcal{C}^\Delta$ . Recall that such a type corresponds to a (covariant) presheaf over the category of elements of  $\mathcal{C}^\Delta$ , i.e., the category whose elements are triples  $(\mathcal{E}, \delta, f)$ , such that the first two components constitute an object in  $\mathbb{T}$  and the last is a map  $f : \Delta \rightarrow \mathcal{E}$ . A morphism  $\sigma : (\mathcal{E}, \delta, f) \rightarrow (\mathcal{E}', \delta', g)$  is a morphism  $\sigma : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')$  in  $\mathbb{T}$  such that  $g = \sigma f$ . We will write  $\mathbf{GR}(\Delta)$  for the category of covariant presheaves over this category, and use the same notation  $(\mathbf{GR}(\Delta))$  for the CwF-structure defined similarly to the CwF structure on  $\mathbf{Set}^\mathbb{T}$ . The semantic universe  $\mathcal{U}^\Delta$  will be an object in  $\mathbf{GR}(\Delta)$  and the type  $\mathbf{El}_\Delta$  will be modelled as a family  $\mathcal{E}l^\Delta$  over  $\mathcal{U}^\Delta$ .

To avoid the problem described above with the standard universe in  $\mathbf{Set}^\mathbb{T}$ , the universe  $\mathcal{U}^\Delta$  should restrict access at level  $(\mathcal{E}, \delta, f)$  to the clocks defined in  $\Delta$ . To do this, we define  $\mathcal{U}_{(\mathcal{E}, \delta, f)}^\Delta$  to be the set of small families in  $\mathbf{Set}^\mathbb{T}$  over  $y(f[\Delta], \delta|_{f[\Delta]})$  invariant under clock introduction. Here  $f[\Delta] \subseteq \mathcal{E}$  is the image of  $f$ , and the notion of small families should be understood as described above. In other words, an element of  $\mathcal{U}_{(\mathcal{E}, \delta, f)}^\Delta$  is a family of sets  $X_\sigma$  indexed over morphisms  $\sigma$  in  $\mathbb{T}$  with domain  $(f[\Delta], \delta|_{f[\Delta]})$  together with maps  $\tau \cdot (-) : X_\sigma \rightarrow X_{\tau\sigma}$  satisfying functoriality. The requirement of invariance under clock introduction means that if  $\iota : (\mathcal{E}', \delta') \rightarrow ((\mathcal{E}', \lambda), \delta'[\lambda \mapsto n])$  is an inclusion, then  $\iota \cdot (-)$  must be an isomorphism.

For  $\sigma : (\mathcal{E}, \delta, f) \rightarrow (\mathcal{E}', \delta', \sigma f)$  we must define  $\sigma \cdot (-) : \mathcal{U}_{(\mathcal{E}, \delta, f)}^\Delta \rightarrow \mathcal{U}_{(\mathcal{E}', \delta', \sigma f)}^\Delta$ . Denote by  $\bar{\sigma}$  the restriction and corestriction of  $\sigma$ :

$$\bar{\sigma} : (f[\Delta], \delta|_{f[\Delta]}) \rightarrow (\sigma f[\Delta], \delta'|_{\sigma f[\Delta]}).$$

Using this, we define the family  $(\sigma \cdot X)_\tau = X_{\tau\bar{\sigma}}$  for  $\tau : (\sigma f[\Delta], \delta'|_{\sigma f[\Delta]}) \rightarrow (\mathcal{E}', \delta'')$ . Note that this is well-defined, i.e., if  $X$  is invariant under clock introduction, so is  $\sigma \cdot X$ .

Since  $\mathbf{GR}(\Delta)$  is equivalent to the slice of  $\mathbf{Set}^\mathbb{T}$  over  $\mathcal{C}^\Delta$ , the notion of invariance under clock introduction extends to objects and families in  $\mathbf{GR}(\Delta)$  by requiring the same for their corresponding projection maps in  $\mathbf{Set}^\mathbb{T}$ . By Lemma 4.1 this can be reformulated as requiring that the maps  $\iota \cdot (-)$  induced by maps of the form  $\iota : (\mathcal{E}, \delta, f) \rightarrow ((\mathcal{E}, \lambda), \delta[\lambda \mapsto n], \iota f)$  are isomorphisms.

**Lemma 7.1.** The object  $\mathcal{U}^\Delta$  is invariant under clock introduction.

*Proof.* If  $\iota : (\mathcal{E}, \delta, f) \rightarrow ((\mathcal{E}, \lambda), \delta[\lambda \mapsto n], \iota f)$  is the inclusion then  $\iota f[\Delta] = f[\Delta]$ , and  $\bar{\iota}$  is the identity, so  $(\iota \cdot X)_\tau = X_{\tau\bar{\iota}} = X_\tau$ , i.e.,  $\iota \cdot (-)$  is the identity and therefore an isomorphism.  $\square$

If  $X$  is an element in  $\mathcal{U}_{(\mathcal{E}, \delta, f)}^\Delta$ , define

$$\mathcal{E}l_{(\mathcal{E}, \delta, f)}^\Delta(X) = X_{i : (f[\Delta], \delta|_{f[\Delta]}) \rightarrow (\mathcal{E}, \delta)}$$

where  $i$  is the inclusion. If  $\sigma : (\mathcal{E}, \delta) \rightarrow (\mathcal{E}', \delta')$  we must define

$$\sigma \cdot (-) : \mathcal{E}l_{(\mathcal{E}, \delta, f)}^\Delta(X) \rightarrow \mathcal{E}l_{(\mathcal{E}', \delta', \sigma f)}^\Delta(\sigma \cdot X)$$

The codomain of this map is

$$\mathcal{E}l_{(\mathcal{E}', \delta', \sigma f)}^\Delta(\sigma \cdot X) = (\sigma \cdot X)_j = X_{j \circ \bar{\sigma}} = X_{\sigma \circ i}$$

where  $j : \sigma[f[\Delta]] \rightarrow \mathcal{E}'$  is the inclusion. We can therefore define  $\sigma \cdot (-) : X_i \rightarrow X_{\sigma oi}$  to be the map that is part of the structure of  $X$ .

**Lemma 7.2.** The family  $\mathcal{E}l^\Delta$  over  $\mathcal{U}^\Delta$  is invariant under clock introduction.

*Proof.* Let  $\iota : (\mathcal{E}, \delta, f) \rightarrow ((\mathcal{E}, \lambda), \delta[\lambda \mapsto n], \iota f)$  be the inclusion and let  $X$  be an element in  $\mathcal{U}_{(\mathcal{E}, \delta, f)}^\Delta$ . We must show that

$$\iota \cdot (-) : \mathcal{E}l_{(\mathcal{E}, \delta, f)}^\Delta(X) \rightarrow \mathcal{E}l_{((\mathcal{E}, \lambda), \delta[\lambda \mapsto n], \iota f)}^\Delta(\iota \cdot X)$$

is an isomorphism. By definition of  $\mathcal{E}l^\Delta$  this map is  $\iota \cdot (-) : X_i \rightarrow X_{\iota oi}$ , which is part of the structure of  $X$ . Since  $X$  is an element in the universe  $\mathcal{U}^\Delta$  it must be invariant under clock introduction, which means exactly that all maps of the form  $\iota \cdot (-)$  are isomorphisms.  $\square$

We now describe an abstract construction that leads to the universe  $\mathcal{U}^\Delta$ . This construction will not be used in the remainder of the paper, and so is not of technical importance, but perhaps of conceptual interest to some readers. Consider the universe  $\mathcal{U}$  of small families invariant under clock introduction in  $\text{Set}^\mathbb{T}$ . There is a functor  $F$  from  $\text{Set}^\mathbb{T}$  to  $\text{GR}(\Delta)$  mapping an object  $\Gamma$  to the presheaf whose value at  $(\mathcal{E}, \delta, f)$  is  $\Gamma(f[\Delta], \delta|_{f[\Delta]})$ . This extends to families by mapping  $A$  over  $\Gamma$  to the family whose value at  $\gamma \in \Gamma(f[\Delta], \delta|_{f[\Delta]})$  is  $A(i \cdot \gamma)$ . Note that this is not a mapping of CwFs, since it does not preserve comprehension:

$$\begin{aligned} F(\Gamma.A)(\mathcal{E}, \delta, f) &= \{\langle \gamma, a \rangle \mid \gamma \in \Gamma(f[\Delta], \delta|_{f[\Delta]}), a \in A(\gamma)\} \\ F(\Gamma).F(A)(\mathcal{E}, \delta, f) &= \{\langle \gamma, a \rangle \mid \gamma \in \Gamma(f[\Delta], \delta|_{f[\Delta]}), a \in A(i \cdot \gamma)\} \end{aligned}$$

(Although it does up to isomorphism if attention is restricted to families invariant under clock introduction). The universe  $\mathcal{U}^\Delta$  is this mapping applied to  $\mathcal{U}$  and  $\mathcal{E}l^\Delta$  is the same mapping applied to the family of elements over  $\mathcal{U}$ .

The next key lemma gives a partial answer to the question of what the universes  $\mathcal{U}^\Delta$  classify. The answer is partial, since it only applies in contexts invariant under clock introduction. As we shall see below, this result is sufficient for constructing codes for type operations on the universes.

**Lemma 7.3.** Let  $\Gamma$  be an object in  $\text{GR}(\Delta)$  invariant under clock introduction and let  $A$  be a small family over  $\Gamma$ , also invariant under clock introduction. There is a unique code  $\ulcorner A \urcorner : \Gamma \rightarrow \mathcal{U}^\Delta$  in  $\text{GR}(\Delta)$  such that  $A = \mathcal{E}l^\Delta[\ulcorner A \urcorner]$ .

*Proof.* The assumption of invariance under clock introduction implies that for any object  $(\mathcal{E}, \delta, f)$  the map  $i \cdot (-)$  induced by  $i : (f[\Delta], \delta|_{f[\Delta]}, f) \rightarrow (\mathcal{E}, \delta, f)$  is an isomorphism on  $\Gamma$ . We will write  $i^{-1} \cdot (-)$  for the inverse map. The code  $\ulcorner A \urcorner$  is defined as

$$(\ulcorner A \urcorner_{(\mathcal{E}, \delta, f)}(\gamma))_{\tau : (f[\Delta], \delta|_{f[\Delta]}, f) \rightarrow (\mathcal{E}', \delta', \tau f)} = A_{(\mathcal{E}', \delta', \tau f)}(\tau \cdot i^{-1} \cdot \gamma)$$

We first show that this defines a map of presheaves: If  $\sigma : (\mathcal{E}, \delta, f) \rightarrow (\mathcal{E}', \delta', \sigma f)$  and  $\tau : (\sigma f[\Delta], \delta'|_{\sigma f[\Delta]}, \sigma f) \rightarrow (\mathcal{E}'', \delta'', \tau \sigma f)$  then

$$(\ulcorner A \urcorner_{(\mathcal{E}', \delta', \sigma f)}(\sigma \cdot \gamma))_\tau = A_{(\mathcal{E}'', \delta'', \tau \sigma f)}(\tau \cdot j^{-1} \cdot \sigma \cdot \gamma)$$

where  $j : (\sigma f[\Delta], \delta'|_{\sigma f[\Delta]}, \sigma f) \rightarrow (\mathcal{E}', \delta', \sigma f)$  is the inclusion. Since  $\sigma i = j \bar{\sigma}$  also  $j^{-1} \cdot \sigma \cdot \gamma = \bar{\sigma} \cdot i^{-1} \cdot \gamma$

and so

$$\begin{aligned} (\ulcorner A^\top_{(\mathcal{E}', \delta', \sigma f)}(\sigma \cdot \gamma) \urcorner)_\tau &= A_{(\mathcal{E}'', \delta'', \tau \sigma f)}(\tau \cdot \bar{\sigma} \cdot i^{-1} \cdot \gamma) \\ &= (\ulcorner A^\top_{(\mathcal{E}, \delta, f)}(\gamma) \urcorner)_{\tau \bar{\sigma}} \\ &= (\sigma \cdot (\ulcorner A^\top_{(\mathcal{E}, \delta, f)}(\gamma) \urcorner))_\tau \end{aligned}$$

so  $(\ulcorner A^\top(\sigma \cdot \gamma) \urcorner) = \sigma \cdot (\ulcorner A^\top(\gamma) \urcorner)$  meaning that  $\ulcorner A^\top$  is a map of presheaves.

This definition defines a code for  $A$  since

$$(\mathcal{E}l^\Delta[\ulcorner A^\top \urcorner])(\gamma) = \mathcal{E}l^\Delta(\ulcorner A^\top(\gamma) \urcorner) = (\ulcorner A^\top(\gamma) \urcorner)_i = A(i \cdot i^{-1} \cdot \gamma) = A(\gamma)$$

For uniqueness, suppose  $\rho : \Gamma \rightarrow \mathcal{U}^\Delta$  satisfies  $\mathcal{E}l^\Delta[\rho] = A$ . We must show that  $\rho(\gamma) = \ulcorner A^\top(\gamma) \urcorner$  for all  $\gamma$ , but consider first the case where  $\gamma \in \Gamma_{(\mathcal{E}, \delta, f)}$  for  $f$  surjective. In that case  $(\ulcorner A^\top(\gamma) \urcorner)_\tau = A(\tau \cdot \gamma)$  and

$$(\rho(\gamma))_\tau = (\rho(\gamma))_{j\bar{\tau}} = (\tau \cdot \rho(\gamma))_j = (\rho(\tau \cdot \gamma))_j = \mathcal{E}l^\Delta(\rho(\tau \cdot \gamma)) = A(\tau \cdot \gamma)$$

where  $j : \tau f[\Delta] \rightarrow \mathcal{E}'$  is the inclusion. In general (when  $f$  is not surjective) the above implies

$$\rho(\gamma) = \rho(i \cdot i^{-1} \cdot \gamma) = i \cdot \rho(i^{-1} \cdot \gamma) = i \cdot \ulcorner A^\top(i^{-1} \cdot \gamma) \urcorner = \ulcorner A^\top(i \cdot i^{-1} \cdot \gamma) \urcorner = \ulcorner A^\top(\gamma) \urcorner$$

proving uniqueness.  $\square$

### 7.3. Reindexing universes

The idea for interpreting the formation rule for the universes  $\mathbf{U}_\Delta$  in a semantic context  $\Gamma$ , is to interpret each  $\kappa \in \Delta$  as an element of  $\text{clock}[\Gamma]$ , then use this to define a map from  $\Gamma$  to  $\mathcal{C}^\Delta$  in  $\text{Set}^\mathbb{T}$ , and reindex the universe  $\mathcal{U}^\Delta$  along this map. The last of these steps uses the fact that an object of  $\text{GR}(\Delta)$  can be considered a family of  $\text{Set}^\mathbb{T}$  over  $\mathcal{C}^\Delta$ . In fact, these two notions are equivalent.

In order to prove the substitution lemma, we will generalise the above idea slightly as follows. Suppose  $\chi$  is a finite set of morphisms from  $\Gamma$  to  $\mathcal{C}$ , and suppose we are given a surjective map from some set  $\Delta$  to  $\chi$  inducing a map  $\langle \chi \rangle : \Gamma \rightarrow \mathcal{C}^\Delta$ . Define

$$\mathcal{U}^\chi \stackrel{\text{def}}{=} \mathcal{U}^\Delta[\langle \chi \rangle] \qquad \mathcal{E}l^\chi \stackrel{\text{def}}{=} \mathcal{E}l^\Delta[\langle \langle \chi \rangle \circ p, q \rangle]$$

**Proposition 7.1.** The objects  $\mathcal{U}^\chi$  and  $\mathcal{E}l^\chi$  are welldefined in the sense that they do not depend on the choice of  $\Delta$  or surjection inducing  $\langle \chi \rangle$ . Moreover, if  $\rho : \Gamma' \rightarrow \Gamma$  then

$$\mathcal{U}^{\chi \circ \rho} = \mathcal{U}^\chi[\rho] \qquad \mathcal{E}l^{\chi \circ \rho} = \mathcal{E}l^\chi[\langle \rho \circ p, q \rangle]$$

where  $\{\kappa_1, \dots, \kappa_n\}[\rho] = \{\kappa_1[\rho], \dots, \kappa_n[\rho]\}$ .

*Proof.* If  $\gamma \in \Gamma_{(\mathcal{E}, \delta)}$  the element  $\langle \chi \rangle(\gamma)$  is a map  $\Delta \rightarrow \mathcal{E}$ . By definition,  $\mathcal{U}^\chi(\gamma) = \mathcal{U}^\Delta(\langle \chi \rangle(\gamma))$  is the set of small families  $(X_\tau)_{\tau : (\langle \chi \rangle(\gamma)[\Delta], \delta)_{\langle \chi \rangle(\gamma)[\Delta]} \rightarrow (\mathcal{E}', \delta')}$ . Since the map  $\Delta \rightarrow \chi$  is assumed surjective,  $\langle \chi \rangle(\gamma)[\Delta] = \{\kappa(\gamma) \mid \kappa \in \chi\}$  and thus independent of the choice of  $\Delta$  and surjection. Since  $\mathcal{E}l^\chi(\gamma)(X) = X_i$  for  $i$  the inclusion, also  $\mathcal{E}l^\chi$  is welldefined. For the last statement, note that  $\langle \chi \rangle \circ \rho$  is the map corresponding to the composition  $\Delta \rightarrow \chi \rightarrow \chi[\rho]$ , where the last of these maps  $\kappa$  to  $\kappa[\rho]$ , and this map is surjective. Therefore,  $\mathcal{U}^{\chi \circ \rho}$  can be defined as  $\mathcal{U}^\Delta[\langle \chi \rangle \circ \rho] = \mathcal{U}^\chi[\rho]$ . The equality  $\mathcal{E}l^{\chi \circ \rho} = \mathcal{E}l^\chi[\langle \rho \circ p, q \rangle]$  follows similarly.  $\square$

The codes on universes will be defined below by constructing objects  $A_\Delta$  in  $\text{GR}(\Delta)$  and families

$B_\Delta$  over  $A_\Delta$  indexed over  $\Delta$  in such a way that whenever  $\chi$  is as above the families  $A_\chi = A_\Delta[\langle\chi\rangle]$  and  $B_\chi = B_\Delta[\langle\chi\rangle \circ p, q]$  are well defined, i.e., independent of choice of  $\Delta$  and surjection  $\Delta \rightarrow \chi$ . In this case, if each  $A_\Delta$  and  $B_\Delta$  are invariant under clock introduction, by Lemma 7.3 there is a unique code  $\ulcorner B_\Delta \urcorner : A_\Delta \rightarrow \mathcal{U}^\Delta$  in  $\text{GR}(\Delta)$  such that  $\mathcal{E}l^\Delta[\ulcorner B_\Delta \urcorner] = B_\Delta$ . In this situation we would like to define

$$\ulcorner B_\chi \urcorner = \ulcorner B_\Delta \urcorner[\langle\chi\rangle] : A_\chi \rightarrow \mathcal{U}^\chi$$

as a map in the category of presheaves over the elements of  $\Gamma$ .

**Lemma 7.4.** In the situation described above, the map  $\ulcorner B_\chi \urcorner$  is well defined, i.e., independent of the choice of  $\Delta$  and surjection  $\Delta \rightarrow \chi$ . Moreover,  $\mathcal{E}l^\chi[\ulcorner B_\chi \urcorner] = B_\chi$ , and if  $\rho : \Gamma' \rightarrow \Gamma$  then  $\ulcorner B_{\chi \circ \rho} \urcorner = \ulcorner B_\chi \urcorner[\rho]$

*Proof.* Suppose we are given two different surjections  $f : \Delta \rightarrow \chi$  and  $f' : \Delta' \rightarrow \chi$  inducing  $\langle\chi\rangle : \Gamma \rightarrow \mathcal{C}^\Delta$  and  $\langle\chi\rangle' : \Gamma \rightarrow \mathcal{C}^{\Delta'}$ . We will assume there is a surjection  $g : \Delta \rightarrow \Delta'$  such that  $f'g = f$ , otherwise apply the argument to each of the two maps in the span of projections  $\Delta \leftarrow \Delta \times \Delta' \rightarrow \Delta'$ . Note that the projections are always surjective, since  $\Delta$  is empty iff  $\chi$  is empty iff  $\Delta'$  is empty. Since  $f'g = f$  also  $\mathcal{C}^g \circ \langle\chi\rangle' = \langle\chi\rangle$ .

We first prove that  $A_\Delta[\mathcal{C}^g] = A_{\Delta'}$ . For this, observe that there is a family

$$\Phi = \{\text{ev}_\kappa : \mathcal{C}^{\Delta'} \rightarrow \mathcal{C} \mid \kappa \in \Delta'\}$$

and a surjection  $\Delta' \rightarrow \Phi$  mapping  $\kappa$  to  $\text{ev}_\kappa$ . The induced map  $\mathcal{C}^{\Delta'} \rightarrow \mathcal{C}^{\Delta'}$  is the identity. There is also a map  $\Delta \rightarrow \Phi$  mapping  $\kappa$  to  $\text{ev}_{g(\kappa)}$ . Since  $g$  is surjective, also this is surjective, and induces  $\mathcal{C}^g : \mathcal{C}^{\Delta'} \rightarrow \mathcal{C}^\Delta$ . Thus by assumption

$$A_\Delta[\mathcal{C}^g] = A_{\Delta'} \qquad B_\Delta[\langle\mathcal{C}^g \circ p, q\rangle] = B_{\Delta'}$$

In particular, these arguments apply to  $\mathcal{U}^\Delta$  and  $\mathcal{E}l^\Delta$  proving

$$\mathcal{U}^\Delta[\mathcal{C}^g] = \mathcal{U}^{\Delta'} \qquad \mathcal{E}l^\Delta[\langle\mathcal{C}^g \circ p, q\rangle] = \mathcal{E}l^{\Delta'}$$

In the second of these equations  $\mathcal{E}l^{\Delta'}$  is considered a family of  $\text{Set}^\mathbb{T}$  over  $\mathcal{C}^{\Delta'}$ .  $\mathcal{U}^{\Delta'}$ . Equivalently,  $\mathcal{E}l^{\Delta'}$  can be considered a family of  $\text{GR}(\Delta')$  over  $\mathcal{U}^{\Delta'}$ , and  $(-)[\mathcal{C}^g]$  a morphism of CwFs from  $\text{GR}(\Delta)$  to  $\text{GR}(\Delta')$ . From this latter point of view the second equation above is  $\mathcal{E}l^\Delta[\mathcal{C}^g] = \mathcal{E}l^{\Delta'}$ , and so

$$\mathcal{E}l^{\Delta'}[\ulcorner B_\Delta \urcorner[\mathcal{C}^g]] = \mathcal{E}l^\Delta[\mathcal{C}^g][\ulcorner B_\Delta \urcorner[\mathcal{C}^g]] = (\mathcal{E}l^\Delta[\ulcorner B_\Delta \urcorner])[\mathcal{C}^g] = B_\Delta[\mathcal{C}^g] = B_{\Delta'}$$

Thus, by the uniqueness statement of Lemma 7.3  $\ulcorner B_\Delta \urcorner[\mathcal{C}^g] = \ulcorner B_{\Delta'} \urcorner$ . So, finally

$$\ulcorner B_\Delta \urcorner[\langle\chi\rangle] = \ulcorner B_\Delta \urcorner[\mathcal{C}^g \circ \langle\chi\rangle'] = \ulcorner B_{\Delta'} \urcorner[\langle\chi\rangle']$$

proving welldefinedness of  $\ulcorner B_\chi \urcorner$ . The equality  $\mathcal{E}l^\chi[\ulcorner B_\chi \urcorner] = B_\chi$  follows from the fact that  $(-)[\langle\chi\rangle]$  induces a morphism of CwFs:

$$\mathcal{E}l^\chi[\ulcorner B_\chi \urcorner] = \mathcal{E}l^\Delta[\ulcorner B_\Delta \urcorner[\langle\chi\rangle]] = B_\Delta[\langle\chi\rangle] = B_\chi$$

The last statement follows as in the proof of Proposition 7.1.  $\square$

#### 7.4. Inclusions of universes

We now show how to model inclusions of universes and the codes for type operations on universes described in Figure 5.

**Proposition 7.2.** Suppose  $\chi \subseteq \chi'$  are sets of elements of  $\text{clock}[\Gamma]$ , and  $t \in \text{Set}^{\mathbb{T}}(\Gamma \vdash \mathcal{U}^{\chi})$ . There is an element  $\text{in}_{\chi, \chi'}(t) \in \text{Set}^{\mathbb{T}}(\Gamma \vdash \mathcal{U}^{\chi'})$  such that  $\mathcal{E}l^{\chi'}[\langle \text{id}_{\Gamma}, \text{in}_{\chi, \chi'}(t) \rangle] = \mathcal{E}l^{\chi}(t)$ . If further  $\chi' \subseteq \chi''$  then  $\text{in}_{\chi', \chi''}(\text{in}_{\chi, \chi'}(t)) = \text{in}_{\chi, \chi''}(t)$ . Moreover, this construction commutes with reindexing in the sense that if  $\rho : \Gamma' \rightarrow \Gamma$  then  $(\text{in}_{\chi, \chi'}(t))[\rho] = \text{in}_{\chi[\rho], \chi'[\rho]}(t[\rho])$ .

*Proof.* Let  $\langle \chi' \rangle : \Gamma \rightarrow \mathcal{C}^{\Delta'}$  be induced by a given surjection  $\Delta' \rightarrow \chi'$ . Let  $\Delta \subseteq \Delta'$  be the subset mapped to  $\chi$ , and let  $\langle \chi \rangle : \Gamma \rightarrow \mathcal{C}^{\Delta}$  be the map corresponding to the projection. There is a projection  $\pi_{\Delta, \Delta'} : \mathcal{C}^{\Delta'} \rightarrow \mathcal{C}^{\Delta}$  and so  $\mathcal{U}^{\Delta}[\pi_{\Delta, \Delta'}]$  is an object of  $\text{GR}(\Delta')$ . Moreover

$$\mathcal{U}^{\Delta}[\pi_{\Delta, \Delta'}][\langle \chi' \rangle] = \mathcal{U}^{\chi} \quad (9)$$

simply because  $\pi_{\Delta, \Delta'} \circ \langle \chi' \rangle = \langle \chi \rangle$ .

Since  $\mathcal{U}^{\Delta}$  and  $\mathcal{E}l^{\Delta}$  are invariant under clock introduction, so are  $\mathcal{U}^{\Delta}[\pi_{\Delta, \Delta'}]$  and the family  $\mathcal{E}l^{\Delta}[\langle \pi_{\Delta, \Delta'} \circ p, q \rangle]$ . The latter is a family over  $\mathcal{C}^{\Delta'}$ .  $\mathcal{U}^{\Delta}[\pi_{\Delta, \Delta'}]$  in  $\text{Set}^{\mathbb{T}}$ , but can be likewise considered a family over  $\mathcal{U}^{\Delta}[\pi_{\Delta, \Delta'}]$  in  $\text{GR}(\Delta')$ . By Lemma 7.3 there is a unique map  $\text{in}_{\Delta, \Delta'}$  in  $\text{GR}(\Delta')$  such that  $\mathcal{E}l^{\Delta'}[\text{in}_{\Delta, \Delta'}] = \mathcal{E}l^{\Delta}[\langle \pi_{\Delta, \Delta'} \circ p, q \rangle]$ . By (9) then

$$\text{in}_{\Delta, \Delta'}[\langle \chi' \rangle] : \mathcal{U}^{\chi} \rightarrow \mathcal{U}^{\chi'}$$

is a map between presheaves over the category of elements of  $\Gamma$ , and the above implies

$$\mathcal{E}l^{\chi'}[\text{in}_{\Delta, \Delta'}[\langle \chi' \rangle]] = \mathcal{E}l^{\chi}$$

We now define

$$\text{in}_{\chi, \chi'}(t) \stackrel{\text{def}}{=} \text{in}_{\Delta, \Delta'}[\langle \chi' \rangle](t)$$

Then

$$\begin{aligned} \mathcal{E}l^{\chi'}[\langle \text{id}_{\Gamma}, \text{in}_{\chi, \chi'}(t) \rangle] &= \mathcal{E}l^{\chi'}[\langle \text{id}_{\Gamma}, \text{in}_{\Delta, \Delta'}[\langle \chi' \rangle](t) \rangle] \\ &= \mathcal{E}l^{\chi}[\langle \text{id}_{\Gamma}, t \rangle] \end{aligned}$$

The statement on composition of these inclusions follow from the uniqueness statement of Lemma 7.3. The element  $\text{in}_{\chi, \chi'}(t)$  as defined above can be proved independent of the choice of  $\Delta'$  using a slight generalisation of Lemma 7.4, but we omit the argument here. Similar arguments can also show that it commutes with reindexing.  $\square$

#### 7.5. Codes for basic type constructors

The codes for  $\Pi$  and  $\Sigma$ -types are modelled as morphisms with domain

$$\mathcal{U}_1^{\chi} = \Sigma(\mathcal{U}^{\chi}, \mathcal{E}l^{\chi} \rightarrow \mathcal{U}^{\chi}[p])$$

The family  $\mathcal{U}_1^{\chi}$  classifies  $\mathcal{U}^{\chi}$ -small families over  $\mathcal{U}^{\chi}$ -small objects in a sense that we now explain.

First note that there is a family

$$\mathcal{E}l_1^{\chi} \stackrel{\text{def}}{=} \mathcal{E}l^{\chi}[\langle p, \pi_1(q) \rangle] \in \text{Set}^{\mathbb{T}}(\Gamma, \mathcal{U}_1^{\chi})$$

and an element

$$\text{ev}(\pi_2(q)[p], q) \in \text{Set}^{\mathbb{T}}(\Gamma, \mathcal{U}_1^{\chi} \cdot \mathcal{E}l_1^{\chi} \vdash \mathcal{U}^{\chi}[pp]),$$



where  $\pi_1, \pi_2$  are the projections out of the  $\Sigma$ -type. So

$$\mathcal{E}l_2^X \stackrel{\text{def}}{=} \mathcal{E}l^X[\langle \text{pp}, \text{ev}(\pi_2(q)[p], q) \rangle] \in \text{Set}^\mathbb{T}(\Gamma. \mathcal{U}_1^X. \mathcal{E}l_1^X)$$

Suppose now  $A \in \text{Set}^\mathbb{T}(\Gamma)$  and  $B \in \text{Set}^\mathbb{T}(\Gamma.A)$  are  $\mathcal{U}^X$ -small in the sense that there are  $\ulcorner A^\urcorner$  and  $\ulcorner B^\urcorner$  satisfying

$$\begin{aligned} \ulcorner A^\urcorner &\in \text{Set}^\mathbb{T}(\Gamma \vdash \mathcal{U}^X) & \mathcal{E}l^X[\langle \text{id}_\Gamma, \ulcorner A^\urcorner \rangle] &= A \\ \ulcorner B^\urcorner &\in \text{Set}^\mathbb{T}(\Gamma.A \vdash \mathcal{U}^X[p]) & \mathcal{E}l^X[\langle p, \ulcorner B^\urcorner \rangle] &= B \end{aligned}$$

Now,  $\lambda(\ulcorner B^\urcorner) \in \text{Set}^\mathbb{T}(\Gamma \vdash A \rightarrow \mathcal{U}^X)$  and so

$$\langle \ulcorner A^\urcorner, \lambda(\ulcorner B^\urcorner) \rangle \in \text{Set}^\mathbb{T}(\Gamma \vdash \mathcal{U}_1^X)$$

Then

$$\begin{aligned} \mathcal{E}l_1^X[\langle \text{id}_\Gamma, \langle \ulcorner A^\urcorner, \lambda(\ulcorner B^\urcorner) \rangle \rangle] &= \mathcal{E}l^X[\langle p, \pi_1(q) \rangle][\langle \text{id}_\Gamma, \langle \ulcorner A^\urcorner, \lambda(\ulcorner B^\urcorner) \rangle \rangle] \\ &= \mathcal{E}l^X[\langle \text{id}_\Gamma, \ulcorner A^\urcorner \rangle] \\ &= A \end{aligned}$$

and

$$\begin{aligned} \mathcal{E}l_2^X[\langle \langle p, \langle \ulcorner A^\urcorner, \lambda(\ulcorner B^\urcorner) \rangle \rangle p, q \rangle] &= \mathcal{E}l^X[\langle \text{pp}, \text{ev}(\pi_2(q)[p], q) \rangle][\langle \langle p, \langle \ulcorner A^\urcorner, \lambda(\ulcorner B^\urcorner) \rangle \rangle p, q \rangle] \\ &= \mathcal{E}l^X[\langle p, \text{ev}(\lambda(\ulcorner B^\urcorner)[p], q) \rangle] \\ &= \mathcal{E}l^X[\langle p, \ulcorner B^\urcorner \rangle] \\ &= B \end{aligned}$$

**Proposition 7.3.** Suppose  $A, \ulcorner A^\urcorner, B$  and  $\ulcorner B^\urcorner$  are as above. There are elements

$$\ulcorner \Pi^{\times\urcorner}(\ulcorner A^\urcorner, \ulcorner B^\urcorner) \in \text{Set}^\mathbb{T}(\Gamma \vdash \mathcal{U}^X) \quad \ulcorner \Sigma^{\times\urcorner}(\ulcorner A^\urcorner, \ulcorner B^\urcorner) \in \text{Set}^\mathbb{T}(\Gamma \vdash \mathcal{U}^X)$$

such that

$$\mathcal{E}l^X[\langle \text{id}_\Gamma, \ulcorner \Pi^{\times\urcorner}(\ulcorner A^\urcorner, \ulcorner B^\urcorner) \rangle] = \Pi(A, B) \quad \mathcal{E}l^X[\langle \text{id}_\Gamma, \ulcorner \Sigma^{\times\urcorner}(\ulcorner A^\urcorner, \ulcorner B^\urcorner) \rangle] = \Sigma(A, B)$$

Moreover, if  $\rho : \Gamma' \rightarrow \Gamma$  then

$$\begin{aligned} (\ulcorner \Pi^{\times\urcorner}(\ulcorner A^\urcorner, \ulcorner B^\urcorner) \rangle)[\rho] &= \ulcorner \Pi^{\times[\rho]\urcorner}(\ulcorner A^\urcorner[\rho], \ulcorner B^\urcorner[\rho]) \\ (\ulcorner \Sigma^{\times\urcorner}(\ulcorner A^\urcorner, \ulcorner B^\urcorner) \rangle)[\rho] &= \ulcorner \Sigma^{\times[\rho]\urcorner}(\ulcorner A^\urcorner[\rho], \ulcorner B^\urcorner[\rho]) \\ \text{in}_{\chi, \chi'}(\ulcorner \Pi^{\times\urcorner}(\ulcorner A^\urcorner, \ulcorner B^\urcorner) \rangle) &= \ulcorner \Pi^{\times'\urcorner}(\text{in}_{\chi, \chi'}(\ulcorner A^\urcorner), \text{in}_{\chi, \chi'}(\ulcorner B^\urcorner)) \\ \text{in}_{\chi, \chi'}(\ulcorner \Sigma^{\times\urcorner}(\ulcorner A^\urcorner, \ulcorner B^\urcorner) \rangle) &= \ulcorner \Sigma^{\times'\urcorner}(\text{in}_{\chi, \chi'}(\ulcorner A^\urcorner), \text{in}_{\chi, \chi'}(\ulcorner B^\urcorner)) \end{aligned}$$

*Proof.* First note that since  $\mathcal{U}^\Delta$  and  $\mathcal{E}l^\Delta$  are invariant under clock introduction, by the closure of these under  $\Pi, \Sigma$  and reindexing (Corollary 4.3), so is  $\mathcal{U}_1^\Delta$ . By a similar argument, also the families  $\mathcal{E}l_1^\Delta, \mathcal{E}l_2^\Delta$  and  $\Pi(\mathcal{E}l_1^\Delta, \mathcal{E}l_2^\Delta)$  are invariant under clock introduction. By Lemma 7.3 there is a unique morphism  $\ulcorner \Pi_\Delta^\urcorner : \mathcal{U}_1^\Delta \rightarrow \mathcal{U}^\Delta$  in  $\text{GR}(\Delta)$  such that

$$\Pi(\mathcal{E}l_1^\Delta, \mathcal{E}l_2^\Delta) = \mathcal{E}l^\Delta[\ulcorner \Pi_\Delta^\urcorner]$$

The element

$$\ulcorner \Pi^{\times\urcorner}(\ulcorner A^\urcorner, \ulcorner B^\urcorner) \stackrel{\text{def}}{=} \ulcorner \Pi_\Delta^\urcorner[\langle \chi \rangle](\langle \ulcorner A^\urcorner, \lambda(\ulcorner B^\urcorner) \rangle)$$

is then welldefined by Lemma 7.4 and satisfies

$$\begin{aligned} \mathcal{E}l^\chi [\langle \text{id}_\Gamma, \ulcorner \Pi^{\chi \neg} (\ulcorner A^\neg, \ulcorner B^\neg \urcorner) \urcorner \rangle] &= \Pi(\mathcal{E}l_1^\chi, \mathcal{E}l_2^\chi) [\langle \text{id}_\Gamma, \langle \ulcorner A^\neg, \lambda(\ulcorner B^\neg \urcorner) \rangle \rangle] \\ &= \Pi(A, B) \end{aligned}$$

For the last statement, note that the map  $\text{in}_{\Delta, \Delta'} : \mathcal{U}^\Delta[\pi_{\Delta, \Delta'}] \rightarrow \mathcal{U}^{\Delta'}$  in  $\text{GR}(\Delta')$  from the proof of Proposition 7.2 induces a map  $\mathcal{U}_1^\Delta[\pi_{\Delta, \Delta'}] \rightarrow \mathcal{U}_1^{\Delta'}$  mapping  $\langle A, B \rangle$  to  $\langle \text{in}_{\Delta, \Delta'}(A), \text{in}_{\Delta, \Delta'} \circ B \rangle$ . This makes the following diagram commute

$$\begin{array}{ccc} \mathcal{U}_1^\Delta[\pi_{\Delta, \Delta'}] & \longrightarrow & \mathcal{U}_1^{\Delta'} \\ \ulcorner \Pi_{\Delta \neg} \urcorner[\pi_{\Delta, \Delta'}] \downarrow & & \downarrow \ulcorner \Pi_{\Delta'} \urcorner \\ \mathcal{U}^\Delta[\pi_{\Delta, \Delta'}] & \xrightarrow{\text{in}_{\Delta, \Delta'}} & \mathcal{U}^{\Delta'} \end{array}$$

by the uniqueness statement of Lemma 7.3 because reindexing  $\mathcal{E}l^{\Delta'}$  along either direction gives  $\Pi(\mathcal{E}l_1^\Delta[\pi_{\Delta, \Delta'}], \mathcal{E}l_2^\Delta[\langle \pi_{\Delta, \Delta'} \circ p, q \rangle])$ . From this the final statement follows.  $\square$

### 7.6. Universal quantification over clocks

We now describe the codes for universal quantification over clocks. Even though clock quantification is modelled using  $\Pi$ -types, this is not a special case of Proposition 7.3, since on the level of codes, clock quantification involves a change of universe.

**Proposition 7.4.** Suppose  $A \in \text{Set}^\mathbb{T}(\Gamma.\text{clock}[\!|], \chi)$ , and that  $\chi$  is a set of elements of  $\text{clock}[\!|]_\Gamma$ . Write  $\chi[p], q$  for the union of the set  $\chi[p]$  and  $q$ , and suppose

$$\ulcorner A^\neg \urcorner \in \text{Set}^\mathbb{T}(\Gamma.\text{clock}[\!|] \vdash \mathcal{U}^{\chi[p], q})$$

is such that  $A = \mathcal{E}l^{\chi[p], q}[\langle \text{id}_{\Gamma.\text{clock}[\!|]}, \ulcorner A^\neg \urcorner \rangle]$ . There is an element  $\ulcorner \forall^{\chi \neg} (\lambda(\ulcorner A^\neg \urcorner)) \urcorner \in \text{Set}^\mathbb{T}(\Gamma \vdash \mathcal{U}^X)$  such that

$$\mathcal{E}l^\chi [\langle \text{id}_\Gamma, \ulcorner \forall^{\chi \neg} (\lambda(\ulcorner A^\neg \urcorner)) \urcorner \rangle] = \Pi(\text{clock}[\!|], A)$$

Moreover, if  $\rho : \Gamma' \rightarrow \Gamma$  then

$$(\ulcorner \forall^{\chi \neg} (\lambda(\ulcorner A^\neg \urcorner)) \urcorner)[\rho] = \ulcorner \forall^{\chi[\rho] \neg} (\lambda(\ulcorner A^\neg \urcorner[\langle \rho \circ p, q \rangle])) \urcorner$$

and if  $\chi \subseteq \chi'$  then

$$\text{in}_{\chi, \chi'}(\ulcorner \forall^{\chi \neg} (\lambda(\ulcorner A^\neg \urcorner)) \urcorner) = \ulcorner \forall^{\chi' \neg} (\lambda(\text{in}_{(\chi[p], q), (\chi'[p], q)}(\ulcorner A^\neg \urcorner))) \urcorner$$

*Proof.* The universe  $\mathcal{U}^{\Delta, \kappa}$  is an object in  $\text{GR}(\Delta, \kappa)$ , which means that it is a family over  $\mathcal{C}^{\Delta, \kappa}$ . Abusing notation slightly, write  $\mathcal{U}^{\Delta, q}$  for the family over  $\mathcal{C}^{\Delta, \text{clock}[\!|]}$  obtained by reindexing  $\mathcal{U}^{\Delta, \kappa}$  along the isomorphism  $\mathcal{C}^{\Delta, \kappa} \cong \mathcal{C}^{\Delta, \text{clock}[\!|]}$  and write  $\mathcal{E}l^{\Delta, q}$  for the result of reindexing the family  $\mathcal{E}l^{\Delta, \kappa}$  along the same map. Note that  $\mathcal{U}^{\Delta, q}[\langle \langle \chi \rangle \circ p, q \rangle] = \mathcal{U}^{\chi[p], q}$  and  $\mathcal{E}l^{\Delta, q}[\langle \langle \langle \chi \rangle \circ pp, q[p] \rangle, q \rangle] = \mathcal{E}l^{\chi[p], q}$ .

We will now construct the generic clock quantified family over  $\mathcal{C}^{\Delta, \Pi(\text{clock}[\!|], \mathcal{U}^{\Delta, q})}$  and construct the  $\ulcorner \forall^{\chi \neg} (\lambda(\ulcorner A^\neg \urcorner)) \urcorner$  using Lemma 7.3. First observe that

$$\text{ev}(q[p], q) \in \text{Set}^\mathbb{T}(\mathcal{C}^{\Delta, \Pi(\text{clock}[\!|], \mathcal{U}^{\Delta, q}).\text{clock}[\!|]} \vdash \mathcal{U}^{\Delta, q}[\langle pp, q \rangle])$$

and so

$$\mathcal{E}l^{\Delta, q}[\langle \langle pp, q \rangle, \text{ev}(q[p], q) \rangle] \in \text{Set}^\mathbb{T}(\mathcal{C}^{\Delta, \Pi(\text{clock}[\!|], \mathcal{U}^{\Delta, q}).\text{clock}[\!|]})$$

and

$$\Pi(\text{clock}[!], \mathcal{E}l^{\Delta, \mathfrak{q}}[\langle\langle \text{pp}, \mathfrak{q} \rangle, \text{ev}(\mathfrak{q}[\text{p}], \mathfrak{q}) \rangle]) \in \text{Set}^{\mathbb{T}}(\mathcal{C}^{\Delta}. \Pi(\text{clock}[!], \mathcal{U}^{\Delta, \mathfrak{q}}))$$

Since  $\mathcal{U}^{\Delta, \kappa}$  and  $\mathcal{E}l^{\Delta, \kappa}$  are invariant under clock introduction and this notion is closed under reindexing and  $\Pi$ -types, also  $\Pi(\text{clock}[!], \mathcal{U}^{\Delta, \mathfrak{q}})$  and  $\Pi(\text{clock}[!], \mathcal{E}l^{\Delta, \mathfrak{q}}[\langle\langle \text{pp}, \mathfrak{q} \rangle, \text{ev}(\mathfrak{q}[\text{p}], \mathfrak{q}) \rangle])$  are invariant under clock introduction, and so by Lemma 7.3 there is a map

$$\ulcorner \forall^{\Delta} \urcorner : \Pi(\text{clock}[!], \mathcal{U}^{\Delta, \mathfrak{q}}) \rightarrow \mathcal{U}^{\Delta}$$

in  $\text{GR}(\Delta)$  such that

$$\Pi(\text{clock}[!], \mathcal{E}l^{\Delta, \mathfrak{q}}[\langle\langle \text{pp}, \mathfrak{q} \rangle, \text{ev}(\mathfrak{q}[\text{p}], \mathfrak{q}) \rangle]) = \mathcal{E}l^{\Delta}[\ulcorner \forall^{\Delta} \urcorner]$$

By Lemma 7.4 the element

$$\ulcorner \forall^{\chi} \urcorner(\lambda(\ulcorner A \urcorner)) \stackrel{\text{def}}{=} \ulcorner \forall^{\Delta} \urcorner[\langle \chi \rangle](\lambda(\ulcorner A \urcorner))$$

is welldefined and satisfies

$$\begin{aligned} \mathcal{E}l^{\chi}[\langle \text{id}_{\Gamma}, \ulcorner \forall^{\chi} \urcorner(\lambda(\ulcorner A \urcorner)) \rangle] &= \Pi(\text{clock}[!], \mathcal{E}l^{\chi[\text{p}], \mathfrak{q}}[\langle\langle \text{pp}, \mathfrak{q} \rangle, \text{ev}(\mathfrak{q}[\text{p}], \mathfrak{q}) \rangle])[\langle \text{id}_{\Gamma}, \lambda(\ulcorner A \urcorner) \rangle] \\ &= \Pi(\text{clock}[!], \mathcal{E}l^{\chi[\text{p}], \mathfrak{q}}[\langle\langle \text{pp}, \mathfrak{q} \rangle, \text{ev}(\mathfrak{q}[\text{p}], \mathfrak{q}) \rangle])[\langle \langle \text{id}_{\Gamma}, \lambda(\ulcorner A \urcorner) \rangle \circ \text{p}, \mathfrak{q} \rangle]) \\ &= \Pi(\text{clock}[!], \mathcal{E}l^{\chi[\text{p}], \mathfrak{q}}[\langle\langle \text{p}, \mathfrak{q} \rangle, \text{ev}(\lambda(\ulcorner A \urcorner)[\text{p}], \mathfrak{q}) \rangle]) \\ &= \Pi(\text{clock}[!], \mathcal{E}l^{\chi[\text{p}], \mathfrak{q}}[\langle \text{id}, \ulcorner A \urcorner \rangle]) \\ &= \Pi(\text{clock}[!], A) \end{aligned}$$

This construction is clearly closed under reindexing, and the last statement can be proved similarly to the last statement of Proposition 7.3.  $\square$

### 7.7. Codes for the later modalities

**Proposition 7.5.** There is a mapping associating  $\kappa \in \chi$  and  $t \in \text{Set}^{\mathbb{T}}(\Gamma \vdash \blacktriangleright^{\kappa} \mathcal{U}^{\chi})$  to

$$\ulcorner \blacktriangleright^{\kappa} \urcorner(t) \in \text{Set}^{\mathbb{T}}(\Gamma \vdash \mathcal{U}^{\chi})$$

such that if  $\Gamma'$  is a telescope of length  $m$  over  $\Gamma$  and  $\ulcorner A \urcorner \in \text{Set}^{\mathbb{T}}(\Gamma. \Gamma' \vdash \mathcal{U}^{\chi[\text{p}^m]})$  and  $A = \mathcal{E}l^{\chi[\text{p}^m]}[\langle \text{id}_{\Gamma. \Gamma'}, \ulcorner A \urcorner \rangle]$  and  $\xi \in \text{Set}^{\mathbb{T}}(\Gamma \rightarrow^{\kappa} \Gamma')$  then

$$\mathcal{E}l^{\chi}[\langle \text{id}_{\Gamma}, \ulcorner \blacktriangleright^{\kappa} \urcorner(\text{next}^{\kappa} \xi. \ulcorner A \urcorner) \rangle] = \ulcorner \blacktriangleright^{\kappa} \urcorner \xi. A$$

Moreover, if  $\rho : \Gamma'' \rightarrow \Gamma$  then  $\ulcorner \blacktriangleright^{\kappa} \urcorner(t)[\rho] = \ulcorner \blacktriangleright^{\kappa[\rho]} \urcorner(t[\rho])$ , and if  $\chi \subseteq \chi'$  then

$$\text{in}_{\chi, \chi'}(\ulcorner \blacktriangleright^{\kappa} \urcorner(t)) = \ulcorner \blacktriangleright^{\kappa} \urcorner(\text{next}^{\kappa}(t). \text{in}_{\chi[\text{p}], \chi'[\text{p}]}(\mathfrak{q}))$$

A few of the typings of the proposition need to be explained. The element  $\text{next}^{\kappa} \xi. \ulcorner A \urcorner$  is a priori an element of  $\ulcorner \blacktriangleright^{\kappa} \urcorner \xi. (\mathcal{U}^{\chi[\text{p}^m]})$  but the latter family equals  $\ulcorner \blacktriangleright^{\kappa} \urcorner \mathcal{U}^{\chi}$  and so  $\ulcorner \blacktriangleright^{\kappa} \urcorner(\text{next}^{\kappa} \xi. \ulcorner A \urcorner)$  is well formed. In the last equation,  $t$  is considered a delayed sequence of elements in  $\text{Set}^{\mathbb{T}}(\Gamma \rightarrow^{\kappa} \mathcal{U}^{\chi})$ , and  $\text{in}_{\chi[\text{p}], \chi'[\text{p}]}(\mathfrak{q}) \in \text{Set}^{\mathbb{T}}(\Gamma. \mathcal{U}^{\chi} \vdash \mathcal{U}^{\chi'[\text{p}]})$  and so

$$\text{next}^{\kappa}(t). \text{in}_{\chi[\text{p}], \chi'[\text{p}]}(\mathfrak{q}) \in \text{Set}^{\mathbb{T}}(\Gamma \vdash \ulcorner \blacktriangleright^{\kappa} \urcorner(t). \mathcal{U}^{\chi'[\text{p}]})) = \text{Set}^{\mathbb{T}}(\Gamma \vdash \ulcorner \blacktriangleright^{\kappa} \urcorner \mathcal{U}^{\chi'})$$

making the right hand side of the final equation well formed.

*Proof.* Suppose note that any  $\hat{\kappa} \in \Delta$  defines an element  $\hat{\kappa} \in \text{Set}^{\mathbb{T}}(\mathcal{C}^{\Delta} \vdash \text{clock}[\!])$  essentially by projection. Since  $q \in \text{Set}^{\mathbb{T}}(\mathcal{C}^{\Delta} \cdot \blacktriangleright \mathcal{U}^{\Delta} \vdash (\blacktriangleright \mathcal{U}^{\Delta})[p])$  it defines a delayed sequence of elements  $(q) \in \text{Set}^{\mathbb{T}}(\mathcal{C}^{\Delta} \cdot \blacktriangleright \mathcal{U}^{\Delta} \rightarrow^{\hat{\kappa}[p]} \mathcal{U}^{\Delta}[p])$ . Since moreover,  $\mathcal{E}l^{\Delta}[\langle pp, q \rangle]$  is a family in  $\text{Set}^{\mathbb{T}}(\mathcal{C}^{\Delta} \cdot \blacktriangleright \mathcal{U}^{\Delta} \cdot \mathcal{U}^{\Delta}[p])$  we can define

$$\blacktriangleright^{\hat{\kappa}[p]}(q) \cdot (\mathcal{E}l^{\Delta}[\langle pp, q \rangle]) \in \text{Set}^{\mathbb{T}}(\mathcal{C}^{\Delta} \cdot \blacktriangleright \mathcal{U}^{\Delta})$$

By Proposition 6.2, both  $\blacktriangleright^{\hat{\kappa}} \mathcal{U}^{\Delta}$  and  $\blacktriangleright^{\hat{\kappa}[p]}(q) \cdot (\mathcal{E}l^{\Delta}[\langle pp, q \rangle])$  are invariant under clock introduction and so by Lemma 7.3 there is a morphism  $\ulcorner \blacktriangleright^{\hat{\kappa}} \urcorner : \blacktriangleright^{\hat{\kappa}} \mathcal{U}^{\Delta} \rightarrow \mathcal{U}^{\Delta}$  in  $\text{GR}(\Delta)$  such that  $\mathcal{E}l^{\Delta}[\ulcorner \blacktriangleright^{\hat{\kappa}} \urcorner] = \blacktriangleright^{\hat{\kappa}[p]}(q) \cdot (\mathcal{E}l^{\Delta}[\langle pp, q \rangle])$ . Using this we define

$$\ulcorner \blacktriangleright^{\kappa} \urcorner(t) \stackrel{\text{def}}{=} \ulcorner \blacktriangleright^{\hat{\kappa}} \urcorner[\langle \chi \rangle](t)$$

where  $\hat{\kappa}$  is an element in  $\Delta$  mapped to  $\kappa$ . This can be proved independent of choice of  $\Delta$  and surjection  $\Delta \rightarrow \langle \chi \rangle$  and  $\hat{\kappa}$  using arguments as in the proof of Lemma 7.4. Then

$$\begin{aligned} \mathcal{E}l^{\chi} \left[ \left\langle \text{id}_{\Gamma}, \ulcorner \blacktriangleright^{\kappa} \urcorner(\text{next}^{\kappa} \xi \cdot \ulcorner A \urcorner) \right\rangle \right] &= (\blacktriangleright^{\kappa[p]}(q) \cdot \mathcal{E}l^{\chi}[\langle pp, q \rangle]) [\langle \text{id}_{\Gamma}, \text{next}^{\kappa} \xi \cdot \ulcorner A \urcorner \rangle] \\ &= \blacktriangleright^{\kappa}(\text{next}^{\kappa} \xi \cdot \ulcorner A \urcorner) \cdot (\mathcal{E}l^{\chi}[\langle pp, q \rangle]) \\ &= \blacktriangleright^{\kappa} \xi [\text{next}^{\kappa} \xi \cdot \ulcorner A \urcorner] \cdot (\mathcal{E}l^{\chi}[\langle pp, q \rangle]) [\langle p^{m+1}, q \rangle] \end{aligned}$$

where in the last step  $m$  is the length of  $\xi$ . Now by Theorem 6.1.1, the latter equals

$$\begin{aligned} \blacktriangleright^{\kappa} \xi \cdot (\mathcal{E}l^{\chi}[\langle pp, q \rangle]) [\langle p^m, \ulcorner A \urcorner \rangle] &= \blacktriangleright^{\kappa} \xi \cdot (\mathcal{E}l^{\chi}[\langle p^{m+1}, \ulcorner A \urcorner \rangle]) \\ &= \blacktriangleright^{\kappa} \xi \cdot (\mathcal{E}l^{\chi[p^m]}[\langle \text{id}, \ulcorner A \urcorner \rangle]) \\ &= \blacktriangleright^{\kappa} \xi \cdot A \end{aligned}$$

For the final statement, can be proved using the uniqueness statement of Lemma 7.3.  $\square$

## 8. Interpreting syntax

The previous sections define the semantic structure of the model corresponding to each of the constructions of GDTT. One can use this to define an interpretation of the syntax into the model, as we briefly sketch here. As is well known, defining interpretation of dependent type theories is not a simple procedure. In particular, the proof of welldefinedness of the interpretation can not be separated from the proof of soundness. Here we follow the approach of Hofmann (1997), which first defines an interpretation of (pre-) contexts, types and term as a partial function, then proves that this function is defined on all wellformed judgements. To define the partial interpretation function, syntax must be annotated with typing information, meaning that the syntax interpreted is not quite the syntax usually presented for dependent type theory. For example,  $\lambda$ -abstractions must be annotated with not just the type of the variable being abstracted, but also with the target type of the function created (which is a dependent family  $(x.A)$ ). Likewise, application is annotated both with the domain type and with the dependent codomain type.

Definedness of the interpretation of well formed judgements is then proved by induction on the structure of judgements. This must be done simultaneously with the proof of soundness of

the interpretation and with the proof of a substitution lemma. We now sketch how each of these ingredients must be adapted to interpret GDTT.

The annotation of terms and types must be extended to the new constructions. Universal quantification over clocks is interpreted as a  $\Pi$ -type, and the annotations must therefore be similar to those of  $\Pi$ -types. Terms like  $\text{next}^\kappa$  and  $\text{fix}^\kappa$  must be annotated with the type at which they are applied. Recall from Section 3.3 that  $\text{prev}$  is compiled away in an initial step using  $\text{force}$ . The constant  $\text{force}$  must be annotated with the dependent type  $(\kappa.A)$  at which it is applied. Type operations on the universe must be annotated with the context  $\Delta$  at which they are applied. The type constructor  $\blacktriangleright^\kappa \xi.A$  must be annotated with the types in the telescope and likewise for  $\text{next}^\kappa \xi.t$ . A notion of pre- delayed substitutions must be defined and these must be (partially) interpreted as delayed sequences of elements.

Once the partial interpretation function has been interpreted, the welldefinedness of the interpretation of wellformed judgements must be proved by induction on judgements simultaneously with soundness and a substitution lemma. To this sequence of lemmas must be added the statement that the interpretation of any type is invariant under clock introduction.

The substitution lemma is mostly standard. In particular, the notation of substitution between contexts can be defined essentially as usual

$$\frac{}{\cdot : \Gamma \rightarrow \cdot} \quad \frac{\rho : \Gamma \rightarrow \Gamma' \quad \Gamma \vdash t : A\rho}{\rho[x \mapsto t] : \Gamma \rightarrow \Gamma', x : A} \quad \frac{\rho : \Gamma \rightarrow \Gamma' \quad \Gamma \vdash \kappa' : \text{clock}}{\rho[\kappa \mapsto \kappa'] : \Gamma \rightarrow \Gamma', \kappa : \text{clock}}$$

and likewise the notion of substitution is defined in the standard way. Note in particular that this means that  $(U_\Delta)\rho = U_{\Delta\rho}$ . Substitution on a delayed substitution  $\xi$  is defined by distributing the interpretation over the terms in  $\xi$ . The substitution lemma is as follows.

**Lemma 8.1.** If  $\rho : \Gamma' \rightarrow \Gamma$  is a substitution, then

- if  $\Gamma \vdash A$  type also  $\Gamma' \vdash A\rho$  type and  $\llbracket \Gamma' \vdash A\rho \rrbracket = \llbracket \Gamma \vdash A \rrbracket \llbracket \rho \rrbracket$ .
- if  $\Gamma \vdash t : A$  also  $\Gamma' \vdash t\rho : A\rho$  and  $\llbracket \Gamma' \vdash t\rho \rrbracket = \llbracket \Gamma \vdash t \rrbracket \llbracket \rho \rrbracket$
- if  $\xi : \Gamma \rightarrow^\kappa \Gamma''$  then also  $\xi\rho : \Gamma' \rightarrow^{\kappa\rho} \Gamma''\rho$  and  $\llbracket \xi\rho \rrbracket = \llbracket \xi \rrbracket \llbracket \rho \rrbracket$ .

## 9. Recovering the categories $\mathfrak{GR}(\Delta)$

In this final section we discuss the relation to the family of categories  $\mathfrak{GR}(\Delta)$  defined in previous work by the authors (Bizjak & Møgelberg 2015). As mentioned in the introduction, this gives a model of guarded recursion with multiple clocks up to a coherence problem. We first recall the definition of the categories  $\mathfrak{GR}(\Delta)$  (note that the notation for this differs from the  $\text{GR}(\Delta)$  used in the paper only by the choice of font).

For a finite set of clock variables  $\Delta$  the category  $\mathfrak{GR}(\Delta)$  is the category of presheaves on the poset  $\mathfrak{J}(\Delta)$ . The elements of this poset are pairs  $(E, \delta)$  where  $E$  is an equivalence relation on  $\Delta$  and  $\delta : \Delta \rightarrow \mathbb{N}$  is a function which respects the equivalence relation  $E$ . The order on  $\mathfrak{J}(\Delta)$  is defined so that  $(E, \delta) \leq (E', \delta')$  if  $E$  is coarser than  $E'$  (i.e.,  $E' \subseteq E$  as subsets of  $\Delta \times \Delta$ ) and  $\delta$  is pointwise less than  $\delta'$ . The idea behind this poset is that  $\delta$  records how much time is left on each clock, and the equivalence relation  $E$  states which clocks are identified. The order is defined so that we can pass to a state where there is less time available on each clock, but we can also identify different clocks, i.e., make the equivalence relation coarser.

The intention of the categories  $\mathfrak{GR}(\Delta)$  is that types and terms in clock variable context  $\Delta$

should be modelled in  $\mathfrak{GR}(\Delta)$ . In the present paper, the corresponding fragment is modelled in  $\mathfrak{GR}(\Delta)$  with the restriction that families must be invariant under clock introduction. Thus the next theorem states that the two models are equivalent.

**Theorem 9.1.** Let  $\Delta$  be a finite set of clocks. The full subcategory of  $\mathfrak{GR}(\Delta)$  on objects invariant under clock introduction is equivalent to the category  $\mathfrak{GR}(\Delta)$ .

*Proof (sketch)* Recall that  $\mathfrak{GR}(\Delta)$  is defined as the category of covariant presheaves on the category of elements of  $\mathcal{C}^\Delta$ , for which we write  $\mathcal{R}(\Delta)$  in this proof. The indexing poset  $\mathcal{J}(\Delta)$  is equivalent to the preorder  $\mathcal{S}(\Delta)^{\text{op}}$  where  $\mathcal{S}(\Delta)$  is the full subcategory of  $\mathcal{R}(\Delta)$  on those objects  $(\mathcal{E}, \delta, f)$  where  $f$  is surjective. Indeed, this equivalence follows from the fact that every function  $f$  on  $\Delta$  determines an equivalence relation on  $\Delta$ , and every equivalence relation  $E$  on  $\Delta$  gives rise to the surjective quotient function  $q : \Delta \rightarrow \Delta/E$ . Straightforward calculations show this extends to the claimed equivalence of the poset  $\mathcal{J}(\Delta)$  and the preorder  $\mathcal{S}(\Delta)^{\text{op}}$ .

Thus we have that  $\mathfrak{GR}(\Delta)$  is equivalent to the category of covariant presheaves on  $\mathcal{S}(\Delta)$ . By definition there is an inclusion functor  $i : \mathcal{S}(\Delta) \rightarrow \mathcal{R}(\Delta)$  which gives rise, by precomposition, to a functor  $i^* : \mathfrak{GR}(\Delta) \rightarrow \mathfrak{GR}(\Delta)$ . Moreover, there is a functor  $g : \mathcal{R}(\Delta) \rightarrow \mathcal{S}(\Delta)$  which maps  $(\mathcal{E}, \delta, f)$  to  $(f[\Delta], \delta|_{f[\Delta]}, f)$ . This functor gives rise to a functor  $g^* : \mathfrak{GR}(\Delta) \rightarrow \mathfrak{GR}(\Delta)$ . It is easy to see  $g \circ i = \text{id}$  and that there is a natural transformation  $\varepsilon : i \circ g \rightarrow \text{id}$  whose component at  $(\mathcal{E}, \delta, f)$  is given by the inclusion  $f[\Delta] \rightarrow \mathcal{E}$ . These transformations define an adjunction  $i \dashv g$ .

Thus  $g^* \dashv i^*$  as well; first from  $g \circ i = \text{id}$  we have  $i^* \circ g^* = \text{id}$ , and so the unit  $\eta^*$  of the adjunction is the identity natural transformation, and, second, from the counit of the adjunction  $i \dashv g$  we define the counit  $\varepsilon^*$  of the adjunction  $g^* \dashv i^*$  pointwise, as in

$$(\varepsilon_X^*)_{(\mathcal{E}, \delta, f)} = X(\varepsilon_{(\mathcal{E}, \delta, f)}).$$

It is standard that an adjunction restricts to an equivalence of full subcategories  $\mathbb{C}$  of  $\mathfrak{GR}(\Delta)$  and  $\mathbb{D}$  of  $\mathfrak{GR}(\Delta)$  on objects where the unit and the counit are isomorphisms, respectively. Because the unit  $\eta^*$  of the adjunction  $g^* \dashv i^*$  is an isomorphism the category  $\mathbb{C}$  is  $\mathfrak{GR}(\Delta)$ .

The category  $\mathbb{D}$  on the other hand is the category of those objects  $X \in \mathfrak{GR}(\Delta)$  where for every  $(\mathcal{E}, \delta, f) \in \mathcal{R}(\Delta)$  the component of the counit

$$(\varepsilon_X^*)_{(\mathcal{E}, \delta, f)} = X(\varepsilon_{(\mathcal{E}, \delta, f)}) = X(\iota)$$

where  $\iota : (f[\Delta], \delta|_{f[\Delta]}, f) \rightarrow (\mathcal{E}, \delta, f)$  is the inclusion, is an isomorphism. By Lemma 4.1 this holds precisely when the object  $X$  is invariant under clock introduction. Hence, the adjunction  $g^* \dashv i^*$  restricts to the equivalence of  $\mathfrak{GR}(\Delta)$  and the full subcategory of  $\mathfrak{GR}(\Delta)$  on objects invariant under clock introduction.  $\square$

Notice, however, that the categories in Theorem 9.1 are *not* isomorphic. This is the key to achieving preservation of structure, chiefly dependent products, in the present model, up to *equality*, as opposed to only up to isomorphism, as in the previous model (Bizjak & Møgelberg 2015).

## Acknowledgements

We thank Patrick Bahr, Lars Birkedal, Hans Bugge Grathwohl and Bassel Manna for helpful discussions. We thank the anonymous reviewers for helpful suggestions which led to a major revision significantly improving the paper. Bizjak was supported by the ModuRes Sapere Aude

Advanced Grant from The Danish Council for Independent Research for the Natural Sciences (FNU). Møgelberg was supported by a research grant (13156) from VILLUM FONDEN and DFF-Research Project 1 Grant no. 4002-00442, from The Danish Council for Independent Research for the Natural Sciences (FNU).

## References

- Abel, A. & Pientka, B. (2013), Wellfounded recursion with copatterns: A unified approach to termination and productivity, *in* ‘Proceedings ICFP 2013’, ACM, pp. 185–196.
- Abel, A., Vezzosi, A. & Winterhalter, T. (2017), ‘Normalization by evaluation for sized dependent types’, *PACMPL* **1**(ICFP), 33:1–33:30.  
**URL:** <https://doi.org/10.1145/3110277>
- Appel, A. W. & McAllester, D. A. (2001), ‘An indexed model of recursive types for foundational proof-carrying code’, *ACM Trans. Program. Lang. Syst* **23**(5), 657–683.
- Appel, A. W., Melliès, P., Richards, C. D. & Vouillon, J. (2007), A very modal model of a modern, major, general type system, *in* ‘POPL’, pp. 109–122.
- Atkey, R. & McBride, C. (2013), Productive coprogramming with guarded recursion, *in* ‘Proceedings of ICFP 2013’, ACM, pp. 197–208.
- Awodey, S. & Warren, M. A. (2009), Homotopy theoretic models of identity types, *in* ‘Mathematical Proceedings of the Cambridge Philosophical Society’, Vol. 146, Cambridge University Press, pp. 45–55.
- Bahr, P., Grathwohl, H. B. & Møgelberg, R. E. (2017), The clocks are ticking: No more delays!, *in* ‘32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017’, pp. 1–12.
- Bezem, M., Coquand, T. & Huber, S. (2013), A model of type theory in cubical sets, *in* ‘19th International Conference on Types for Proofs and Programs, TYPES 2013, April 22-26, 2013, Toulouse, France’, pp. 107–128.
- Birkedal, L., Bizjak, A., Clouston, R., Grathwohl, H. B., Spitters, B. & Vezzosi, A. (2016), Guarded cubical type theory: Path equality for guarded recursion, *in* ‘25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France’, pp. 23:1–23:17.
- Birkedal, L., Møgelberg, R. E., Schwinghammer, J. & Støvring, K. (2012), ‘First steps in synthetic guarded domain theory: step-indexing in the topos of trees’, *Logical Methods in Computer Science* **8**(4).
- Bizjak, A., Birkedal, L. & Miculan, M. (2014), A model of countable nondeterminism in guarded type theory, *in* ‘RTA-TLCA’, pp. 108–123.
- Bizjak, A., Grathwohl, H. B., Clouston, R., Møgelberg, R. E. & Birkedal, L. (2016), Guarded dependent type theory with coinductive types, *in* ‘FoSSaCS’, pp. 20–35.
- Bizjak, A. & Møgelberg, R. E. (2015), ‘A model of guarded recursion with clock synchronisation’, *Electronic Notes in Theoretical Computer Science* **319**, 83 – 101. The 31st Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXI).
- Cohen, C., Coquand, T., Huber, S. & Mörtberg, A. (2016), ‘Cubical type theory: a constructive interpretation of the univalence axiom’, *CoRR* **abs/1611.02108**.  
**URL:** <http://arxiv.org/abs/1611.02108>
- Danielsson, N. A. (2010), Beating the productivity checker using embedded languages, *in* ‘PAR’, Vol. 43, pp. 29–48.
- Dybjer, P. (1995), Internal type theory, *in* ‘International Workshop on Types for Proofs and Programs’, Springer, pp. 120–134.
- Hofmann, M. (1994), On the interpretation of type theory in locally cartesian closed categories, *in* ‘Proceedings of Computer Science Logic, Lecture Notes in Computer Science’, Springer, pp. 427–441.

- Hofmann, M. (1997), Syntax and semantics of dependent types, *in* ‘Extensional Constructs in Intensional Type Theory’, Springer, pp. 13–54.
- Hofmann, M. & Streicher, T. (1999), Lifting Grothendieck universes. Unpublished.  
**URL:** [www.mathematik.tu-darmstadt.de/~streicher/NOTES/lift.pdf](http://www.mathematik.tu-darmstadt.de/~streicher/NOTES/lift.pdf)
- Hughes, J., Pareto, L. & Sabry, A. (1996), Proving the correctness of reactive systems using sized types, *in* ‘Conference Record of POPL’96: The 23rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Papers Presented at the Symposium, St. Petersburg Beach, Florida, USA, January 21-24, 1996’, pp. 410–423.
- Hyland, J. M. E., Robinson, E. P. & Rosolini, G. (1990), ‘The discrete objects in the effective topos’, *Proceedings of the London mathematical society* **3**(1), 1–36.
- Kapulkin, C. & Lumsdaine, P. L. (2012), ‘The simplicial model of univalent foundations (after voevodsky)’, *CoRR* **abs/1211.2851**.  
**URL:** <https://arxiv.org/abs/1211.2851>
- Luo, Z. (1994), *Computation and Reasoning. A Type Theory for Computer Science*, number 11 *in* ‘International Series of Monographs on Computer Science’, Oxford University Press.
- MacLane, S. (1998), *Categories for the Working Mathematician*, Graduate Texts in Mathematics, second edn, Springer New York.
- Mannaa, B. & Møgelberg, R. E. (2018), The clocks they are adjunctions denotational semantics for clocked type theory, *in* ‘3rd International Conference on Formal Structures for Computation and Deduction, FSCD 2018, July 9-12, 2018, Oxford, UK’, pp. 23:1–23:17.
- Martin-Löf, P. (1973), An intuitionistic theory of types: Predicative part, *in* H. Rose & J. Shepherdson, eds, ‘Logic Colloquium’, North-Holland, Amsterdam, pp. 73–118.
- McBride, C. & Paterson, R. (2008), ‘Applicative programming with effects’, *J. Funct. Programming* **18**(1), 1–13.
- Møgelberg, R. E. (2014), A type theory for productive coprogramming via guarded recursion, *in* ‘Proceedings of CSL-LICS 2014’, ACM, pp. 71:1–71:10.
- Møgelberg, R. E. & Paviotti, M. (2016), Denotational semantics of recursive types in synthetic guarded domain theory, *in* ‘Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS ’16, New York, NY, USA, July 5-8, 2016’, pp. 317–326.
- Nakano, H. (2000), A modality for recursion, *in* ‘Proceedings of LICS 2000’, IEEE, pp. 255–266.
- Norell, U. (2007), Towards a practical programming language based on dependent type theory, PhD thesis, Chalmers University of Technology.
- Sacchini, J. L. (2013), Type-based productivity of stream definitions in the calculus of constructions, *in* ‘28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013’, pp. 233–242.
- Sterling, J. & Harper, R. (2018), Guarded computational type theory, *in* ‘Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science’, LICS ’18, ACM, New York, NY, USA, pp. 879–888.  
**URL:** <http://doi.acm.org/10.1145/3209108.3209153>
- Svendsen, K. & Birkedal, L. (2014), Impredicative concurrent abstract predicates, *in* ‘ESOP’.
- The Coq Development Team (2004), *The Coq proof assistant reference manual*, LogiCal Project. Version 8.0.  
**URL:** <http://coq.inria.fr>
- Veltri, N. & van der Weide, N. (2019), Guarded recursion in agda via sized types, *in* ‘4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany’, pp. 32:1–32:19.