Theoretical Computer Science 722 (2018) 1–18

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A model of guarded recursion via generalised equilogical spaces

Aleš Bizjak *, Lars Birkedal

Department of Computer Science, Aarhus University, Denmark

ARTICLE INFO

Article history:
Received 1 October 2015
Received in revised form 6 March 2017
Accepted 12 February 2018
Available online 21 February 2018
Communicated by D. Sannella

Keywords:
Semantics
Dependent type theory
Guarded recursion
Equilogical spaces

ABSTRACT

We present a new model, called GuardedEqu, of guarded dependent type theory using generalised equilogical spaces. GuardedEqu models guarded recursive types, which can be used to program with coinductive types and we prove that GuardedEqu ensures that all definable functions on coinductive types, e.g., streams, are continuous with respect to the natural topology. We present a direct, elementary, construction of the new model, which, importantly, is coherent (split) by construction.

1. Introduction

Type theories with support for guarded recursive functions and guarded recursive types are useful for programming with coinductive types and also for serving as a meta-theory for constructing sophisticated models of programming languages with effects [1,2].

In this paper, we present a new model of guarded dependent type theory, based on a generalisation of equilogical spaces [3]. We refer to the new model as GuardedEqu.

In contrast to earlier models of guarded dependent type theory, GuardedEqu ensures that definable functions on coinductive types are suitably continuous. For example, any function definable on the type of streams is continuous with respect to the standard topology on streams. Thus, if f is such a function on streams and xs is a stream, a finite amount of the output $f(xs)$ only depends on a finite amount of the input xs. We prove that an analogous result holds for final coalgebras of arbitrary polynomial functors.

It is well-known that models of dependent type theory can be tricky to construct. A virtue of GuardedEqu is that the model construction is quite elementary and can be presented via a simple generalisation of constructions known from realizability models of type theory. An important feature of GuardedEqu is that it is coherent (split) by construction. A limitation of the model is that it does not include universes.

We now explain how GuardedEqu is related to earlier models of variations of guarded type theory.

Originally a type theory with a single ▸ modality for expressing guardedness was modelled using the category $\mathrm{PSh}(\omega)$, the topos of trees [1]. The model and the type theory allows for the solution of guarded recursive domain equations. It was later realised that guarded recursion can also be used for ensuring that functions producing values of coinductive types

* Corresponding author.
E-mail addresses: abizjak@cs.au.dk (A. Bizjak), birkedal@cs.au.dk (L. Birkedal).

https://doi.org/10.1016/j.tcs.2018.02.012

are productive in a precise sense. To support such encodings the type theory needs to be extended with the ability to eliminate $\blacktriangleright$ in a controlled way. This led Atkey and McBride [4] to generalise $\blacktriangleright$ to a family of modalities indexed by clocks, and to support clock quantification for controlled elimination of $\blacktriangleright$. Atkey and McBride's development was for a simply typed calculus. They developed a model of their type theory and showed that, e.g., all streams definable in the calculus were interpreted as actual streams, i.e., non bottom elements. Møgelberg [5] extended their work to a model of dependent type theory with universes. This model was subsequently refined [6] to support clock synchronisation which considerably simplified the calculus. As it currently stands this model is complex, in particular in its split form which is needed to soundly model the rules of guarded dependent type theory [7].

GuardedEqu can be seen as a generalisation of the model by Atkey and McBride to a dependent type theory. The type theory we are considering has, to be useful, certain type isomorphisms [4,5]. An example is the type isomorphism $\forall\kappa.\mathbb{N} \cong \mathbb{N}$, where $\mathbb{N}$ is the type of natural numbers. In Atkey and McBride's model these were type equalities, but in the presheaf models of guarded dependent type theory [5,6] the types $\forall\kappa.\mathbb{N}$ and $\mathbb{N}$ are only modelled as canonically[1] isomorphic types. In GuardedEqu these type isomorphisms are again type equalities. This generalises the results of Atkey and McBride to a dependent type theory.

Overview of the paper In Section 2 we introduce the basics of guarded dependent type theory, referring to previous work for more details.

Then in Section 3 we construct a general class of models of dependent type theory with dependent products, sums, and extensional equality. These models are parametrised by posets P.

In Section 4 we instantiate these general models with specific posets $\mathfrak{I}(\Delta)$ to also model the later modality and clock quantification. Moreover we show that models of dependent type theory for different $\mathfrak{I}(\Delta)$ combine into a model of polymorphic dependent type theory [8, Chapter 11], satisfying additional axioms governing the interaction between the later modality and clock quantification.

In Section 5 we show some of the advantages of GuardedEqu, namely that it shows that coinductive types definable in guarded dependent type theory are interpreted as spaces with the expected topology and that definable functions are realised by continuous functions with respect to this topology. We illustrate this concretely on the example of streams.

Finally in Section 6 we compare GuardedEqu to previous models an discuss its advantages and some of its deficiencies.

2. Guarded dependent type theory

In this section we give a very brief introduction to the syntax of guarded dependent type theory. We refer the reader to [7] for the full set of typing rules, their motivation and more detailed explanation.

Guarded dependent type theory can be seen as a version of polymorphic dependent type theory [8]. It includes two contexts. A context Δ of clock variables $\kappa, \kappa', \cdots$ and a context Γ of ordinary term variables. Types depend on clocks, that is, clocks can appear in types, but clocks are only names in the sense that there are no constructions on clocks themselves. Guarded dependent type theory has the following basic judgements.

$$\Gamma \vdash_\Delta$$
$$\Gamma \vdash_\Delta A \text{ type}$$
$$\Gamma \vdash_\Delta t : A$$

The judgement $\Gamma \vdash_\Delta$ expresses that the free clocks in Γ are contained in Δ, the judgement $\Gamma \vdash_\Delta A$ type expresses that A is a well-formed type in context $\Gamma \vdash_\Delta$ and the last judgement expresses that t has type A in context $\Gamma \vdash_\Delta$. As usual in dependent type theory, there are also judgements for type and term equality for which we refer to [7]. Clocks are used to distinguish different $\blacktriangleright$ modalities. Thus, for each clock there is a modality $\blacktriangleright^\kappa$ and a term next^κ with the typing judgement

$$\frac{\Gamma \vdash_\Delta t : A}{\Gamma \vdash_\Delta \text{next}^\kappa\, t : \blacktriangleright^\kappa A}\,\kappa \in \Delta$$

Clock weakening is admissible, e.g., there is a derivation of

$$\frac{\Gamma \vdash_\Delta A \text{ type}}{\Gamma \vdash_{\Delta,\kappa} A \text{ type}}$$

for $\kappa \notin \Delta$ and analogously for other judgements. Clock weakening has a right adjoint, which we write as $\forall\kappa$. The introduction rule is

$$\frac{\Gamma \vdash_\Delta \qquad \Gamma \vdash_{\Delta,\kappa} t : A}{\Gamma \vdash_\Delta \Lambda\kappa.t : \forall\kappa.A}$$

[1] Canonical means there is a term of type $\mathbb{N} \to \forall\kappa.\mathbb{N}$ definable using just the ordinary introduction rules for $\forall\kappa$ and the denotation of this term is an isomorphism.