GRADUAL TYPE PRECISION AS RETRACTION

Max S. New
maxnew@ccs.neu.edu

Semantics for Gradual Typing?

Dana Scott related typed and untyped semantics in the 70s, can that work provide insight into gradual typing?

We explore type *precision* and blame from this semantic point of view.

$$A \sqsubseteq ? \qquad \frac{A \sqsubseteq A' \qquad B \sqsubseteq B'}{(A \to B) \sqsubseteq (A' \to B')}$$

Originally called *naïve subtyping* due to its covariance, it has a constructive interpretation as retractions

What do Gradual Types Mean?

A syntactic type A denotes a *section-retraction pair* to the dynamic type in the cast language:

$$A \overset{s_A}{\rightarrowtail} ? \overset{r_A}{\twoheadrightarrow} A$$

such that $rs = \mathrm{id}_A$, which together we denote $A \rightleftarrows ?$. A retraction is a typed contract, the retraction enforces the contract and the section forgets values satisfy the contract.

From base types like

$$? \rightleftarrows ?, \mathbb{B} \rightleftarrows ?, \mathbb{N} \rightleftarrows ?, (? \to ?) \rightleftarrows ?$$

connectives build types compositionally:

$$(A \to B) \rightleftarrows (? \to ?) \rightleftarrows ?$$

Precision and Casts

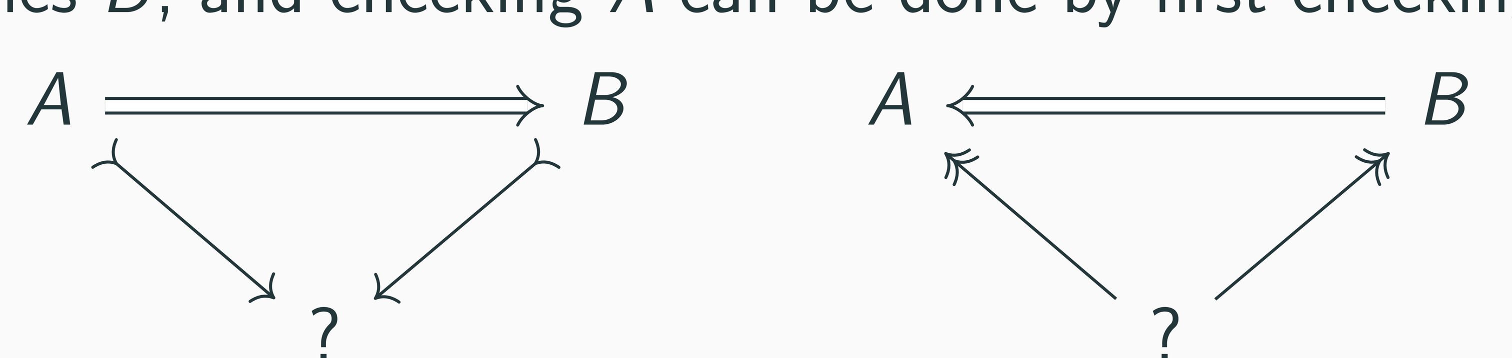
All casts in a gradually typed language can be specified by the equation

$$A \Rightarrow B \cong A \overset{s_A}{\rightarrowtail} ? \overset{r_B}{\twoheadrightarrow} B$$

Type precision can then be defined to mean the casts form a retraction:

$$A \sqsubseteq B \overset{\text{def}}{=} A \Rightarrow B \Rightarrow A \cong A \overset{id}{\to} A$$

Which is equivalent to saying that A's retraction from ? is a *subretraction* of B's, which intuitively means that A's contract is stronger than B's. That is, everything that satisfies A also satisfies B, and checking A can be done by first checking B:



Retractions and Blame

Wadler and Findler (09) gave a semantic definition of type precision in terms of *blame* in the explicit cast language, but they don't provide justification for the definition of blame. We turn this around to provide seamntic criteria for blame. They decompose precision into positive and negative subtyping:

$$A \leq^+ B \overset{\text{def}}{=} A \Rightarrow B \text{ never blames } A$$
$$B \leq^- A \overset{\text{def}}{=} B \Rightarrow A \text{ never blames } A$$

Then $A \sqsubseteq B \cong A \leq^+ B \wedge B \leq^- A$. We can instead interpret them constructively, $A \leq^+ B$ means $A \sqsubseteq B$ but denotes the section $A \Rightarrow B$, whereas $B \leq^- A$ denotes the retraction $B \Rightarrow A$. Then the positive subtyping rules admit both interpretations:

$$A \leq^+ ? \qquad \frac{A' \leq^- A \qquad B \leq^+ B'}{(A \to B) \leq^+ (A' \to B')}$$

The dual negative subtyping rules are admissible under both interpretations:

$$? \leq^- A \qquad \frac{A' \leq^+ A \qquad B \leq^- B'}{(A \to B) \leq^- (A' \to B')}$$

But WF09contains a rule not admissible by the retraction interpretation, for any ground type G, such as $\mathbb{N}, \mathbb{B}, (? \to ?)$,

$$G \leq^- ?$$

However, this is the result of an arbitrary choice in the definition of blame! As an example, take a language with two primitive printing functions, $\mathrm{print}_{\mathbb{B}}$, $\mathrm{print}_{\mathbb{N}}$, and consider a program that reduces to the following:

$$\top \overset{\texttt{true}}{\to} \mathbb{B} \rightarrowtail ? \twoheadrightarrow \mathbb{N} \overset{\mathrm{print}_{\mathbb{N}}}{\to} \bot$$

Who's to blame? In WF09, the term is always blamed, but the language doesn't know which choice was right, so we propose that both the term and the continuation at ? should be blamed! With this change to this system, the definitions coincide.