We start with some simple examples, that explains the method. To make the examples more readable we will use numbers to denote atomic types and we will add to each inference rule used in the deduction the type equation which makes it valid.

10.2.1. EXAMPLE. (i) Let $M \equiv \lambda x.xx$. In order to type M, we build from below the following derivation.

$$\cfrac{\cfrac{x{:}1 \vdash x : 1 \quad x{:}1 \vdash x : 1 \quad 1 = 1\to 2}{x{:}1 \vdash xx : 2 \qquad 3 = 1\to 2}}{\vdash \lambda x.xx : 3}$$

This gives the triple

$$a_M = 3, \Gamma_M = \emptyset, \mathcal{A}_M = \mathbb{T}^{\mathbf{c}_M}/\mathcal{E}_M,$$

where $\mathbf{c}_M = \{1, 2, 3\}$ and $\mathcal{E}_M = \{1 = 1\to 2, 3 = 1\to 2\}$. We can simplify this triple to the isomorphic $T_M = 1$, $\Gamma_M = \emptyset$, $\mathcal{A}_M = \mathbb{T}^{\{1,2\}}/\{1 = 1\to 2\}$. Indeed

$$\vdash_{\mathcal{A}_M} (\lambda x.xx) : 1$$

In order to show that this assignment is initial, suppose that $\Gamma \vdash_{\mathcal{A}} M : a_1$. Then one can reconstruct from below, using the generation Lemma 10.1.3.

$$\cfrac{\cfrac{x{:}a_1 \vdash x : a_1 \quad x{:}a_1 \vdash x : a_1 \quad a_1 = a_1\to a_2}{x{:}a_1 \vdash xx : a_2 \qquad a_1 = a_1\to a_2}}{\vdash \lambda x.xx : a_1}$$

The required morphism is determined by $h(k) = a_k$ for $k \leq 2$. Indeed one has $h(a_M) = a_0$ and $h(\Gamma_M) \subseteq \Gamma$.

(ii) If we consider the (open) term $M \equiv x(xx)$ we have, using Church notation to represent deduction in a more compact way:

$$\vdash_{\mathcal{A}_M} x^1\,(x^1\,x^1)^2 : 3,$$

where $a_M = 3$, $\mathcal{A}_M = \mathbb{T}^{\{1,2,3\}}/\{1 = 1\to 2 = 2\to 3\}$ and $\Gamma_M = \{x : 1\}$.

Moreover if we want to consider type assignment with respect to invertible type algebras we can convert $\mathcal{A}_M$ in a invertible type algebra $\mathcal{A}_M^{inv}$. Assuming invertibility we get $1 = 2 = 3$ and then $\mathcal{A}_M^{inv}$ can be simplified to the *trivial* type algebra $\mathbb{T}^{\{1\}}/\{1 = 1\to 1\}$.

(iii) If we consider term having a simple type in $(\lambda \to)$ the construction sketched above that does not imply recursive definitions as in case (i). Moreover if we assume invertibility the resulting type algebra is isomorphic to a free one and we get the same principal type as in $(\lambda\to)$. Let $M \equiv \mathbf{c}_2 \equiv \lambda fx.f(fx)$.

$$\cfrac{\cfrac{\cfrac{\cfrac{f{:}1, x{:}2 \vdash f{:}1 \quad f{:}1, x{:}2 \vdash x : 2 \quad 1 = 2\to 3}{f{:}1, x{:}3 \vdash fx : 3 \qquad 1 = 3\to 4}}{f{:}1, x{:}2 \vdash f(fx) : 4 \qquad 5 = 2\to 4}}{f{:}1 \vdash \lambda x.f(fx) : 5 \qquad 6 = 1\to 5}}{\vdash \lambda fx.f(fx) : 6}$$

Simplifying this gives the triple

$$\vdash_{\mathcal{E}} \mathbf{c}_2 : (2\to 3)\to(2\to 4), \tag{1}$$

with $\mathcal{E}_{\mathbf{c}_2} = \{2\to 3 = 3\to 4\}$. We can understand this by looking at

$$\vdash \lambda f^{(2\to 3)=(2\to 4)}\, x^2.(f^{2\to 4}(f^{2\to 3}x^2)^3)^4 : (2\to 3)\to 2\to 4.$$

Also in $\lambda_{\to}$ this term M can be typed.

$$\vdash_{\lambda_{\to}} \mathbf{c}_2 : (\alpha\to\alpha)\to\alpha\to\alpha, \tag{2}$$

Note that there is a morphism $h : \mathbb{T}/\mathcal{E}_{\mathbf{c}_2} \to \mathbb{T}^{\alpha}$ determined by

$$h(2) = h(3) = h(4) \quad = \quad \alpha.$$

This h respects the equations in $\mathcal{E}_{\mathbf{c}_2}$. In this way the type assignment (2) is seen to follow from (1), applying Lemma 8.1.18.

Note also in this case assuming invertibility we get $2 = 3 = 4$ and so $\mathcal{E}_{\mathbf{c}_2}$ contains only identities. We have than that $\mathcal{A}^{inv}_{\mathbf{c}_2}$ is isomorphic to $\mathbb{T}^{\{\alpha\}}$, the free type algebra over the atomic type α.

(iv) $M \equiv \mathsf{II}$. Bottom-up we construct the following derivation-tree.

$$\dfrac{\dfrac{x{:}2 \vdash x : 2 \quad 1 = 2\to 2}{\vdash (\lambda x.x) : 1} \qquad \dfrac{y{:}2' \vdash y : 2' \quad 1' = 2'\to 2'}{\vdash (\lambda y.y) : 1'} \qquad 1 = 1'\to 0}{\vdash (\lambda x.x)(\lambda y.y) : 0}$$

Changeing names of type constants, this gives the following type assignment

$$\vdash_{\mathcal{E}} \mathsf{II} : 0, \tag{3}$$

with $\mathcal{E} = \{1\to 1 = (1'\to 1')\to 0\}$ on $\mathbb{T}^{\{0,1,1'\}}$. In $\lambda_{\to}$ this term has as principal type $\alpha\to\alpha$. This can be obtained as image of (3) under the morphism $h : \mathbb{T}^{\{0,1,1'\}} \to \mathbb{T}^{\{\alpha\}}$ defined by

$$\begin{aligned} h(0) &= \alpha\to\alpha; \\ h(1) &= \alpha\to\alpha; \\ h(1') &= \alpha. \end{aligned}$$

Note that it was important to keep the names of the bound variables of the two occurrences of I different.

We present now the formal algorithm to build the principal triple for a term M. To simplify its definition we make the assumption that the names of all free and bound variables of M are distinct. This can always be achieved by α-conversion.

10.2.2. Definition. **[PTS]**
Let $M \in \Lambda$. Assume that all free and bound variables have maximally different names. Define a set of type constants $\mathbf{c}_M$, a type a_M, a basis Γ_M, a set of

equations $\mathcal{E}_M$, and a type algebra $\mathcal{A}_M$ with element a_M as follows. We do this by defining first for each subterm-occurrence $L \subseteq M$ for L not a variable a type atom α_L. For variable subterms x we choose a fixed type α_x independent of the occurrence of x. Then we define $\mathcal{E}_L$ for each for each subterm-occurrence $L \subseteq M$, obtaining this notion also for M as highest subterm of itself.

L	$\mathcal{E}_L$
x	$\emptyset$
PQ	$\mathcal{E}_P \cup \mathcal{E}_Q \cup \{\alpha_P = \alpha_Q \to \alpha_{PQ}\}$
$\lambda x.P$	$\mathcal{E}_P \cup \{\alpha_{\lambda x.P} = \alpha_x \to \alpha_P\}$

Define $\mathbf{c}_M$ as the set of all atomic types occurring in $\alpha_x, \mathcal{E}_M$. Finally we define

$$\begin{aligned} \Gamma_M &= \{x{:}\alpha_x \mid x \in \mathrm{FV}(M)\}; \\ \mathcal{A}_M &= \mathbb{T}^{\mathbf{c}_M}/\mathcal{E}_M; \\ a_M &= [\alpha_M]_{\mathcal{E}_M}. \end{aligned}$$

The type a_M is called the *principal type* of M *for recursive types* and $\mathcal{A}_M$ its *principal type algebra*. Γ_M is the principal context of M, which is empty if M is closed.

Type inference

Of the following theorem part (iii) solves both versions of the type inference problem.

10.2.3. Theorem. **[PTS.TA]**
[PTSuno]
For every $M \in \Lambda$ there exists a principal triple $\langle \Gamma_M,\ \mathcal{A}_M,\ a_M\rangle$ such that the following holds.

(i) $\Gamma_M \vdash_{\mathcal{A}_M} M : a_M$.

(ii) $\Gamma \vdash_{\mathcal{A}} M : a \quad \Leftrightarrow \quad$ *there is a morphism $h : \mathcal{A}_M \to \mathcal{A}$ such that $h(\Gamma_M) \subseteq \Gamma$ and $h(a_M) = a$.*

(iii) *For $M \in \Lambda^{\emptyset}$ this simplifies to*

$$\vdash_{\mathcal{A}} M : a \;\Leftrightarrow\; \exists h{:}\mathcal{A}_M \to \mathcal{A}.h(a_M) = a.$$

Proof. Take as triple the one defined in the previous Definition.

(i) By induction on the structure of $L \subseteq M$ we show that this statement holds for M replaced by L and hence also for M itself. Case $L \equiv x$. Then clearly $x{:}\alpha_x \vdash x : \alpha_x$. Case $L \equiv PQ$, and $\Gamma_P \vdash_{\mathbb{T}/\mathcal{E}} P : \alpha_P$, $\Gamma_Q \vdash_{\mathbb{T}/\mathcal{E}} Q : \alpha_Q$. Then $\Gamma_P \cup \Gamma_Q \vdash_{\mathbb{T}/\mathcal{E}} PQ : \alpha_{PQ}$, as $\alpha_P =_{\mathcal{E}} \alpha_Q \to \alpha_{PQ}$. Case $L \equiv \lambda x.P$ and $\Gamma_P \vdash_{\mathbb{T}/\mathcal{E}} P : \alpha_P$. Then $\Gamma_P - \{x{:}\alpha_x\} \vdash_{\mathbb{T}/\mathcal{E}} \lambda x.P : \alpha_x \to \alpha_P = \alpha_{\lambda x.P}$.

(ii) ($\Leftarrow$) By (i), Lemma 8.1.18 and weakening, Proposition 8.1.4.
($\Rightarrow$) By Remark 8.2.6 it is enough to define a morphism $h^\natural : \mathbb{T}^{\mathbf{c}_M} \to \mathcal{A}$ such that $h^\natural(a_M) = A$ and $B = C \in \mathcal{E}_M \;\Rightarrow\; h^\natural(B) = h^\natural(C)$, for all $B, C \in \mathbb{T}^{\mathbf{c}_M}$. As in Definition 10.2.2 we can assume that all bound and free variables in M have distinct names. Take a deduction $\mathcal{D}$ of $\Gamma \vdash_{\mathcal{A}} M : a$. Note that in definition

10.2.2 for every $L \subseteq M$ a type derivation $\mathcal{D}_L$ is constructed in which each variable x occurring in L is assigned type α_x and to each subterm occurrence of L, that is not a variable, a fresh type variable is assigned. Moreover $\mathcal{D}_M$ and $\mathcal{D}$ have the same shape corresponding to the structure of M. Now we define $h^\natural$.

(1) $h^\natural(\alpha_x) = \Gamma(x)$ for each $x \in \mathrm{dom}(\Gamma_M)$;

(2) $h^\natural(\alpha_L) = \alpha'_L$ for all α_L assigned in $\mathcal{D}_M$ to the non variable subterm occurrences L of M, where α'_L is assigned in $\mathcal{D}$ to the corresponding L.

It is easy to verify, using the generation Lemma 10.1.3, that all equations in $\mathcal{E}_M$ are preserved. Moreover, by construction, $h^\natural(\alpha_M) = a$. Obviously we have $h(\Gamma_M) \subseteq \Gamma$, since Γ_M contains only assumptions for variables occurring in M.

(iii) By (ii), knowing that $\Gamma_M = \emptyset$ for $M \in \Lambda^{\emptyset}$. ■

If we want to consider only invertible type algebras, by Lemma 9.3.18, we can take $\mathcal{B}_M = \mathbb{T}^{\mathbf{c}_M}/\mathcal{E}_M^{inv}$ as the initial type algebra for M. Let $\mathcal{R}_M = \mathcal{E}_M^{inv}$. Note that, by Lemma 9.3.17, $\mathcal{R}_M$ is a proper sr and we have $\mathcal{B}_M = \mathbb{T}^{\mathbf{c}_M}/\mathcal{R}_M$.

10.2.4. Corollary. **[PTS.ITA]**
For every $M \in \Lambda$ there exists an invertible principal triple Γ_M, $\mathcal{B}_M$, a_M such that the following holds.

(i) $\Gamma_M \vdash_{\mathcal{B}_M} M : b_M$, *with $\mathcal{B}_M$ invertible.*

(ii) $\Gamma \vdash_{\mathcal{B}} M : a$, *with $\mathcal{B}$ invertible* $\Leftrightarrow$ $\exists h : \mathcal{B}_M \to \mathcal{B}$ $h(\Gamma_M) \subseteq \Gamma$ *and* $h(a_M) = a$.

Proof. Given M, consider its principal type $\Gamma_M, \mathcal{A}_M, a_M$ and replace $\mathcal{A}_M$ by $\mathcal{B}_M = \mathcal{A}_M^{\mathrm{inv}}$. Let $k : \mathcal{A}_M \to \mathcal{B}_M$ be the canonical morphism and take $\Gamma'_M = k(\Gamma_M), b_M = k(a_M)$. Then (i) holds by the Theorem and Lemma 8.1.18. Property (ii) follows by (ii) of the Theorem and Proposition 9.3.18. ■

The typeability problems

Theorem 10.2.3 provides the abstract answer to the question whether a term has a type within a given type algebra $\mathcal{A}$. The first typability question, i.e. given $M \in \Lambda^{\emptyset}$

$$\text{does there exist } \mathcal{A} \text{ and } a \in \mathcal{A} \text{ such that } \vdash_{\mathcal{A}} M : a,$$

is trivially checked: this is always the case. Indeed, take $\mathcal{A} = \mathbb{T}^{\{\alpha\}}/\{\alpha = \alpha \to \alpha\}$ and $a = [\alpha]$; then one has $\vdash_{\mathcal{A}} M : a$, see Exercise 8.6.12. However this problem is not more trivial when term constants are considered, see the discussion in Section 12.1.

The decidability of the second question, given $M \in \Lambda^{\emptyset}$ and type algebra $\mathcal{A}$

$$\text{does there exist an } a \in \mathcal{A} \text{ such that } \vdash_{\mathcal{A}} M : a,$$

only makes sense if $\mathcal{A}$ is 'finitely presented', i.e. of the form $\mathbb{T}^{\mathbb{A}}/\mathcal{E}$ with $\mathbb{A}$ and $\mathcal{E}$ finite. Then typability is decidable.