Intersection types

In this section, we will work with both the simple types introduced earlier (definition given again below), as well as intersection types, defined in the following way:

Definition (Types) - Note that o and φ are constants. ω is used to denote an empty list of strict intersection types and $\bigcap \tau \equiv [\tau]$.

i) Simple types:

$$\sigma ::= \mathsf{o} \mid \sigma \to \sigma$$

ii) Intersection types:

$$\mathcal{T}_s ::= \varphi \mid \mathcal{T} \rightsquigarrow \mathcal{T}$$

$$\mathcal{T} ::= \mathsf{List}\ \mathcal{T}_s$$

The reason why $\mathcal{T}$ is defined as a list of strict types $\mathcal{T}_s$ is due to the requirement that the types in $\mathcal{T}$ be finite. The decision to use lists was taken because the Agda standard library includes a definition of lists along with definitions of list membership $\in$ for lists and other associated lemmas.

Next, we redefine the λ-terms slightly, by annotating the terms with simple types. The reason for this will be clear later on.

Definition (Terms) - Let σ range over simple types in the following definition:

i) Simply-typed terms:

$$M ::= x_\sigma \mid MM \mid \lambda x_\sigma.M \mid Y_\sigma \text{ where } x \in Var$$

ii) Simply-typed pre-terms:

$$M' ::= x_\sigma \mid i \mid M'M' \mid \lambda_\sigma.M' \mid Y_\sigma \text{ where } x \in Var \text{ and } i \in \mathbb{N}$$

Note that both definitions implicitly assume that in the case of application, a well formed simply-typed term will be of the form st, where s has some simple type $A \to B$ and t is typed with the simple type A. Sometimes the same subscript notation will be used to indicate the simple type of a given pre-/term, for example: $m_{A\to B}$.
The typed versions of substitution and the open and close operations are virtually identical to the untyped versions.

Type refinement

Next, we introduce the notion of type refinement by defining the refinement relation ::, between simple types and intersection types.

Definition (::) - Since intersection types are defined in terms of strict ($\mathcal{T}_s$) and non-strict ($\mathcal{T}$) intersection types, for correct typing, the definition of :: is split into two versions, one for strict and another for non-strict types. In the definition below, τ ranges over strict intersection types $\mathcal{T}_s$, with τ_i, τ_j ranging over non-strict intersection types $\mathcal{T}$, and A, B range over simple types σ:

$$(base)\ \frac{}{\varphi ::_s \mathsf{o}} \qquad (arr)\ \frac{\tau_i :: A \qquad \tau_j :: B}{\tau_i \rightsquigarrow \tau_j ::_s A \to B}$$

$$(nil)\ \frac{}{\omega :: A} \qquad (cons)\ \frac{\tau ::_s A \qquad \tau_i :: A}{(\tau,\ \tau_i) :: A}$$

Having a notion of refinement, we define a restricted version of a subset relation on intersection types, which is defined only for pairs of intersection types, which refine the same simple type.

Definition ($\subseteq^A$) - In the definition below, τ, τ' range over $\mathcal{T}_s$, $\tau_i, \ldots, \tau_n$ range over $\mathcal{T}$ and A, B range over σ:

$$(base)\ \frac{}{\varphi \subseteq_s^\circ \varphi} \qquad (arr)\ \frac{\tau_i \subseteq^A \tau_j \qquad \tau_m \subseteq^B \tau_n}{\tau_j \rightsquigarrow \tau_m \subseteq_s^{A\to B} \tau_i \rightsquigarrow \tau_n}$$

$$(nil)\ \frac{\tau_i :: A}{\omega \subseteq^A \tau_i} \qquad (cons)\ \frac{\exists \tau' \in \tau_j.\ \tau \subseteq_s^A \tau' \qquad \tau_i \subseteq^A \tau_j}{(\tau,\ \tau_i) \subseteq^A \tau_j}$$

$$(\rightsquigarrow \cap)\ \frac{(\tau_i \rightsquigarrow (\tau_j +\!\!+ \tau_k),\ \tau_m) :: A \to B}{(\tau_i \rightsquigarrow (\tau_j +\!\!+ \tau_k),\ \tau_m) \subseteq^{A\to B} (\tau_i \rightsquigarrow \tau_j,\ \tau_i \rightsquigarrow \tau_k,\ \tau_m)} \qquad (trans)\ \frac{\tau_i \subseteq^A \tau_j \qquad \tau_j \subseteq^A \tau_k}{\tau_i \subseteq^A \tau_k}$$

It's easy to show the following properties hold for the $\subseteq^A$ and :: relations:

Lemma ($\subseteq \implies ::$)

i) $\tau \subseteq_s^A \delta \implies \tau ::_s A \wedge \delta ::_s A$
ii) $\tau_i \subseteq^A \delta_i \implies \tau_i :: A \wedge \delta_i :: A$

Proof: By **?mutual?** induction on the relations $\subseteq_s^A$ and $\subseteq^A$.

Lemma ($\subseteq$-*refl*)

i) $\tau ::_s A \implies \tau \subseteq_s^A \tau$
ii) $\tau_i :: A \implies \tau_i \subseteq^A \tau_i$

Proof: By induction on τ and τ_i.

Intersection type assignment

Having annotated the λ-terms with simple types, the following type assignment only permits the typing of simply-typed λ-terms with an intersection type, which refines the simple type of the λ-term:

Definition (Intersection-type assignment)

$$(var)\ \frac{\exists (x, \tau_i, A) \in \Gamma.\ \bigcap \tau \subseteq^A \tau_i}{\Gamma \Vdash_s x_A : \tau} \qquad (app)\ \frac{\Gamma \Vdash_s s : \tau_i \rightsquigarrow \tau_j \qquad \Gamma \Vdash t : \tau_i}{\Gamma \Vdash_s st : \tau}\ (\bigcap \tau \subseteq^B \tau_j)$$

$$(abs)\ \frac{\forall x \notin L.\ (x, \tau_i, A), \Gamma \Vdash t_B^x : \tau_j}{\Gamma \Vdash_s \lambda_A.t : \tau_i \rightsquigarrow \tau_j} \qquad (Y)\ \frac{\exists \tau_x.\ \bigcap \tau_x \rightsquigarrow \tau_x \subseteq^{A\to A} \tau_i \wedge \tau_j \subseteq^A \tau_x}{\Gamma \Vdash_s Y_A : \tau_i \rightsquigarrow \tau_j}$$

$$(\rightsquigarrow \cap)\ \frac{\Gamma \Vdash_s m_{A\to B} : \tau_i \rightsquigarrow \tau_j \qquad \Gamma \Vdash_s m_{A\to B} : \tau_i \rightsquigarrow \tau_k}{\Gamma \Vdash_s m_{A\to B} : \tau_i \rightsquigarrow \tau_x}\ (\tau_j +\!\!+ \tau_k \subseteq^B \tau_x)$$

$$(nil)\ \frac{\tau_i :: A}{\omega \subseteq^A \tau_i} \qquad (cons)\ \frac{\exists \tau' \in \tau_j.\ \tau \subseteq_s^A \tau' \qquad \bigcap \tau_i \subseteq^A \tau_j}{(\tau,\ \tau_i) \subseteq^A \tau_j}$$