PLP - Práctica 4: Subtipado

Zamboni, Gianfranco

5 de marzo de 2018

Reglas de subtipado

4.1. Ejercicio 1

1)

$$\frac{\{y\} \subseteq \{x,y,z\} \qquad y : Nat = y : Nat}{\{x : Nat,\ y : Nat,\ z : Nat\} <: \{y : Nat\}}\text{S-Rcd}$$

Esta demostración no es única porque existe la regla $S - Trans$ que nos permite probar la transitividad de los tipos, si hubiesemos decidido usarla para primero conseguir un supertipo $\{x : Nat, y : Nat\}$ del primer término y luego probar que ese supertipo es subtipo de $\{y : Nat\}$ entonces también hubiese sido una demostración válida.

2)

$$\frac{\emptyset \subseteq \{x,y,z\}}{\{x : Nat,\ y : Nat\} <: \{\}}\text{S-Rcd}$$

$$\frac{\dfrac{\{x\} \subseteq \{x,y\} \qquad x : Nat = x : Nat}{\{x : Nat,\ y : Nat\} <: \{x : Nat\}}\text{S-Rcd} \qquad \dfrac{\emptyset \subseteq \{x,y,z\}}{\{x : Nat\} <: \{\}}\text{S-Rcd}}{\{x : Nat,\ y : Nat\} <: \{\}}\text{S-Trans}$$

4.2. Ejercicio 2

1) Los registros tienen subtipos infinitos porque dado el tipo de un registro $\omega = \{l_i : \sigma_i\}_{i \in 1..n}$, entonces cualquier tipo de la forma $\omega' = \{l_i : \tau_i\}_{i \in 1..k}$ con $k \geq n$ tal que $\tau_i <: \sigma_i^{i \in 1..n}$ es subtipo del tipo de ω.

Top tiene como subtipo a los registros y los registros tienen infinitos subtipos, entonces Top tiene infinitos subtipos.

Por S-Arrow, los subtipos de una función $\sigma \to \tau$ son los tipos de la forma $\sigma' \to \tau'$ tal que $\sigma <: \sigma'$ y $\tau' <: \tau$. En particular si τ es de tipo registro, entonces τ tiene infinitos subtipos, por lo que $\sigma \to \tau$ tambien los tiene (son las funciones que devuelven registros).

2) Top no tiene supertipos.

Los registros tiene una cantidad finita de supertipos, siendo el máximo registro $\{\} <: Top$

Otra vez, hay casos en que las funciones tienen infinitos supertipos y es cuando toman como parámetro a un registro. Esto es porque la regla S-Arrow es contravariante respecto del tipo del párametro de la función, es decir, para que un tipo $\sigma \to \tau$ sea supertipo de $\sigma' \to \tau$ tiene que valer que $\sigma <: \sigma'$. Y si $sigma'$ es un registro, entonces tiene infinitos subtipos.

4.3. Ejercicio 3

1) $S = Top$

2) Si solo consideramos los tipos básicos $Bool$, Nat, Int, $Float$, entonces $S = Bool$, pero cuando empezamos a considerar registros o listas u otros tipos, entonces, los “mínimos” de cada tipo no están relacionados de ninguna forma, incluso, en el caso de los registros, ese mínimo nisiquiera existe.

3) Por S-Arrow, tenemos que $S_1 \to S_2 <: T_1 \to T_2$ si $T_1 <: S_1$ y $T_2 <: S_2$.
El primer caso es el punto **1)**, el segundo es el punto **2)**.

4) Es similar al anterior pero con los casos invertidos.

4.4. Ejercicio 4

1) $T <: S \underset{\text{S-Trans}}{\iff} T <: T \ \wedge\ T <: S \underset{\text{S-Arrow}}{\iff} S \to T <: T \to T$

2) Si $S = Bool$ y $T = Top$, entonces $\{x : Bool, y : Top\}$ tiene 26 supertipos y el tipo $Bool \to Top$ solo tiene como supertipo a Top, porque $Bool$ no tiene subtipos y Top no tiene supertipos.

3) Si $S = Top$ y $T = Top$, entonces $\{x : Top, y : Top\}$ tiene como supertipos a $\{x : Top\}$, $\{y : Top\}$, $\{\}$ y a Top y el tipo $Top \to Top$ tiene infinitos por el ejercicio 2.

4.5. Ejercicio 5

Subtipado en el contexto de tipado

4.6. Ejercicio 6

a)

$$\cfrac{\cfrac{\cfrac{\cfrac{y : Nat \in \Gamma'}{\Gamma' \mapsto y : Nat}\,\text{T-Var}}{\Gamma' = \Gamma, \{y : Nat\} \mapsto succ(y) : Nat}\,\text{T-Succ}}{\Gamma \mapsto \lambda y : Nat.succ(y) : Nat \to Nat}\,\text{T-Abs} \qquad \cfrac{x : Bool \in \Gamma}{\Gamma \mapsto x : Bool}\,\text{T-Var} \qquad \cfrac{}{Bool <: Nat}\,\text{S-BoolNat}}{\cfrac{\Gamma = \{x : Bool\} \mapsto (\lambda y : Nat.succ(y))\ x : Nat}{\emptyset \mapsto \lambda x : Bool.(\lambda y : Nat.succ(y))\ x : Bool \to Nat}\,\text{T-Abs}}\,\text{T-App}$$

b) $\Gamma = \{r : \{l_1 : Bool, l_2 : Float\}\}$

$$\cfrac{\textbf{(1)} \qquad \textbf{(2)} \qquad \cfrac{\{l_1, l_2\} \subseteq \{l_1, l_2, l_3\} \qquad l_1 \Rightarrow Bool = Bool \qquad \cfrac{}{l_2 \Rightarrow Int <: Float}\,\text{S-IntFloat}}{\{l_1 : Bool,\ l_2 : Int,\ l_3 : Float\} <: \{l_1 : Bool,\ l_2 : Float\}}\,\text{S-RCD}}{\emptyset \mapsto (\lambda r : \{l_1 : Bool,\ l_2 : Float\}.\text{if } r.l_1 \text{ then } r.l_2 \text{ else } 5,5)\ \{l_1 = \texttt{true},\ l_2 = -8,\ l_3 = 9,0\} : Float}\,\text{T-App}$$

(1)

$$\cfrac{\cfrac{r : \{l_1 : Bool,\ l_2 : Float\} \in \Gamma \qquad 1 \in \{1,2\}}{\Gamma \mapsto r.l_1 : Bool}\,\text{T-Proj} \qquad \cfrac{r : \{l_1 : Bool,\ l_2 : Float\} \in \Gamma \qquad 1 \in \{1,2\}}{\Gamma \mapsto r.l_2 : Float}\,\text{T-Proj} \qquad \Gamma \mapsto 5,5 : Float}{\Gamma \mapsto \text{if } r.l_1 \text{ then } r.l_2 \text{ else } 5,5 : Float}\,\text{T-If}$$

(2)

$$\cfrac{\Gamma \mapsto \texttt{true} : Bool \qquad \Gamma \mapsto -8 : Int \qquad \Gamma \mapsto 9,0 : Float}{\Gamma \mapsto \{l_1 = \texttt{true},\ l_2 = -8,\ l_3 = 9,0\} : \{l_1 : Bool,\ l_2 : Int,\ l_3 : Float\}}\,\text{T-RCD}$$

4.7. Ejercicio 7

Cálculo λ clásico Aplicamos el algoritmo de inferencia a $x\ x$

$x\ x$ (3)

x (1) $\quad$ x (2)

(1) $\mathbb{W}(x) = \{x : \sigma\} \triangleright x : \sigma$
(2) $\mathbb{W}(x) = \{x : \tau\} \triangleright x : \tau$
(3) $\mathbb{W}(x\ x) = S\{x : \sigma\} \cup S\{x : \tau\} \triangleright S(x\ x) : S\theta$

$$S = MGU(\{\sigma \doteq \tau,\ \sigma \doteq \tau \to \theta\}) \overset{4}{\underset{\tau/\sigma}{\rightsquigarrow}} \{\tau \doteq \tau \to \theta\} \overset{6}{\rightsquigarrow} \texttt{falla}$$

Cálculo λ subtipado :

$$\dfrac{\dfrac{x : \sigma \to \tau \in \Gamma}{\Gamma \triangleright x : \sigma \to \tau} \qquad \dfrac{x : \theta \in \Gamma}{\Gamma \triangleright x : \theta} \qquad \theta <: \sigma}{\Gamma \mapsto x\ x : \tau}\ \text{T-App}$$

Entonces debemos unificar $\sigma \to \tau \doteq \theta$ y $\sigma \doteq \theta$ de tal manera que se cumpla $\theta <: \sigma$. Si elegimos que $\sigma = \theta = Top$, entonces nos queda que x es de tipo $Top \to \tau$ y de tipo Top simultaneamente. Y se cumple que $Top <: Top$. Además, como x es una función, puede tomar cualquier cosa que sea subtipo de Top. En particular $Top \to \tau <: Top$, por lo que el jucio de tipado $\{x : Top \to \tau\} \mapsto x\ x : \tau$ es válido.

4.8. Ejercicio 8

a) Si valiese S-Arrow', la expresión $M = (\lambda x : Int.x)$ 0,5 es de tipo Int, ya que $(\lambda x : Int.x) : Int \to Int$ y puede ser remplazada por su expresión equivalente $\lambda x : Float.x) : Float \to Float$ porque $Float \to Float <: Int \to Int$ según S-Arrow'. Sin embargo, si evaluamos $M = (\lambda x : Float.x)$ 0,5 tenemos que $M = 0{,}5$ que no es un entero.

b) Si la regla fuese covariante tanto en el argumento como en el resultado, entonces en la expresión $M = (\lambda x : Float.x)$ 0,5 : $Float$ podemos remplazar $\lambda x : Float.x : Float \to Float$ por $\lambda x : Int.x : Int \to Int$ sin generar ningún error. Pero si intentamos evaluar $M = (\lambda x : Int.x)$ 0,5 no podremos hacerlo porque la abstracción esta esperando un entero y 0,5 no lo es.

4.9. Ejercicio 9

a) Si la regla de referencias es covariante, entonces $\tau <: \sigma$, si intentamos subtipar una referencia $M : Ref\ \sigma$ con $Ref\ \tau$, entonces:

$$let\ r = \text{ref } 3\ in\ r := 2{,}1;$$
$$!r$$

En este caso, definimos $r : Ref\ Int$ en el let, por lo que cuando derreferenciemos r esperaremos entero. En la asignación, queremos asignar un $Float$ a r, por lo que esperariamos que r fuese de tipo $Ref\ Float$ y como la regla de subtipado es covariante y $Int <: Float$, entonces la asignación se realiza sin ningún problema.

Ahora, cuando derreferenciemos r obtendremos 2,1 que es un $Float$, no un Int por lo que el programa falla.

b) Si la regla es contravariante, entonces en el programa:

$$let\ r = \text{ref } 2{,}1\ in\ !r$$

Definimos a r como una referencia de $Float$, sin embargo, como r es subtipable a $Ref\ Int$, podemos usar la derreferenciación de enteros para derreferenciarla, lo que provocaría el error en el programa.

4.10. Ejercicio 10

a) El término es tipable. Demostración:

$$\dfrac{\dfrac{\dfrac{c : Comp_{\{x:Int\}} \in \Gamma}{\Gamma \mapsto c : Comp_{\{x:Int\}}}\text{T-Var} \qquad \dfrac{(1)}{\Gamma \mapsto \{x = 1,\ y = 2\} : \{x : Int\}} \qquad \dfrac{}{\Gamma \mapsto \{x = 0\} : \{x : Int\}}\text{T-Rcd}}{\Gamma = \{c : Comp_{\{x:Int\}}\} \mapsto \text{mejorSegún}(c, \{x = 1,\ y = 2\}, \{x = 0\}) : Bool}\text{T-Comp}}{\emptyset \mapsto \lambda c : Comp_{\{x:Int\}}.\text{mejorSegún}(c, \{x = 1,\ y = 2\}, \{x = 0\}) : Comp_{\{x:Int\}} \to Bool}\text{T-Abs}$$

(1)
$$\dfrac{\dfrac{}{\Gamma \mapsto \{x = 1,\ y = 2\} : \{x : Int,\ y : Int\}}\text{T-Rcd} \qquad \dfrac{\{x\} \subseteq \{x, y\} \qquad \dfrac{}{Int <: Int}\text{S-Refl}}{\{x : Int,\ y : Int\} <: \{x : Int\}}\text{S-Rcd}}{\Gamma \mapsto \{x = 1,\ y = 2\} : \{x : Int\}}\text{T-Sub}$$

b)

$$\dfrac{\tau <: \sigma}{Comp_\sigma <: Comp_\tau}\text{S-Comp}$$

c)

$$\dfrac{\dfrac{\dfrac{\dfrac{x : Comp_{Nat} \in \Gamma'}{\Gamma' \mapsto x : Comp_{Nat}}\text{T-Var} \qquad \Gamma' \mapsto 3 : Nat \qquad \Gamma' \mapsto 4 : Nat}{\Gamma' = \Gamma, \{x : Comp_{Nat}\} \mapsto \text{mejorSegún}(x, 3, 4)) : Bool}\text{T-Comp}}{\Gamma \mapsto \lambda x : Comp_{Nat}.\text{mejorSegún}(x, 3, 4)) : Comp_{Nat} \to Bool}\text{T-Abs} \qquad \dfrac{c : Comp_{Float} \in \Gamma}{\Gamma \mapsto c : Comp_{Float}}\text{T-Var} \qquad \dfrac{\dfrac{}{Int <: Float}\text{S-IntFloat}}{Comp_{Float} <: Comp_{Int}}\text{S-Comp}}{\dfrac{\Gamma = \{c : Comp_{Float}\} \mapsto (\lambda x : Comp_{Nat}.\text{mejorSegún}(x, 3, 4))\ c : Bool}{\emptyset \mapsto \lambda c : Comp_{Float}.(\lambda x : Comp_{Nat}.\text{mejorSegún}(x, 3, 4))\ c : Comp_{Float} \to Bool}\text{T-Abs}}\text{T-App}$$

4.11. Ejercicio 11

1)

$$\frac{}{< A \times B > \to C <: A \to B \to C}\,(\text{S-Curry}) \qquad \frac{}{A \to B \to C \;\; <: \;\; < A \times B > \to C}\,(\text{S-Uncurry})$$

2)

$$\dfrac{\dfrac{\dfrac{\{x\} \subseteq \{x, y\} \qquad A <: A'}{\{x : A, y : A'\} <: \{x : A'\}}\,\text{S-Rcd} \quad \dfrac{B <: B' \qquad C <: C}{B' \to C <: B \to C}\,\text{S-Arrow}}{\{x : A'\} \to B' \to C <: \{x : A, y : A'\} \to B \to C}\,\text{S-Arrow} \qquad \dfrac{}{\{x : A, y : A'\} \to B \to C \;\; <: \;\; < \{x : A, y : A'\} \times B > \to C}\,\text{S-Unurry}}{\{x : A'\} \to (B' \to C) \;\; <: \;\; < \{x : A, y : A'\} \times B > \to C}\,\text{S-Trans}$$

3) Nada

4.12. Ejercicio 12

1)

$$\dfrac{\dfrac{}{\emptyset \mapsto Clarabelle : Vaca}\,\text{T-Clara} \qquad \dfrac{\dfrac{}{\emptyset \mapsto Clarabelle : Vaca}\,\text{T-Clara} \quad \dfrac{\textbf{(a)}}{Vaca <: AlimentoPara(Vaca)}}{\emptyset \mapsto Clarabelle : AlimentoPara(Vaca)}\,\text{T-Subs}}{\emptyset \mapsto Comer(Clarabelle, Clarabelle) : Vaca}\,\text{T-Comer}$$

(a) $$\dfrac{\dfrac{}{Vaca <: AlimentoPara(Leon)}\,\text{S-VacaLeon} \qquad \dfrac{Leon <: Vaca}{AlimentoPara(Leon) <: AlimentoPara(Vaca)}\,\text{S-Alim}}{Vaca <: AlimentoPara(Vaca)}\,\text{S-Trans}$$

4.13. Ejercicio 13

1)

$$\frac{\sigma <: \tau}{\times_1(\sigma) <: \times_1(\tau)}\text{(S-Proj1)} \qquad \frac{\sigma <: \tau}{\times_2(\sigma) <: \times_2(\tau)}\text{(S-Proj2)}$$

$$\frac{\sigma <: \sigma'}{< \sigma \times \tau > \quad <: \quad \times_1\,(\sigma')}\text{(S-TuplaProj1)}$$

$$\frac{\tau <: \tau'}{< \sigma \times \tau > \quad <: \quad \times_2\,(\tau')}\text{(S-TuplaProj2)}$$

Todas las reglas son covariantes, siempre que usemos un elemento del tipo proyección podremos remplazar su tipo por uno más especifico sin ningún problema. Con las reglas S-TuplaProj, consideramos a las tuplas como subtipos de las proyecciones, pues las tuplas tienen la operacion para proyectar cada una de sus coordenadas.

Además, no tendría sentido la inversa, ya que si quisieramos remplazar una tupla por una proyección y necesitamos hacer uso de las dos coordenadas de las tuplas, estariamos perdiendo información.

2)

$$\dfrac{\dfrac{(a)}{\Gamma \mapsto \lambda y : \times_2(Int).\pi_2(y) : \times_2(Int) \to Float} \qquad \dfrac{p :< Nat, Nat > \in \Gamma}{\Gamma \mapsto p :< Nat, Nat >}\text{T-Var} \qquad \dfrac{\dfrac{}{Nat <: Int}\text{S-NatInt}}{< Nat, Nat ><: \times_2(Int)}\text{S-TuplaProj2}}{\Gamma = \{p :< Nat, Nat >\} \mapsto (\lambda y : \times_2(Int).\pi_2(y))\ p : Float}\text{T-App}$$

(a)

$$\dfrac{\dfrac{\dfrac{\dfrac{y : \times_2(Int) \in \Gamma'}{\Gamma' \mapsto y : \times_2(Int)}\text{T-Var}}{\Gamma' = \Gamma, \{y : \times_2(Int)\} \mapsto \pi_2(y) : Int}\text{T-Proj1}}{\Gamma \mapsto \lambda y : \times_2(Int).\pi_2(y) : \times_2(Int) \to Int}\text{T-Abs} \qquad \dfrac{\dfrac{}{\times_2(Int) <: \times_2(Int)}\text{S-Refl} \qquad \dfrac{}{Int <: Float}\text{S-IntFloat}}{\times_2(Int) \to Int <: \times_2(Int) \to Float}\text{S-Arrow}}{\Gamma \mapsto \lambda y : \times_2(Int).\pi_2(y) : \times_2(Int) \to Float}\text{T-Subs}$$

4.14. Ejercicio 14

$$\frac{\sigma <: \tau}{det(\sigma) <: det(\tau)}\ \text{S-Det}$$

$$\dfrac{\dfrac{}{x : det(Nat) \mapsto 0 : det(Int)}\ {****} \qquad \dfrac{\dfrac{x : det(Nat) \in \{x : det(Nat)\}}{x : det(Nat) \mapsto x : det(Nat)}\ \text{T-Var} \qquad \dfrac{Nat <: Int}{det(Nat) <: det(Int)}\ \text{S-Det}}{x : det(Nat) \mapsto x : det(Int)} \qquad \dfrac{}{x : det(Nat) \mapsto True : Bool}\ \text{T-True}}{\dfrac{x : det(Nat) \mapsto \text{if True then x else 0: } det(Int)}{\{\} \mapsto \lambda x : det(Nat).\text{if True then x else 0: } Nat \to det(Int)}\ \text{T-Abs}}$$