La théorie des types simples de Church

Alexandre Miquel

La théorie des types simples (due à Church) est une formalisation de la logique d'ordre supérieur dont le langage est construit sur le λ-calcul simplement typé plutôt que sur le langage des théories du premier ordre.

1 Syntaxe

Les types de la théorie des types simples sont les types du λ-calcul simplement typé formés à partir de deux types de base, à savoir le type o (« omicron ») des *propositions*, et le type ι (« iota ») des *objets élémentaires* :

Types $\qquad \tau, \sigma \quad ::= \quad \iota \quad | \quad o \quad | \quad \tau \to \sigma$

Les termes de la théorie des types simples sont tout simplement les termes du λ-calcul simplement typé

Termes $\qquad M, N \quad ::= \quad x \quad | \quad \lambda x\!:\!\tau\,.\,M \quad | \quad MN \quad | \quad \Rightarrow \quad | \quad \forall^\tau$

auxquels on a ajouté des constantes $\Rightarrow$ (implication) et $\forall^\tau$ (quantification universelle sur τ) pour chaque type τ.

La β-réduction est définie comme dans le λ-calcul simplement typé, et comme dans le λ-calcul simplement typé, on note $M \succ M'$ la relation de réduction en une étape, $M \succ^* M'$ la relation de réduction en un nombre arbitraire d'étapes, et $M \simeq M'$ la relation de convertibilité.

Vis-à-vis de la réduction, les constantes $\Rightarrow$ et $\forall^\tau$ sont neutres et se comportent comme de simples variables libres.

Dans le cadre de la théorie des types simples, les contextes de typage sont appelés des *signatures*, et désignés par la lettre Σ. (On préfère réserver la terminologie de contexte et la lettre Γ aux contextes logiques définis dans la section suivante.) Mis à part cette différence purement cosmétique, la définition est exactement la même que dans le λ-calcul simplement typé :

Signatures $\qquad \Sigma \quad ::= \quad x_1 : \tau_1,\ \ldots,\ x_n : \tau_n$

(avec la contrainte que $x_i \not\equiv x_j$ si $i \neq j$).

Les règles de typage sont celles du λ-calcul simplement typé auxquelles on ajoute deux règles pour typer les constantes $\Rightarrow$ et $\forall^\tau$:

$$\frac{}{\Sigma \vdash x : \tau}\ {\scriptstyle (x:\tau)\in\Sigma} \qquad \frac{\Sigma,\ x : \tau \vdash M : \sigma}{\Sigma \vdash \lambda x\!:\!\tau\,.\,M : \tau \to \sigma} \qquad \frac{\Sigma \vdash M : \tau \to \sigma \quad \Sigma \vdash N : \tau}{\Sigma \vdash MN : \sigma}$$

$$\frac{}{\Sigma \vdash \Rightarrow\, : o \to o \to o} \qquad \frac{}{\Sigma \vdash \forall^\tau : (\tau \to o) \to o}$$

(On retrouve évidemmenent toutes les propriétés du système de types du λ-calcul simplement typé.)

Dans ce qui suit, on utilise les abréviations :[1]

$$A \Rightarrow B \quad \equiv \quad \Rightarrow A\ B \qquad\qquad \text{et} \qquad\qquad \forall x\!:\!\tau\,.\,A \quad \equiv \quad \forall^{\tau}\ (\lambda x\!:\!\tau\,.\,A)\,.$$

Ces deux abréviations sont bien typées, dans le sens où les règles suivantes sont admissibles :

$$\frac{\Sigma \vdash A : o \qquad \Sigma \vdash B : o}{\Sigma \vdash A \Rightarrow B : o} \qquad\qquad \frac{\Sigma,\ x : \tau \vdash A : o}{\Sigma \vdash \forall x\!:\!\tau\,.\,A : o}$$

2 Dérivabilité

Comme en théorie des types simples les propositions sont représentées par des termes du λ-calcul simplement typé (de type o), les contextes logiques sont définis comme des suites de termes :

Contextes $\qquad \Gamma, \Delta \quad ::= \quad A_1, \ldots, A_k$

Afin de s'assurer que tous les termes qui figurent dans un contexte Γ sont des propositions bien formées dans une signature donnée, on introduit un jugement $\Sigma \vdash \Gamma\ \texttt{ctx}$ (« dans la signature Σ le contexte Γ est bien formé »). Ce jugement est défini à partir des deux règles d'inférence suivantes :

$$\frac{}{\Sigma \vdash \varnothing\ \texttt{ctx}} \qquad\qquad \frac{\Sigma \vdash \Gamma\ \texttt{ctx} \qquad \Sigma \vdash A : o}{\Sigma \vdash \Gamma,\ A\ \texttt{ctx}}$$

La partie purement logique du formalisme, enfin, est assurée par un jugement $\langle\Sigma\rangle\ \Gamma \vdash A$ (« dans la signature Σ et sous les hypothèses Γ, on a A ») dont les règles d'inférence sont les suivantes :

$$\frac{\Sigma \vdash \Gamma\ \texttt{ctx}}{\langle\Sigma\rangle\ \Gamma \vdash A}\ {\scriptstyle (A \in \Gamma)}$$

$$\frac{\langle\Sigma\rangle\ \Gamma, A \vdash B}{\langle\Sigma\rangle\ \Gamma \vdash A \Rightarrow B} \qquad\qquad \frac{\langle\Sigma\rangle\ \Gamma \vdash A \Rightarrow B \qquad \langle\Sigma\rangle\ \Gamma \vdash A}{\langle\Sigma\rangle\ \Gamma \vdash B}$$

$$\frac{\langle\Sigma; [x : \tau]\rangle\ \Gamma \vdash A}{\langle\Sigma\rangle\ \Gamma \vdash \forall x\!:\!\tau\,.\,A}\ {\scriptstyle (x \notin FV(\Gamma))} \qquad\qquad \frac{\langle\Sigma\rangle\ \Gamma \vdash \forall x\!:\!\tau\,.\,A \qquad \Sigma \vdash N : \tau}{\langle\Sigma\rangle\ \Gamma \vdash A\{x := N\}}$$

$$\frac{\langle\Sigma\rangle\ \Gamma \vdash A \qquad \Sigma \vdash A' : o}{\langle\Sigma\rangle\ \Gamma \vdash A'}\ {\scriptstyle (A =_{\beta\delta} A')}$$

On retrouve ainsi les règles habituelles du fragment $\forall/\Rightarrow$ de la déduction naturelle intuitionniste (Axiome + $\Rightarrow$-intro/élim + $\forall$-intro/élim.) étendues au nouveau formalisme. On notera cependant que le passage à un langage plus riche nécessite d'introduire des contraintes de typage sous la forme de prémisses supplémentaires de la forme $\Sigma \vdash \Gamma\ \texttt{ctx}$ ou $\Sigma \vdash N : \tau$. Ainsi :

[1] On notera que la quantification universelle $\forall x\!:\!\tau\,.\,A(x)$ est décomposée en deux étapes : d'abord une étape de liaison — effectuée par la λ-abstraction — suivie de l'application d'un opérateur. De façon plus générale, tous les symboles lieurs en mathématiques (les quantificateurs, le symbole « $\sum$ » de sommation, le symbole d'intégrale, etc.) ou en informatique (la boucle `for` de Caml) peuvent être décomposés de cette manière. Dans le cadre d'une implémentation, cette décomposition est intéressante dans la mesure où elle fait reposer tout le problème — épineux ! — de la liaison de variable sur un unique symbole : la λ-abstraction.

– La règle axiome comporte à présent une prémisse $\Sigma \vdash \Gamma\ \texttt{ctx}$ qui garantit la bonne formation du contexte.[2]
– La règle d'élimination du quantificateur universel comporte à présent une prémisse de typage $\Sigma \vdash N : \tau$ qui sert juste à garantir que l'objet N avec lequel la quantification est éliminée est bien formé et a le bon type.

La seule vraie nouvelle règle dans le formalisme est la dernière règle, appelée *règle de conversion* :

$$\frac{\langle\Sigma\rangle\ \Gamma \vdash A \qquad \Sigma \vdash A' : o}{\langle\Sigma\rangle\ \Gamma \vdash A'}\ {\scriptstyle (A \simeq A')}$$

qui exprime que toute proposition A' calculatoirement équivalente ($A' \simeq A$) à une proposition A dérivable est elle même dérivable — sous réserve que cette nouvelle proposition A' soit elle-même bien typée. Cette règle est en fait fondamentale, car c'est elle qui permet d'incorporer le calcul au niveau des démonstrations, ainsi que l'illustre le fragment de dérivation suivant

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \langle\rangle\ \vdash \forall x:\iota\,.\,x =_\iota x\end{array} \qquad \begin{array}{c}\vdots\\ \vdash 4 : \iota\end{array}}{\langle\rangle\ \vdash 4 =_\iota 4} \qquad \begin{array}{c}\vdots\\ \vdash 2+2 =_\iota 4 : o\end{array}}{\langle\rangle\ \vdash 2+2 =_\iota 4}\ {\scriptstyle (2+2=_\iota 4)\simeq(4=_\iota 4)}$$

dans lequel une preuve de $2+2 =_\iota 4$ est déduite de la propriété de réflexivité par simple application de la règle de conversion en utilisant le fait que les propositions $4 =_\iota 4$ et $2+2 =_\iota 4$ sont convertibles. (On suppose donné un codage des entiers et de l'addition pour lequel les termes $2+2$ et 4 sont convertibles).

Les propriétés syntaxiques élémentaires du formalisme sont les suivantes :

Bonne formation du séquent

Si $\langle\Sigma\rangle\ \Gamma \vdash A$, alors $\Sigma \vdash \Gamma\ \texttt{ctx}$ et $\Sigma \vdash A : o$.

Affaiblissement généralisé

Si $\langle\Sigma\rangle\ \Gamma \vdash A$, $\Sigma \vdash \Gamma'\ \texttt{ctx}$ et $\Gamma \subset \Gamma'$, alors $\langle\Sigma\rangle\ \Gamma' \vdash A$.

Substitutivité

Si $\langle\Sigma,\ x:\tau\rangle\ \Gamma \vdash A$, et $\Sigma \vdash N : \tau$ alors $\langle\Sigma\rangle\ \Gamma'\{x := N\} \vdash A\{x := N\}$.

3 Codages des connecteurs et de l'égalité

Bien que la logique de la théorie des types simples soit ultimement basée sur l'implication et la quantification universelle, les autres connecteurs et la quantification existentielle sont définissables à partir de ces deux constructions primitives en utilisant la quantification du *second-ordre*, c'est-à-dire la quantification « $\forall^o$ » sur toutes les propositions.

Unités et connecteurs Ainsi, les unités $\top$ et $\bot$, la conjonction $A \wedge B$ et la disjonction $A \vee B$ sont définies en théorie des types simples par :

$$\begin{array}{rcl}
\top & \equiv & \forall x:o\,.\,(x \Rightarrow x) \\
\bot & \equiv & \forall x:o\,.\,x \\
A \wedge B & \equiv & \forall x:o\,.\,((A \Rightarrow B \Rightarrow x) \Rightarrow x) \\
A \vee B & \equiv & \forall x:o\,.\,((A \Rightarrow x) \Rightarrow (B \Rightarrow x) \Rightarrow x)
\end{array}$$

[2]Ce qui permet d'éviter de dériver des séquents insensés, tels que $\langle\Sigma\rangle\ 3 \vdash 3$.

Avec ces définitions, les règles d'introduction et d'élimination des connecteurs $\top$, $\bot$, $\wedge$ et $\vee$ en déduction naturelle deviennent admissibles en théorie des types simples (après ajout des contraintes de typage nécessaires) :

$$(\top\text{-intro},\ \bot\text{-élim})\qquad \frac{\Sigma \vdash \Gamma\ \texttt{ctx}}{\langle\Sigma\rangle\ \Gamma \vdash \top} \qquad\qquad \frac{\langle\Sigma\rangle\ \Gamma \vdash \bot \qquad \Sigma \vdash A : o}{\langle\Sigma\rangle\ \Gamma \vdash A}$$

$$(\wedge\text{-intro})\qquad \frac{\langle\Sigma\rangle\ \Gamma \vdash A \quad \langle\Sigma\rangle\ \Gamma \vdash B}{\langle\Sigma\rangle\ \Gamma \vdash A \wedge B}$$

$$(\wedge\text{-élim}_{1,2})\qquad \frac{\langle\Sigma\rangle\ \Gamma \vdash A \wedge B}{\langle\Sigma\rangle\ \Gamma \vdash A} \qquad\qquad \frac{\langle\Sigma\rangle\ \Gamma \vdash A \wedge B}{\langle\Sigma\rangle\ \Gamma \vdash B}$$

$$(\vee\text{-intro}_{1,2})\qquad \frac{\langle\Sigma\rangle\ \Gamma \vdash A \quad \Sigma \vdash B : o}{\langle\Sigma\rangle\ \Gamma \vdash A \vee B} \qquad\qquad \frac{\langle\Sigma\rangle\ \Gamma \vdash B \quad \Sigma \vdash A : o}{\langle\Sigma\rangle\ \Gamma \vdash A \vee B}$$

$$(\vee\text{-élim})\qquad \frac{\langle\Sigma\rangle\ \Gamma, A \vdash C \quad \langle\Sigma\rangle\ \Gamma, B \vdash C \quad \langle\Sigma\rangle\ \Gamma \vdash A \vee B}{\langle\Sigma\rangle\ \Gamma \vdash C}$$

Quantification existentielle Pour tout type simple τ on pose

$$\exists^\tau \quad\equiv\quad \lambda p : (\tau \to o) . \forall z : o . (\forall x : \tau . (p\, x \Rightarrow z) \ \Rightarrow\ z) \quad : \quad (\tau \to o) \to o$$

et on introduit l'abréviation $\exists x : \tau . A \ \equiv\ \exists^\tau\ (\lambda x : \tau . A)$. Là encore, les règles d'introduction et d'élimination de la quantification existentielle deviennent admissibles (après ajout des contraintes de typage nécessaires) :

$$(\exists\text{-intro})\qquad \frac{\Sigma,\ x : \tau \vdash A : o \quad \Sigma \vdash N : \tau \quad \langle\Sigma\rangle\ \Gamma \vdash A\{x := N\}}{\langle\Sigma\rangle\ \Gamma \vdash \exists x : \tau . A}$$

$$(\exists\text{-élim})\qquad \frac{\langle\Sigma,\ x : \tau\rangle\ \Gamma, A \vdash B \quad \langle\Sigma\rangle\ \Gamma \vdash \exists x : \tau . A}{\langle\Sigma\rangle\ \Gamma \vdash B}\ {\scriptstyle x \notin FV(\Gamma, B)}$$

Égalité de Leibniz L'égalité de Leibniz $M_1 =_\tau M_2$ (« M_1 égale M_2 dans le type τ ») est définie par :

$$M_1 =_\tau M_2 \quad\equiv\quad \forall p : (\tau \to o) . (p\, M_1 \Rightarrow p\, M_2) .$$

La relation ainsi définie est une relation d'équivalence

$$\begin{array}{l} \forall x : \tau . (x =_\tau x) \\ \forall x : \tau . \forall y : \tau . (x =_\tau y \ \Rightarrow\ y =_\tau x) \\ \forall x : \tau . \forall y : \tau . \forall z : \tau . (x =_\tau y \ \wedge\ y =_\tau z \ \Rightarrow\ x =_\tau z) \end{array}$$

qui satisfait le principe de Leibniz attendu :

$$\forall p : (\tau \to o) . \forall x : \tau . \forall y : \tau .\ (p\, x \ \wedge\ x =_\tau y \ \Rightarrow\ p\, y) .$$