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. The following sugar notation will also occasionally be used: $\bigcap \tau \equiv [\tau]$ and $\tau \cap \tau' \equiv \bigcap \tau +\!\!+ \bigcap \tau' \equiv [\tau, \tau']$.

i) Simple types:

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

ii) Intersection types:

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

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

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: $m_{A\to B}$. Also, rather confusingly, the simple type of Y_A is $(A \to A) \to A$, and thus Y_A should not be confused with a constant Y having a simple type A. **Maybe use something like this instead?:** $m_{:A\to B}$ i.e. $Y_{A:(A\to A)\to A}$.
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:

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

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

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 $\mathcal{T}_s$, $\tau_i, \dots, \tau_n$ range over $\mathcal{T}$ and A, B range over σ:

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

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

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

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 ($\subseteq$ admissible) The following rules are admissible in $\subseteq_s^A$ / $\subseteq^A$:

i) $(refl_s)\ \dfrac{\tau ::_s A}{\tau \subseteq_s^A \tau} \qquad (refl)\ \dfrac{\tau_i :: A}{\tau_i \subseteq^A \tau_i} \qquad (trans_s)\ \dfrac{\tau \subseteq_s^A \tau' \qquad \tau' \subseteq_s^A \tau''}{\tau \subseteq_s^A \tau''} \qquad (\subseteq)\ \dfrac{\tau_i \subseteq \tau_j}{\tau_i \subseteq^A \tau_j}\ (\tau_j :: A)$

ii) $(\mathbin{+\!\!+}_L)\ \dfrac{\tau_i :: A \qquad \tau_j \subseteq^A \tau_{j'}}{\tau_i \mathbin{+\!\!+} \tau_j \subseteq^A \tau_i \mathbin{+\!\!+} \tau_{j'}} \qquad (\mathbin{+\!\!+}_R)\ \dfrac{\tau_i \subseteq^A \tau_{i'} \qquad \tau_j :: A}{\tau_i \mathbin{+\!\!+} \tau_j \subseteq^A \tau_{i'} \mathbin{+\!\!+} \tau_j} \qquad (glb)\ \dfrac{\tau_i \subseteq^A \tau_k \qquad \tau_j \subseteq^A \tau_k}{\tau_i \mathbin{+\!\!+} \tau_j \subseteq^A \tau_k}$

iii) $(mon)\ \dfrac{\tau_i \subseteq^A \tau_j \qquad \tau_{i'} \subseteq^A \tau_{j'}}{\tau_i \mathbin{+\!\!+} \tau_{i'} \subseteq^A \tau_j \mathbin{+\!\!+} \tau_{j'}}$

iv) $(\rightsquigarrow\cap')\ \dfrac{\tau_i :: A \qquad \tau_j :: A}{\bigcap((\tau_i \mathbin{+\!\!+} \tau_j) \rightsquigarrow (\tau_i \mathbin{+\!\!+} \tau_j)) \subseteq^{A \to B} \tau_i \rightsquigarrow \tau_i \cap \tau_j \rightsquigarrow \tau_j}$

Proof:

i) By induction on τ and τ_i.

ii) By induction on $\tau_i \subseteq^A \tau_{i'}$.

iii)
$$(glb)\ \dfrac{(trans)\ \dfrac{\tau_i \subseteq^A \tau_j \qquad (\subseteq)\ \dfrac{\overline{\tau_j \subseteq \tau_j \mathbin{+\!\!+} \tau_{j'}}}{\tau_j \subseteq^A \tau_j \mathbin{+\!\!+} \tau_{j'}}}{\tau_i \subseteq^A \tau_j \mathbin{+\!\!+} \tau_{j'}} \qquad (trans)\ \dfrac{\tau_{i'} \subseteq^A \tau_{j'} \qquad (\subseteq)\ \dfrac{\overline{\tau_{j'} \subseteq \tau_j \mathbin{+\!\!+} \tau_{j'}}}{\tau_{j'} \subseteq^A \tau_j \mathbin{+\!\!+} \tau_{j'}}}{\tau_{i'} \subseteq^A \tau_j \mathbin{+\!\!+} \tau_{j'}}}{\tau_i \mathbin{+\!\!+} \tau_{i'} \subseteq^A \tau_j \mathbin{+\!\!+} \tau_{j'}}$$

iv) Follows from $(\rightsquigarrow\cap)$, $(cons)$ and $(trans)$.

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)

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

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

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

$$(nil)\ \frac{}{\Gamma\Vdash m:\omega}\qquad (cons)\ \frac{\Gamma\Vdash_s m:\tau\qquad \Gamma\Vdash m:\tau_i}{\Gamma\Vdash m:\tau,\tau_i}$$

In the definition above, Γ is the typing context, consisting of triples of the variable name and the corresponding intersection and simple types. Γ is defined as a list of these triples in the Agda implementation. It is assumed in the typing system, that Γ is well-formed. Formally, this can be expressed in the following way:

Definition (Well-formed intersection-type context)

$$(nil)\ \frac{}{\mathsf{Wf\text{-}ICtxt}\ [\ ]}\qquad (cons)\ \frac{x\notin\mathsf{dom}\ \Gamma\qquad \tau_i::A\qquad \mathsf{Wf\text{-}ICtxt}\ \Gamma}{\mathsf{Wf\text{-}ICtxt}\ (x,\tau_i,A),\Gamma}$$

Subtyping

In the typing system, the rules (Y) and $(\rightsquigarrow\cap)$ are defined in a slightly more complicated way than might be necessary. For example, one might assume, the (Y) rule could simply be:

$$(Y)\ \frac{}{\Gamma\Vdash_s Y_A:\bigcap(\tau_x\rightsquigarrow\tau_x)\rightsquigarrow\tau_x}$$

The reason why the more complicated forms of both rules were introduced was purely an engineering one, namely to make the proof of sub-typing/weakening possible, as the sub-typing rule is required in multiple further proofs:

Lemma (Sub-typing) The following rule(s) are admissible in $\Vdash_s/\Vdash$:

$$(\supseteq_s)\ \frac{\Gamma\Vdash_s m_A:\tau}{\Gamma'\Vdash_s m_A:\tau'}\ (\Gamma'\subseteq_\Gamma\Gamma,\tau\supseteq^A_s\tau')\qquad (\supseteq)\ \frac{\Gamma\Vdash m_A:\tau_i}{\Gamma'\Vdash m_A:\tau_j}\ (\Gamma'\subseteq_\Gamma\Gamma,\tau_i\supseteq^A_s\tau_j)$$

Proof: Ommited.

The relation $\Gamma\subseteq_\Gamma\Gamma'$ is defined for any well-formed contexts Γ,Γ', where for each triple $(x,\tau_i,A)\in\Gamma$, there is a corresponding triple $(x,\tau_j,A)\in\Gamma'$ s.t. $\tau_i\subseteq^A\tau_j$.

Inversion lemmas

The shape of the derivation tree is not always unique for arbitrary typed term $\Gamma\Vdash_s m:\tau$. For example, given a typed term $\Gamma\Vdash_s\lambda_A.m:\tau_i\rightsquigarrow\tau_j$, either of the following two derivation trees, could be valid:

$$(abs)\ \frac{\dfrac{\vdots}{\forall x\notin L.\ (x,\tau_i,A),\Gamma\Vdash m^x:\tau_j}}{\Gamma\Vdash_s\lambda_A.m:\tau_i\rightsquigarrow\tau_j}$$

$$(\rightsquigarrow\cap)\ \dfrac{\dfrac{\vdots}{\Gamma \Vdash_s \lambda_A.m_B : \tau_i \rightsquigarrow \tau_p} \qquad \dfrac{\vdots}{\Gamma \Vdash_s \lambda_A.m_B : \tau_i \rightsquigarrow \tau_q}}{\Gamma \Vdash_s \lambda_A.m_B : \tau_i \rightsquigarrow \tau_j}\ (\tau_j \subseteq^B \tau_p +\!\!+ \tau_q)$$

However, it is obvious that the second tree will always necessarily have to have an application of (*abs*) in all its branches. Because it will be necessary to reason about the shape of the typing derivation trees, it is useful to prove the following inversion lemmas:

Lemma (Y-inv, *abs*-inv)

i) $\Gamma \Vdash_s Y_A : \tau_i \rightsquigarrow \tau_j \implies \exists \tau_x.\ \bigcap(\tau_x \rightsquigarrow \tau_x) \subseteq^{A\to A} \tau_i \wedge \tau_j \subseteq^A \tau_x$
ii) $\Gamma \Vdash_s \lambda_A.m : \tau_i \rightsquigarrow \tau_j \implies \exists L.\ \forall x \notin L.\ (x, \tau_i, A), \Gamma \Vdash m^x : \tau_j$

Proof:

i) There are two cases to consider, one, where the last rule in the derivation tree of $\Gamma \Vdash_s Y_A : \tau_i \rightsquigarrow \tau_j$ was (Y). Otherwise, the last rule was $(\rightsquigarrow\cap)$:

   (Y): Follows immediately.
   $(\rightsquigarrow\cap)$: We must have a derivation tree of the shape:

$$(\rightsquigarrow\cap)\ \dfrac{\dfrac{\vdots}{\Gamma \Vdash_s Y_A : \tau_i \rightsquigarrow \tau_p} \qquad \dfrac{\vdots}{\Gamma \Vdash_s Y_A : \tau_i \rightsquigarrow \tau_q}}{\Gamma \Vdash_s Y_A : \tau_i \rightsquigarrow \tau_j}\ (\tau_j \subseteq^B \tau_p +\!\!+ \tau_q)$$

   Then by IH, we have:

   - $\exists \tau_{xp}.\ \bigcap(\tau_{xp} \rightsquigarrow \tau_{xp}) \subseteq^{A\to A} \tau_i \wedge \tau_p \subseteq^A \tau_{xp}$ and
   - $\exists \tau_{xq}.\ \bigcap(\tau_{xq} \rightsquigarrow \tau_{xq}) \subseteq^{A\to A} \tau_i \wedge \tau_q \subseteq^A \tau_{xq}$

   We then take $\tau_x \equiv \tau_{xp} +\!\!+ \tau_{xq}$:

$$(trans)\ \dfrac{(trans)\ \dfrac{(\rightsquigarrow\cap')\ \dfrac{}{\bigcap(\tau_x \rightsquigarrow \tau_x) \subseteq^{A\to A} \tau_{xp} \rightsquigarrow \tau_{xp} \cap \tau_{xq} \rightsquigarrow \tau_{xq}} \qquad (mon)\ \dfrac{(IH)\ \dfrac{}{\tau_{xp} \rightsquigarrow \tau_{xp} \subseteq^{A\to A} \tau_i} \qquad (IH)\ \dfrac{}{\tau_{xq} \rightsquigarrow \tau_{xq} \subseteq^{A\to A} \tau_i}}{\tau_{xp} \rightsquigarrow \tau_{xp} \cap \tau_{xq} \rightsquigarrow \tau_{xq} \subseteq^{A\to A} \tau_i +\!\!+ \tau_i}}{\bigcap(\tau_x \rightsquigarrow \tau_x) \subseteq^{A\to A} \tau_i +\!\!+ \tau_i} \qquad (\subseteq)\ \dfrac{\dfrac{}{\tau_i +\!\!+ \tau_i \subseteq \tau_i}}{\tau_i +\!\!+ \tau_i \subseteq^{A\to A} \tau_i}}{\bigcap(\tau_x \rightsquigarrow \tau_x) \subseteq^{A\to A} \tau_i}$$

$$(trans)\ \dfrac{\dfrac{}{\tau_j \subseteq^A \tau_p +\!\!+ \tau_q} \qquad (mon)\ \dfrac{(IH)\ \dfrac{}{\tau_p +\!\!+ \subseteq^A \tau_{xp}} \qquad (IH)\ \dfrac{}{\tau_q +\!\!+ \subseteq^A \tau_{xq}}}{\tau_p +\!\!+ \tau_q \subseteq^A \tau_x}}{\tau_j \subseteq^A \tau_x}$$

ii) Follows in a similar fashion.

Proofs of subject expansion and reduction

An interesting property of the intersection types, is the fact that they admit both subject expansion and subject reduction, namely $\Vdash$ is closed under β-equality. Subject expansion and reduction are proved in two separate lemmas:

Theorem ($\Vdash$ closed under $=_\beta$)

i) $\Gamma \Vdash_s m : \tau \implies m \Rightarrow_\beta m' \implies \Gamma \Vdash_s m' : \tau$
ii) $\Gamma \Vdash m : \tau_i \implies m \Rightarrow_\beta m' \implies \Gamma \Vdash m' : \tau_i$
iii) $\Gamma \Vdash_s m' : \tau \implies m \Rightarrow_\beta m' \implies \Gamma \Vdash_s m : \tau$
iv) $\Gamma \Vdash m' : \tau_i \implies m \Rightarrow_\beta m' \implies \Gamma \Vdash m : \tau_i$

Proof: By induction on $\Rightarrow_\beta$. The proofs in both directions follow by straightforward induction for all the rules except for (Y) and $(beta)$. Note that the (Y) rule here is not the typing rule, but rather the reduction rule $Y_A m \Rightarrow_\beta m(Y_A m)$.

i) (Y): By assumption, we have $Y_A m \Rightarrow_\beta m(Y_A m)$ and $\Gamma \Vdash_s Y_A m : \tau$. By case analysis of the last rule applied in the derivation tree of $\Gamma \Vdash_s Y_A m : \tau$, we have two cases:
   - (app) We have:

$$(app)\ \dfrac{\dfrac{\vdots}{\Gamma \Vdash_s Y_A : \tau_i \rightsquigarrow \tau_j} \qquad \dfrac{\vdots}{\Gamma \Vdash m_{A \to A} : \tau_i}}{\Gamma \Vdash_s Y_A m : \tau}\ (\bigcap \tau \subseteq^A \tau_j)$$

     Then, by (Y-inv) we have some τ_x s.t $\bigcap(\tau_x \rightsquigarrow \tau_x) \subseteq^{A \to A} \tau_i \wedge \tau_j \subseteq^A \tau_x$.
   - $(\rightsquigarrow\cap)$ Then we have:

$$(\rightsquigarrow\cap)\ \dfrac{\dfrac{\vdots}{\Gamma \Vdash_s Y_{B \to C} m : \tau_i \rightsquigarrow \tau_j} \qquad \dfrac{\vdots}{\Gamma \Vdash_s Y_{B \to C} m : \tau_i \rightsquigarrow \tau_k}}{\Gamma \Vdash_s Y_{B \to C} m : \tau_i \rightsquigarrow \tau_{jk}}\ (\tau_{jk} \subseteq^C \tau_j \mathbin{+\!\!\!+} \tau_k)$$

     Where $A \equiv B \to C$.
     By IH, we get $\Gamma \Vdash_s m(Y_{B \to C} m) : \tau_i \rightsquigarrow \tau_j$ and $\Gamma \Vdash_s m(Y_{B \to C} m) : \tau_i \rightsquigarrow \tau_k$, thus from $(\rightsquigarrow\cap)$ it follows that $\Gamma \Vdash_s m(Y_{B \to C} m) : \tau_i \rightsquigarrow \tau_{jk}$