A Guarded Syntactic Model of Gradual Dependent Types

-we provide a syntactic model of a gradual dependently typed language in Guarded Cubical Type Theory.

This serves a dual purpose

Implementation

A syntactic model consists of a semantics-respecting translation from a gradual language to a static dependent language.

This provides a path to implementing a compiler for gradual dependent types by compiling through a static dependent language's existing compiler.

additionally, it provides a path to fast type-checking, specifically checking if two types normalize to consistent types. Idris's compiler contains experimental support for evaluating terms at compile-time by translating them to Chez Scheme, rather than the usual interpreter-style approach. Translating gradual to static allows such an evaluator to be used.

Metatheory

A syntactic model provides a relatively simple way to get a denotational account of gradual dependent types. It provides a path to proving the following, which were either difficult or tedious with a purely operational semantics:

A) Proving termination for approximate normalization

B) Proving the gradual guarantees,
and a weak version of New et al's
groduality. This establishes that costs
are not "lossy" at run-time, and
that the guarantees are not satisfied by simply producing? as a result for all casts.

c) Proving fully abstract embedding
of the static language. This
ensures that reasoning principles
that apply in fully static code are
not violated when that code is
used in gradual contexts

D) Proving that the composition operator computes the greatest lower bound. This justifies its design, establishing that no precision is needlessly lost.

Extinguishing the Fire Triangle

Lennon-Bertrand et al show no gradual language can have:

Conservatively extend
CIC
Strong Normalization
EP Pairs (New et. al.)

and 3 Are means to an end:

Strong normalization is used to show decidable type-checking, and EP-Pairs show the gradual guarantees, and that ? is not used as a result unnecessarily.

Instead, we show 1 , along with

Decidable type checking
Costs between precision-related types form a Galois Connection

is achieved with approximate normalisation,

so that type-checking computations terminate.

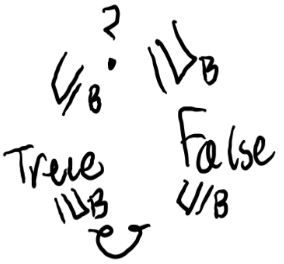
We use guarded type theory to provide a model of exod run-time execution while using only terminating computations at the type level. In the model, terms are represented as pairs of type:

$$(\triangleright [\![T]\!]_{exact} \times [\![T]\!]_{approx})$$

ie. an exact computation under the "later" modality of guarded type theory, paired with a terminating approximation.

lets us prove the gradual guarantees.

We can also show that ? is not produced unnecessarily by showing a Galois connection for exact execution for a different lattice. For approx and exact, we have:

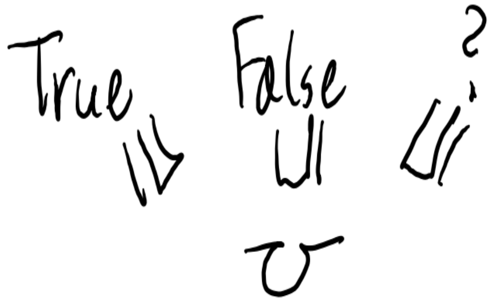


Semantic Precision:

$\Gamma \vDash t_1 \sqsubseteq t_2 : T$ if, for all $\gamma : (x{:}T)\Gamma \to \mathbb{B}$, if $\gamma(t_1) \to^* v_1$, then $\gamma(t_2) \to^* v_2$ where $v_1 \sqsubseteq_B v_2$

Then, if $\Gamma \vdash t_1 : T_1$, $\Gamma \vdash t_2 : T_2$, and $T_1 \sqsubseteq T_2$, we have $\Gamma \vDash \langle T_2 \Leftarrow T_1 \Leftarrow T_2 \rangle t_2 \sqsubseteq t_2$ and $\Gamma \vDash t_1 \sqsubseteq \langle T_1 \Leftarrow T_2 \Leftarrow T_1 \rangle t_1$

We * also * show the same Galois connection for exact execution, but for semantic precision defined over the following Boolean lattice:



That is, casting up then down may cause fewer errors, but will never change the final result. This is a more accurate adaptation of New et al's notion of precision as "errors less" to a language with ? as a term.

Results (Hopefully)

The goal is to provide translations:

$$T[\![\ ]\!]_{approx} : GTerm \to MLTT \subseteq GTT$$

$$\mathcal{E}_{approx}[\![\ ]\!] : GTerm \to MLTT \subseteq GTT$$

$$T_{exact}[\![\ ]\!] : GTerm \to GTT$$

$$\mathcal{E}_{exact}[\![\ ]\!] : GTerm \to GTT$$ such that

if $\Gamma \vdash t : T$

then $[\![\Gamma]\!] \vdash \mathcal{E}_{approx}[\![t]\!] : T_{approx}[\![T]\!]$

and $[\![\Gamma]\!] \vdash \mathcal{E}_{exact}[\![t]\!] : \triangleright T_{exact}[\![T]\!]$

Soundness:

if $t_1 \longrightarrow^* t_2 : T$ then $\mathcal{E}[\![t_1]\!] =_{T[\![T]\!]} \mathcal{E}[\![t_2]\!]$

for approx and exact

Other theorems to prove:

- Decidable checking
- Full abstraction
- Galois connection
- Weak consistency