Subtyping

Artur Jarocki

University of Wrocaw

April 2, 2014

Subsumption

In object-oriented languages it is possible to emulate object that has fewer methods with another object, if the latter supports entire protocol of the former. This notion is called **subsumption**.

Judgement $E \vdash A <: B$ asserts that A is a subtype of B in environment E.

Judgement $E \vdash A <: B$ asserts that A is a subtype of B in environment E.
We also add a type constant Top, that is a supertype of every type.

Reflexivity, transitivity, subsumption

(Sub Refl)

$$\frac{E \vdash A}{E \vdash A <: A}$$

Reflexivity, transitivity, subsumption

(Sub Refl)

$$\frac{E \vdash A}{E \vdash A <: A}$$

(Sub Trans)

$$\frac{E \vdash A <: B \qquad E \vdash B <: C}{E \vdash A <: C}$$

Reflexivity, transitivity, subsumption

(Sub Refl)

$$\frac{E \vdash A}{E \vdash A <: A}$$

(Sub Trans)

$$\frac{E \vdash A <: B \qquad E \vdash B <: C}{E \vdash A <: C}$$

(Val Subsumption)

$$\frac{E \vdash a : A \qquad E \vdash A <: B}{E \vdash a : B}$$

Top supertype

$$\text{(Type } Top\text{)}\quad \frac{E \vdash \diamond}{E \vdash Top}$$

Top supertype

$$\text{(Type } Top\text{)} \quad \frac{E \vdash \diamond}{E \vdash Top}$$

$$\text{(Sub } Top\text{)} \quad \frac{E \vdash A}{E \vdash A <: Top}$$

$\Delta_{<:\to}$

(Sub Arrow)

$$\frac{E \vdash A' <: A \qquad E \vdash B <: B'}{E \vdash A \to B <: A' \to B'}$$

$\Delta_{<:Ob}$

$$\frac{E \vdash B_i \;\; \forall i \in 1..n+m}{E \vdash [l_i : B_i{}^{i \in 1..n+m}] <: [l_i : B_i{}^{i \in 1..n}]} \quad \text{(Sub Object) } (l_i \text{ distinct})$$

Incomplete type

We denote $A\{\bullet\}$ for an **incomplete type** with zero or more subexpressions missing represented as holes $\{\bullet\}$. Then $A\{B\}$ is a type obtained from A by filling all the holes with B.

$A\{\bullet\}$ is *covariant* if for all B, B' $B <: B'$ implies $A\{B\} <: A\{B'\}$.

$A\{\bullet\}$ is *covariant* if for all B, B' $B <: B'$ implies $A\{B\} <: A\{B'\}$.
$A\{\bullet\}$ is *contravariant* if for all B, B' $B <: B'$ implies $A\{B'\} <: A\{B\}$.

$A\{\bullet\}$ is *covariant* if for all B, B' $B <: B'$ implies $A\{B\} <: A\{B'\}$.
$A\{\bullet\}$ is *contravariant* if for all B, B' $B <: B'$ implies $A\{B'\} <: A\{B\}$.
$A\{\bullet\}$ is *invariant* if it is neither covariant or contravariant.

$$Ob_{I<:} \quad \triangleq \quad Ob_I \cup \Delta_{<:} \cup \Delta_{<:Ob}$$

$$Ob_{I<:} \triangleq Ob_I \cup \Delta_{<:} \cup \Delta_{<:Ob}$$
$$F_{I<:} \triangleq F_I \cup \Delta_{<:} \cup \Delta_{<:\rightarrow}$$

$$
\begin{aligned}
Ob_{I<:} &\triangleq Ob_I \cup \Delta_{<:} \cup \Delta_{<:Ob} \\
F_{I<:} &\triangleq F_I \cup \Delta_{<:} \cup \Delta_{<:\rightarrow} \\
FOb_{I<:} &\triangleq FOb_I \cup \Delta_{<:} \cup \Delta_{<:\rightarrow} \cup \Delta_{<:Ob}
\end{aligned}
$$

Example

$$
\begin{aligned}
RomCell &\triangleq [get : Nat] \\
PromCell &\triangleq [get : Nat, set : Nat \rightarrow RomCell] \\
PrivateCell &\triangleq [contents : Nat, get : Nat, set : Nat \rightarrow RomCell]
\end{aligned}
$$

Example

$$
\begin{array}{lll}
RomCell & \triangleq & [get : Nat] \\
PromCell & \triangleq & [get : Nat, set : Nat \to RomCell] \\
PrivateCell & \triangleq & [contents : Nat, get : Nat, set : Nat \to RomCell]
\end{array}
$$

$$
\begin{array}{l}
myCell : PromCell \triangleq \\
\ [contents = 0, \\
\ get = \varsigma(s : PrivateCell)s.contents, \\
\ set = \varsigma(s : PrivateCell)\lambda(n : Nat)s.contents := n]
\end{array}
$$

Minimum types

By adding subsumption we have lost unique-types property of Ob_I. However, $Ob_{I<:}$ has a weaker property: every term has a minimum type (if it has any type at all).

$MinOb_{1<:}$ rules

(Val Min Object) (where $A \equiv [l_i : B_i{}^{i \in 1..n}]$)

$$\frac{E, x_i : A \vdash b_i : B_i' \qquad \varnothing \vdash B_i' <: B_i \qquad \forall i \in 1..n}{E \vdash [l_i = \varsigma(x_i : A) b_i{}^{i \in 1..n}] : A}$$

$MinOb_{1<:}$ rules

$$\frac{E, x_i : A \vdash b_i : B_i' \qquad \varnothing \vdash B_i' <: B_i \qquad \forall i \in 1..n}{E \vdash [l_i = \varsigma(x_i : A)b_i{}^{i \in 1..n}] : A} \quad \text{(Val Min Object) (where } A \equiv [l_i : B_i{}^{i \in 1..n}])$$

$$\frac{E \vdash a : A' \qquad \varnothing \vdash A' <: A \qquad E, x : A \vdash b : B_j' \qquad \varnothing \vdash B_j' <: B_j \; j \in 1..n}{E \vdash a.l_j \Leftarrow \varsigma(x : A)b : A} \quad \text{(Val Min Update) (where } A \equiv [l_i : B_i{}^{i \in 1..n}])$$

Lemma

$MinOb_{I<:}$ typings are $Ob_{I<:}$ typings

If $E \vdash a : A$ is derivable in $MinOb_{I<:}$, then it is also derivable in $Ob_{1<:}$.

Lemma

$MinOb_{I<:}$ typings are $Ob_{I<:}$ typings

If $E \vdash a : A$ is derivable in $MinOb_{I<:}$, then it is also derivable in $Ob_{1<:}$.

Lemma

$MinOb_{I<:}$ has unique types

If $E \vdash a : A$ and $E \vdash a : A'$ are derivable in $MinOb_{I<:}$, then $A \equiv A'$.

Lemma

$MinOb_{I<:}$ typings are $Ob_{I<:}$ typings

If $E \vdash a : A$ is derivable in $MinOb_{I<:}$, then it is also derivable in $Ob_{1<:}$.

Lemma

$MinOb_{I<:}$ has unique types

If $E \vdash a : A$ and $E \vdash a : A'$ are derivable in $MinOb_{I<:}$, then $A \equiv A'$.

Lemma

$MinOb_{I<:}$ has smaller types than $Ob_{I<:}$

If $E \vdash a : A$ is derivable in $Ob_{I<:}$, then $E \vdash a : A'$ is derivable in $MinOb_{I<:}$ for some A' such that $E \vdash A' <: A$ is derivable (in either system).

$Ob_{1<:}$ has minimum types

Theorem

In $Ob_{1<:}$, if $E \vdash a : A$ then there exists B such that $E \vdash a : B$ and, for any A', if $E \vdash a : A'$ then $E \vdash B <: A'$.

Lemma

Bound weakening

If $E, x : D, E' \vdash J$ *and* $E \vdash D' <: D$*, then* $E, x : D', E' \vdash J$.

Lemma

Substitution

If $E, x : D, E' \vdash J\{x\}$ *and* $E \vdash d : D$*, then* $E, E' \vdash J\{d\}$.

subject reduction theorem for $Ob_{I<:}$

Theorem

Let c be a closed term and v be a result, and assume $\vdash c \leadsto v$. If $\varnothing \vdash c : C$, then $\varnothing \vdash v : C$.

$\Delta_{=<:}$

(Eq Subsumption)

$$\frac{E \vdash a \leftrightarrow a' : A \qquad E \vdash A <: B}{E \vdash a \leftrightarrow a' : B}$$

(Eq Top)

$$\frac{E \vdash a : A \qquad E \vdash b : B}{E \vdash a \leftrightarrow b : Top}$$

$\Delta_{=<:Ob}$

(Eq Sub Object) (where $A \equiv [l_i : B_i{}^{i \in 1..n}], A' \equiv [l_i : B_i{}^{i \in 1..n+m}]$)

$$\frac{E, x_i : A \vdash b_i : B' \qquad \forall i \in 1..n \qquad E, x_j : A' \vdash b_j : B_j \qquad \forall j \in n+1..n+m}{E \vdash [l_i = \varsigma(x_i : A) b_i{}^{i \in 1..n} \leftrightarrow [l_i = \varsigma(x_i : A') b_i{}^{i \in 1..n+m}] : A}$$

$\Delta_{=<:Ob}$

(Eq Sub Object) (where $A \equiv [l_i : B_i{}^{i\in 1..n}], A' \equiv [l_i : B_i{}^{i \in 1..n+m}]$)

$$\frac{E, x_i : A \vdash b_i : B' \qquad \forall i \in 1..n \qquad E, x_j : A' \vdash b_j : B_j \qquad \forall j \in n+1..n+m}{E \vdash [l_i = \varsigma(x_i : A)b_i{}^{i\in 1..n} \leftrightarrow [l_i = \varsigma(x_i : A')b_i{}^{i\in 1..n+m}] : A}$$

(Eval Select) (where $A \equiv [l_i : B_i{}^{i\in 1..n}], a \equiv [l_i = \varsigma(x_i : A')b_i\{x_i\}^{i\in 1..n+m}]$

$$\frac{E \vdash a : A \qquad j \in 1..n}{E \vdash a.l_j \leftrightarrow b_j\{a\} : B_j}$$

$\Delta_{=<:Ob}$

(Eq Sub Object) (where $A \equiv [l_i : B_i{}^{i \in 1..n}], A' \equiv [l_i : B_i{}^{i \in 1..n+m}]$)

$$\frac{E, x_i : A \vdash b_i : B' \qquad \forall i \in 1..n \qquad E, x_j : A' \vdash b_j : B_j \qquad \forall j \in n+1..n+m}{E \vdash [l_i = \varsigma(x_i : A) b_i{}^{i \in 1..n} \leftrightarrow [l_i = \varsigma(x_i : A') b_i{}^{i \in 1..n+m}] : A}$$

(Eval Select) (where $A \equiv [l_i : B_i{}^{i \in 1..n}], a \equiv [l_i = \varsigma(x_i : A') b_i\{x_i\}^{i \in 1..n+m}]$

$$\frac{E \vdash a : A \qquad j \in 1..n}{E \vdash a.l_j \leftrightarrow b_j\{a\} : B_j}$$

(Eval Update) (where $A \equiv [l_i : B_i{}^{i \in 1..n}], a \equiv [l_i = \varsigma(x_i : A') b_i\{x_i\}^{i \in 1..n+m}]$)

$$\frac{E \vdash a : A \qquad E, x : A \vdash b : B_j \qquad j \in 1..n}{E \vdash a.l_j \Leftarrow \varsigma(x : A) b \leftrightarrow [l_j = \varsigma(x : A') b, l_i = \varsigma(x_i : A') b_i{}^{i \in (1..n+m)-j}] : A}$$

$$\begin{aligned} A &\triangleq [x : Nat, f : Nat] \\ b : A &\triangleq [x = 1, f = \varsigma(s : A)1] \\ c : A &\triangleq [x = 1, f = \varsigma(s : A)s.x] \end{aligned}$$

$$
\begin{aligned}
A &\triangleq [x : Nat, f : Nat] \\
b : A &\triangleq [x = 1, f = \varsigma(s : A) l] \\
c : A &\triangleq [x = 1, f = \varsigma(s : A) s.x]
\end{aligned}
$$

Terms b and c cannot be equal at type A, but using (Eq Sub Object) it is now at least possible to show that b and c are equal at type [x:Nat].

If $A \equiv [l_i : B_i{}^{i \in 1..n}]$ is an object type, then:

$$Class(A) \quad \triangleq \quad [new : A, l_i : A \rightarrow B_i{}^{i \in 1..n}]$$

is the type of classes generating objects of type A. They are of form:

$$[new = \varsigma(z : Class(A))[L_i = \varsigma(s : A)z.l_i(s)^{i \in 1..n}], l_i = \lambda(s : A)b_i{}^{i \in 1..n}]$$

We want to say a class of type $Class(A')$ that has more pre-methods (may) inherit from a class of type $Class(A)$ with fewer pre-methods.

We want to say a class of type $Class(A')$ that has more pre-methods (may) inherit from a class of type $Class(A)$ with fewer pre-methods.

However, inheritance is not simply related to subtyping of class types, since $A' <: A$ implies neither $Class(A') <: Class(A)$ nor $Class(A) <: Class(A')$, because A and A' occur invariantly in $Class(A)$ and $Class(A')$.

When A and A' are object types, we set:
$Class(A')$ may inherit from $Class(A)$ iff $A' <: A$

When A and A' are object types, we set:
$Class(A')$ may inherit from $Class(A)$ iff $A' <: A$

If $Class(A')$ may inherit from $Class(A)$, then they are of forms $A' \equiv [l_i B_i{}^{i \in 1..n+m}]$ and $A \equiv [l_i B_i{}^{i \in 1..n}]$. Thus $A \to B_i <: A' \to B_i$ for $i \in 1..n$, because of contravariance of function types. Therefore pre-methods of a class $c : Class(A)$ of type $A \to B_i$ by subsumption may be reused in assembling class $c' : Class(A')$.