Subtyping in Type Theory: Coercion Contexts and Local Coercions

Z. Luo and F. Part
Dept of Computer Science
Royal Holloway, Univ of London

This talk

- Subsumptive v.s. coercive subtyping
  - Review and background
- Coercion contexts and local coercions
  - Subtyping in contexts/terms
  - Coherence

(work in progress)

I. Subsumptive v.s. Coercive Subtyping

- Two views of typing
  - Type assignment
    - Objects/types exist independently & types are assigned to objects (eg, λ-terms may reside in different types.)
    - ML-like programming languages (eg, $\lambda x.x : \alpha \rightarrow \alpha$)
  - Types as collections of canonical objects
    - Types/objects co-exist (objects do not without types!)
    - Eg, canonical nats: 0 & succ(n) of type N.
    - TTs in proof assistants (eg, Martin-Löf's TT)

Two views of types → Two views of subtyping

Type Assignment → Subsumptive Subtyping

TTs with Canonical Objects → Coercive Subtyping

Subsumptive Subtyping

- Subsumption

$$\frac{a : A \qquad A \leq B}{a : B}$$

- Widely employed in type assignment systems
- Incompatible with canonical objects
  - Canonicity fails (LSX 2012)
  - Subject reduction fails (Luo 1999)

(Russell-style universes are a special case.)

Coercive subtyping

- Global coercions
  - T ➔ T[C], coercive subtyping extension where C is a set of global coercions

    $\Gamma \vdash A <_c B : Type$ (eg, $x{:}N \vdash Vect(N,x) < List(N)$)
- Subtyping as abbreviations

$$\frac{f : B \to D \quad a : A \quad A <_c B}{f(a) : D} \qquad \frac{f : B \to D \quad a : A \quad A <_c B}{f(a) = f(c(a)) : D}$$

- Meta-theoretic properties:
  - Coherence ➔ conservativity (SL02, LSX12)
  - Preserves consistency, canonicity, SR, …

II. Coercion Contexts and Local Coercions

- Local subtyping/coercions
  - Coercion contexts (cf, Coq): x:C, …, $A <_c B$, … $\vdash$ …
    - Some subtyping relations only hold in certain theories. (eg, group ➔ carrier type of a group)
    - Certain "reference transfers" only make sense in some specific contexts. (eg, "ham sandwich" ➔ human being)
  - Local coercions in terms: <u>coercion</u> $A <_c B$ <u>in</u> t
    - Two different monoids in a ring (<u>coercion</u> Ring$<_{c1}$Monoid <u>in</u> … and <u>coercion</u> Ring$<_{c2}$Monoid <u>in</u> …)
    - Disambiguation of word meanings in NL semantics (eg, "bank" ➔ riverside/financial institution)

Rules to start with:

$$(CC_1) \quad \frac{\Gamma \vdash A : Type \quad \Gamma \vdash B : Type \quad \Gamma \vdash c : (A)B}{\Gamma,\ A <_c B\ \ valid}$$

$$(CC_2) \quad \frac{\Gamma,\ A <_c B,\ \Gamma'\ \ valid}{\Gamma,\ A <_c B,\ \Gamma' \vdash A <_c B : Type}$$

$$(LC_J) \quad \frac{\Gamma,\ A <_c B \vdash J}{\Gamma \vdash \mathbf{coercion}\ A <_c B\ \mathbf{in}\ J}$$

where J is $k : K$, $k = k' : K$, $K\ kind$, $K = K'$, $A <_c B : Type$ or $K <_c K'$.

- Note: these are the two sides of the same coin: Coercions are
  - introduced into contexts as assumptions, and
  - moved to the right of $\vdash$ to form local coercions.

  (cf, bounded quantification ∀X≤A.B (Cardelli & Wegner 85))
- But, this is not enough: we need **coherent** contexts!

Coherence

- Coherence: uniqueness of coercions
- With coercions in contexts, coherence becomes more tractable:
  - With global coercions [LSX12], coherence is a global notion (based on derivability of a subsystem of the extension); coherence-checking is undecidable.
  - For coercion contexts, graph-based coherence checking (as in Coq) can do a lot.

Rules

- Coherence checking whenever a new context is formed – only coherent contexts are valid.
- Context extension:

$$\frac{\Gamma \vdash A : \mathrm{Type} \quad \Gamma \vdash B : \mathrm{Type} \quad \Gamma \vdash c : (A)B \quad \Gamma, A <_c B \text{ coherent}}{\Gamma, A <_c B \text{ valid}}$$

- Substitutions: eg,

$$\frac{\Gamma, x{:}K, \Gamma' \text{ valid} \qquad \Gamma \vdash k : K \qquad \Gamma, [k/x]\Gamma' \text{ coherent}}{\Gamma, [k/x]\Gamma' \text{ valid}}$$

- Alternatively, one might check coherence only in the coercive application rule:

$$\frac{\Gamma \vdash f : (x{:}B)D \quad \Gamma \vdash a : A \quad \Gamma \vdash A <_c B \quad \Gamma \text{ coherent}}{\Gamma \vdash f(a) : [c(a)/x]D}$$

- But a caveat: this would allow incoherent contexts, although arguably more efficient.

Abbreviations and Simplifications

- Abbreviations: eg,

$$\begin{aligned} & \textbf{coercion } A <_c B \textbf{ in } (k : K) \\ \equiv \quad & (\textbf{coercion } A <_c B \textbf{ in } k) \; : \; (\textbf{coercion } A <_c B \textbf{ in } K) \end{aligned}$$

- Simplifications:

$$\frac{\Gamma,\ A <_c B\ \mathit{valid} \quad \Gamma \vdash J}{\Gamma \vdash J = \textbf{coercion } A <_c B \textbf{ in } J}$$

eg, $\textbf{coercion } A <_c B \textbf{ in } Type = Type.$

Conservativity

- Let $T_<$ be the extension of T with coercion contexts and local coercions.

  $T_<$ is conservative over T,

  ie, any T-judgement derivable in $T_<$ is derivable in T.

  (proof to be done)
- Note: conservativity can now be expressed straightforwardly (no need for a *-calculus as in the case of global coercions.)