6 Subtyping

In questa sezione trattiamo un'estensione fondamentale del sistema di tipi, che interagisce in modo non triviale con molte caratteristiche dei linguaggi: il subtyping, o polimorfismo per sottotipaggio. Poiché il subtyping è una delle caratteristiche essenziali dei linguaggi ad oggetti consideriamo ora il semplice linguaggio con record, dove già si maifestano alcune proprietà interessanti del subtyping.

$$\begin{array}{rrl} \textit{Termini } M & ::= & x \mid c \mid \mathsf{op}(M_i) \mid \mathsf{fn}\, x : T.M \mid MM \mid \{\ell_i = M_i\ ^{i\in 1..n}\} \mid M.\ell \\ \textit{Valori } v & ::= & n \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{fn}\, x : T.M \mid \{\ell_i = v_i\ ^{i\in 1..n}\} \\ \textit{Tipi } T & ::= & \mathsf{Nat} \mid \mathsf{Bool} \mid T \to T \mid \{\ell_i : T_i\ ^{i\in 1..n}\} \end{array}$$

Con le regole che abbiamo dato, il typing è piuttosto rigido: richiede che gli argomenti passati alle funzioni abbiano *esattamente* lo stesso tipo dei parametri formali. Con questa regola il termine

$$(\mathsf{fn}\, r{:}\{\ell{:}\mathsf{Nat}\}.r.\ell + 2)\ \{\ell = 0,\ \ell' = 1\}$$

non è ben tipato: il tipo dell'argomento è infatti $\{\ell{:}\mathsf{Nat},\ \ell'{:}\mathsf{Nat}\}$, mentre la funzione si aspetta un $\{\ell{:}\mathsf{Nat}\}$. D'altra parte, la funzione avrebbe comunque un comportamento corretto poiché richiede che il parametro sia un record con un campo ℓ, e non importa se ha anche altri campi. Inoltre, per sapere che la funzione non usa nessun altro campo di r è sufficiente guardare il suo tipo, senza dover ispezionare il suo corpo. Quindi *è sempre corretto* passare un argomento di tipo $\{\ell{:}\mathsf{Nat},\ \ell'{:}\mathsf{Nat}\}$ ad una funzione che si aspetta un tipo $\{\ell{:}\mathsf{Nat}\}$. Lo scopo del subtying è quello di accettare/tipare termini come il precedente.

La relazione di subtyping formalizza l'intuizione che alcuni tipi sono *più informativi* di altri: diciamo che S è sottotipo di T, scritto $S <: T$, per indicare che

- i valori di tipo S sono un sottoinsieme dei valori di tipo T (subset semantics);
- un termine di tipo S può essere usato correttamente in un contesto che si aspetta un termine di tipo T (Principio di sostituzione).

Questa intuizione viene formalizzata introducendo:

1. una *relazione di subtyping* tra tipi, cioè un sistema di regole per dire quando (i.e. derivare giudizi della forma) $S <: T$
2. una regola che lega la relazione di typing e quella di subtyping facendo davvero in modo che un termine del sottotipo si possa usare dove ci si aspetta un termine del sopratipo:

$$\frac{\Gamma \vdash M : S \quad S <: T}{\Gamma \vdash M : T}\ (\textsc{Subsumption})$$

Tornando all'esempio, se definiamo la relazione di subtyping in modo che $\{\ell{:}\mathsf{Nat},\ \ell'{:}\mathsf{Nat}\} <: \{\ell{:}\mathsf{Nat}\}$, allora il giudizio $\Gamma \vdash \{\ell = 0,\ \ell' = 1\} : \{\ell{:}\mathsf{Nat}\}$ è derivabile grazie alla regola (SUBSUMPTION), e quindi il termine $(\mathsf{fn}\, r{:}\{\ell{:}\mathsf{Nat}\}.r.\ell)\ \{\ell = 0,\ \ell' = 1\}$ risulta ben tipato.

6.1 La relazione di sottotipo

La relazione di sottotipo è formalizzata tramite un insieme di regole di inferenza che derivano giudizi della forma $S <: T$. Oltre alle regole specifiche per i vari costruttori di tipo, questo insieme contiene due regole che definiscono la riflessività e la transitività della relazione di subtyping, in accordo con l'intuizione data dal principio di sostituzione.

$$\frac{}{T <: T}\ (\textsc{Reflex}) \qquad \frac{S <: U \quad U <: T}{S <: T}\ (\textsc{Trans})$$

Il subtyping per i tipi record Per i tipi record, abbiamo già osservato che è corretto utilizzare un termine con più campi al posto di un termine con meno campi. Ciò suggerisce la seguente regola di *subtyping in larghezza*:

$$\text{(SUB WIDTH)}\quad \frac{}{\{\ell_i : T_i \ ^{i\in 1...n+k}\} <: \{\ell_i : T_i \ ^{i\in 1...n}\}}$$

In altre parole, il tipo $\{\ell : \mathsf{Nat}\}$ è il tipo di tutti i record con *almeno* un campo ℓ di tipo Nat. Quindi il tipo con più campi descrive un vincolo più rigido sui suoi valori.

Esiste inoltre un altro caso in cui è corretto il subtyping, descritto dalla seguente regola di *subtyping in profondità*, che permette anche ai tipi dei singoli campi di cambiare:

$$\text{(SUB DEPTH)}\quad \frac{S_i <: T_i \quad \forall i \in 1...n}{\{\ell_i : S_i \ ^{i\in 1...n}\} <: \{\ell_i : T_i \ ^{i\in 1...n}\}}$$

Nota che la regola (SUB DEPTH) abbinata alla riflessività del subtyping permette di raffinare il tipo di un singolo campo invece che di tutti.

EXERCISE 6.1. Scrivere le derivazioni dei giudizi

- $\{\ell : \{\mathsf{a{:}Nat}, \mathsf{b{:}Nat}\},\ \ell' : \{\mathsf{m{:}Nat}\}\} <: \{\ell : \{\mathsf{a{:}Nat}\},\ \ell' : \{\ \}\}$
- $\{\ell : \{\mathsf{a{:}Nat}, \mathsf{b{:}Nat}\},\ \ell' : \{\mathsf{m{:}Nat}\}\} <: \{\ell : \{\mathsf{a{:}Nat}\},\ \ell' : \{\ \mathsf{m{:}Nat}\}\}$
- $\{\ell : \{\mathsf{a{:}Nat}, \mathsf{b{:}Nat}\},\ \ell' : \{\mathsf{m{:}Nat}\}\} <: \{\ell : \{\mathsf{a{:}Nat}\}\}$

□

Si noti che con le regole date fino a questo momento, il giudizio $\{\mathsf{a{:}Nat}, \mathsf{b{:}Nat}\} <: \{\ \mathsf{b{:}Nat}\}$ non è derivabile. Affinchè si possa derivare, dobbiamo aggiungere la regola seguente:

$$\text{(PERMUTE)}\quad \frac{\{k_j : S_j \ ^{j\in 1...n}\} \text{ è una permutazione di } \{\ell_i : T_i \ ^{i\in 1...n}\}}{\{k_j : S_j \ ^{j\in 1...n}\} <: \{\ell_i : T_i \ ^{i\in 1...n}\}}$$

Con questa regola $\{\mathsf{a{:}Nat}, \mathsf{b{:}Bool}, \mathsf{c{:}Nat}\}$ è sottotipo di $\{\mathsf{b{:}Bool}, \mathsf{c{:}Nat}, \mathsf{a{:}Nat}\}$ e viceversa, quindi la relazione di subtyping non è antisimmetrica (è solo un preordine). Con l'aggiunta di questa regola si può quindi ottenere un supertipo eliminando da un record un campo qualsiasi e non solo l'ultimo.

EXERCISE 6.2. Si scriva la derivazione di $\{\mathsf{a{:}Nat}, \mathsf{b{:}Bool}, \mathsf{c{:}Nat}\} <: \{\mathsf{b{:}Bool}\}$. □

Le tre regole di subtyping per i tipi record si possono riassumere in una singola:

$$\text{(SUB)}\quad \frac{\{\ell_i \ ^{i\in 1..n}\} \subseteq \{k_j \ ^{j\in 1..m}\} \quad k_j = \ell_i \text{ implica } S_j <: T_i}{\{k_j : S_j \ ^{j\in 1..m}\} <: \{\ell_i : T_i \ ^{i\in 1..n}\}}$$

Vale la pena però tenere separate le regole di subtyping anche perché esistono linguaggi reali che non le implementano tutte e tre. Ad esempio in Java una classe e una sua permutazione (con nome diverso ma stessi membri) non sono sottotipi. In Java inoltre non c'e' subtyping in profondità in quanto una sottoclasse non può modificare i tipi dei campi o i tipi degli argomenti dei metodi (anche se l'overriding ammette la covarianza del tipo di ritorno).

Il subtyping per i tipi funzione Nel nostro linguaggio higher-order anche le funzioni possono essere argomenti di altre funzioni, va quindi definita una regola di subtyping anche per i tipi funzione, una regola

cioè che specifichi quando una funzione di un certo tipo può essere correttamente usata al posto di una funzione di un altro tipo.

$$\frac{T_1 <: S_1 \quad S_2 <: T_2}{S_1 \to S_2 \ <: \ T_1 \to T_2} \ (\textsc{Arrow})$$

La regola dice che il tipo funzione è *controvariante* nel tipo degli argomenti e *covariante* nel tipo del valore di ritorno. Intuitivamente, se abbiamo una funzione f di tipo $S_1 \to S_2$, allora sappiamo che f accetta elementi di tipo S_1; chiaramente f accetterà anche elementi di ogni sottotipo T_1 si S_1. Il tipo di f ci dice anche che restituisce elementi di tipo S_2; possiamo pensare questi risultati come elementi di ogni supertipo T_2 di S_2. Cioè ogni funzione f di tipo $S_1 \to S_2$ può essere vista come una funzione di tipo $T_1 \to T_2$.

EXERCISE 6.3. Dare la derivazione del giudizio $\varnothing \vdash (\mathsf{fn}\, r{:}\{\ell{:}\mathsf{Nat}\}.r.\ell + 2)\ \{\ell = 0,\ \ell' = 1\} : \mathsf{Nat}$. Esiste una sola derivazione di questo giudizio? □

EXERCISE 6.4. Quale potrebbbe essere la relazione di sottotipo dei variant types? (pag 196 libro B.Pierce) □

6.2 Proprietà del sistema di tipi con subtyping

Avendo aggiunto al sistema di tipi la relazione di subtyping, bisogna controllare che valgano ancora i teoremi di correttezza.

EXERCISE 6.5. Quali proprietà del sistema di tipi si perderebbero se avessimo definito la relazione di subtyping con una regola di troppo? E se l'avessimo definita con una regola in meno? Se avessimo definito le regole in modo tale che $\{\ell : \mathsf{Nat}\} <: \{\ell : \mathsf{Nat},\ \ell' : \mathsf{Nat}\}$, quale proprietà del sistema non sarebbe più vera? Identificarla e darne un controesempio. □

Le dimostrazioni dei due teoremi principali sono più complesse, perché dato un giudizio di tipo, la sua derivazione non è più univocamente guidata dalla sintassi, poiché una o più applicazioni della regola di subsumption possono apparire qua e là, anche nell'ultimo passo.

Lemma 6.6 (Lemmi di inversione)**.**

1. *Se* $\Gamma \vdash x : T$ *allora* $\exists S$ *tale che* $x : S \in \Gamma$ *e* $S <: T$.
2. *Se* $\Gamma \vdash \mathsf{fn}\, x{:}S_1.M : T$ *è derivabile, allora* $\exists T_1, T_2$ *tali che* $T = T_1 \to T_2$*, con* $T_1 <: S_1$ *e* $\Gamma, x : S_1 \vdash M : T_2$.
3. *Se* $\Gamma \vdash \{k_r = M_r\ ^{r\in 1\ldots m}\} : T$ *è derivabile, allora* $\exists T_1$*,* $i = 1, .., n$ *tali che* $T = \{\ell_i : T_i\ ^{i\in 1..n}\}$*, con* $\{\ell_i\ ^{i\in 1..n}\} \subseteq \{k_r\ ^{r\in 1..m}\}$ *e* $\Gamma \vdash M_r : T_i$ *per ogni etichetta comune* $k_r = \ell_i$.

□

Lemma 6.7 (Lemma di inversione della relazione di subtyping)**.**

1. *Se* $S <: T_1 \to T_2$ *allora* S *è della forma* $S_1 \to S_2$ *con* $T_1 <: S_1$ *e* $S_2 <: T_2$.
2. *Se* $S <: \{\ell_i : T_i\ ^{i\in 1..n}\}$ *allora* S *è della forma* $\{k_j : S_j\ ^{j\in 1..m}\}$ *tale che* $\{\ell_i\ ^{i\in 1..n}\} \subseteq \{k_j\ ^{j\in 1..m}\}$ *e* $S_j <: T_i$ *per ogni etichetta comune* $\ell_i = k_j$.

□

Il lemma di sostituzione e il teorema di subject reduction hanno lo stesso enunciato di prima, ma la loro dimostrazione cambia.

Lemma 6.8 (Substitution)**.** *Se* $\Gamma, x : S \vdash M : T$ *e* $\Gamma \vdash N : S$*, allora* $\Gamma \vdash M\{x := N\} : T$. □

Theorem 6.9 (Preservazione dei tipi)**.** *Se* $\Gamma \vdash M : T$ *e* $M \longrightarrow M'$*, allora* $\Gamma \vdash M' : T$. □

Anche il teorema di progressione non cambia, ma si basa sul lemma delle forme canoniche, la cui dimostrazione è complicata dal subtyping.

Lemma 6.10 (Forme canoniche)**.**

- *Se v è un valore di tipo $T_1 \to T_2$ allora v è della forma* $\mathsf{fn}\, x{:}S_1.M$.
- *Se v è un valore di tipo $\{\ell_i : T_i\ ^{i\in 1..n}\}$, allora v è della forma $\{k_j = v_j\ ^{j\in 1..m}\}$ tale che $\{\ell_i\ ^{i\in 1..n}\} \subseteq \{k_j\ ^{j\in 1..m}\}$*

□

Theorem 6.11 (Progressione)**.** *Sia M un termine chiuso e ben tipato, i.e. $\emptyset \vdash M : T$, allora o M è un valore, oppure esiste un termine M' tale che $M \longrightarrow M'$.* □