Inductive Type Schemas as Functors

Freiric Barral   Sergeï Soloviev   David Chemouil

IRIT
Institut de Recherche en Informatique de Toulouse
Équipe ACADIE

LMU
Institut für Informatik
Theoretische Informatik

WIT 2005
28 Nov 2005

Types

Definition (Types)

$$(\to)\ \frac{\rho, \tau \in \mathsf{Ty}}{\rho \to \tau \in \mathsf{Ty}} \qquad (\mu)\ \frac{\overrightarrow{\mathsf{c}} \subseteq \mathsf{Const} \qquad \overrightarrow{\rho}, \overrightarrow{\sigma} \subseteq \mathsf{Ty}}{\mu\alpha\,(\overrightarrow{\mathsf{c} : \kappa_{\overrightarrow{\rho},\overrightarrow{\sigma}}(\alpha)})}$$

Definition (Strictly positive Operator)

$$\kappa_{\overrightarrow{\rho},\overrightarrow{\sigma}}(\alpha) = \alpha \mid \rho_i \to \kappa(\alpha) \mid (\sigma_{i_1} \to \cdots \to \sigma_{i_j} \to \alpha) \to \kappa(\alpha)$$

For $\kappa_{\overrightarrow{\rho},\overrightarrow{\sigma}}(\alpha) \equiv \overrightarrow{\tau} \to \alpha$, we note $\kappa^{-}_{\overrightarrow{\rho},\overrightarrow{\sigma}}(\alpha) \equiv \overrightarrow{\tau}$

Definition (Inductive Schemas)

$$\mu\alpha\,(\overrightarrow{k : \kappa_{\overrightarrow{\pi},\overrightarrow{\theta}}(\alpha)})$$

Example

Some definable inductive types:

$\mathbf{N}$	$\hat{=}$	$\mu\alpha\,(0 : \alpha, S : \alpha \to \alpha)$	Peano's naturals
$\mathbf{L}(\rho)$	$\hat{=}$	$\mu\alpha\,(\mathsf{nil} : \alpha, \mathsf{cons} : \rho \to \alpha \to \alpha)$	lists (schema)
$\mathbf{\Sigma}(\rho, \sigma)$	$\hat{=}$	$\mu\alpha\,(\mathsf{in}_L : \rho \to \alpha, \mathsf{in}_R : \sigma \to \alpha)$	sums (schema)
$\mathbf{T}(\rho)$	$\hat{=}$	$\mu\alpha\,(\mathsf{l} : \alpha, \mathsf{b} : (\rho \to \alpha) \to \alpha)$	Tree of arity ρ

Typed Terms

Definition (Terms)

$$t ::= x \mid \lambda x^{\tau} t \mid (t\ t) \mid \mathsf{c_i}^{\mu} \mid (\!| \overrightarrow{t} |\!)^{\mu,\sigma} \quad ,$$

with $x \in \mathrm{Var}$, $i \in \mathbb{N} \setminus \{0\}$ *and* $\sigma, \mu \subseteq \mathrm{Ty}$.

Definition (Typing)

$$\frac{(x,\rho) \in \Gamma}{\Gamma \vdash x : \rho} \qquad \frac{\Gamma, x : \rho \vdash r : \sigma}{\Gamma \vdash \lambda x^{\rho}.r : \rho \to \sigma} \qquad \frac{\Gamma \vdash s : \rho \quad \Gamma \vdash r : \rho \to \sigma}{\Gamma \vdash rs : \sigma}$$

$$\frac{(\mathsf{c} : \kappa_{\vec{\rho},\vec{\sigma}}(\alpha) \in \mu) \quad \Gamma \vdash \overrightarrow{r} : \kappa_{\vec{\rho},\vec{\sigma}}^{-}(\mu)}{\Gamma \vdash \mathsf{c}\,\overrightarrow{r} : \mu} \qquad \frac{\Gamma \vdash \overrightarrow{t} : \overrightarrow{\kappa_{\vec{\rho},\vec{\sigma}}(\tau)}}{\Gamma \vdash (\!| \overrightarrow{t} |\!)^{\mu,\tau} : \mu \to \tau}$$

Rewrite Rules and Reduction

Definition ($\beta\eta\iota$-rewrite rules)

$$
\begin{array}{llll}
(\beta) & (\lambda x^\tau \cdot t)\ u & \longmapsto_\beta & t\{{}^u/{}_x\} \\
(\iota) & (\!|\ \overrightarrow{t}\ |\!)^{\mu,\tau}(\mathsf{c_i}^\mu\ \overrightarrow{p}\ \overrightarrow{u}) & \longmapsto_\iota & t_i\ \overrightarrow{p}\ \overrightarrow{((\!|\ \overrightarrow{t}\ |\!)^{\mu,\tau} \circ u)}. \\
 & & & \text{with } g^{\sigma\to\tau} \circ f^{\overrightarrow{\rho}\to\sigma} = \lambda \overrightarrow{x}^{\overrightarrow{\rho}} g(f\ \overrightarrow{x}) \\
(\eta) & t & \longmapsto_\eta & \lambda x^\tau \cdot t\ x \\
 & & & \text{if } \begin{cases} t : \tau \to \upsilon, x \notin \mathrm{FV}(t) \\ t \text{ is not an abstraction} \\ t \text{ is not in applicative position} \end{cases}
\end{array}
$$

Definition

The one-step reduction relation $\to_R$ is obtained by taking the contextual closure of $\mapsto_R$ (and respecting the proviso of η)

Example

$$\mathrm{nil} : \mathbf{L}(\rho)$$
$$\mathrm{cons} : \rho \to \mathbf{L}(\rho) \to \mathbf{L}(\rho)$$
$$a : \tau$$
$$f : \rho \to \tau \to \tau$$
$$(\!| a, f |\!)\ \mathrm{nil} \longrightarrow_{\iota} a$$
$$(\!| a, f |\!)\ (\mathrm{cons}\ h\ t) \longrightarrow_{\iota} f\ h\ ((\!| a, f |\!)\ t)$$

We will use the infix notation $a{::}l$ for cons.

Example (List)

Given a function $f : \rho \to \rho'$, we can define:

$$\mathrm{Map}(f) ::= (\!|\mathrm{nil}, \lambda xy.(fx)::y|\!)$$

$$\begin{array}{lll} \mathrm{Map}(f)\ \mathrm{nil} & \longrightarrow_{\beta\iota} & \mathrm{nil} \\ \mathrm{Map}(f)\ a::l & \longrightarrow_{\beta\iota} & fa::\mathrm{Map}(f)\ l \end{array}$$

And given two function $f : \rho \to \rho'$ and $g : \rho' \to \rho''$, we can verify that for every list, we have:

$$\begin{array}{ll} \mathrm{Map}(g) \circ \mathrm{Map}(f) l & = \mathrm{Map}(g \circ f) l \\ \mathrm{Map}(\mathrm{id}) l & = l \end{array}$$

Functoriality of Inductive Types

Definition (The functor Cp)

Given $\mu\alpha\,(\overrightarrow{\mathsf{c} : \kappa_{\vec{\rho},\vec{\sigma}}(\alpha)})$, $\mu\alpha\,(\overrightarrow{\mathsf{c}' : \kappa_{\vec{\rho'},\vec{\sigma'}}(\alpha)})$ and $\mu\alpha\,(\overrightarrow{\mathsf{c}'' : \kappa_{\vec{\rho''},\vec{\sigma''}}(\alpha)})$, and

$$\begin{array}{ll} l : \mathsf{c} \to \mathsf{c}', & l' : \mathsf{c}' \to \mathsf{c}'', \\ f_\rho : \rho \to \rho', & g_{\rho'} : \rho' \to \rho'', \\ f_\sigma : \sigma' \to \sigma, & g_{\sigma'} : \sigma'' \to \sigma', \end{array}$$

one can define a function **Cp** such that the functorial laws are provable:

$$\begin{array}{rcl} \mathbf{Cp}(\vec{l'}, \vec{g_{\rho'}}, \vec{g_{\sigma'}}) \circ \mathbf{Cp}(\vec{l}, \vec{f_\rho}, \vec{f_\sigma}) & = & \mathbf{Cp}(\overrightarrow{l' \circ l}, \overrightarrow{g_{\rho'} \circ f_\rho}, \overrightarrow{f_\sigma \circ g_{\sigma'}}) \\ \mathbf{Cp}(\mathrm{id}, \vec{\mathrm{id}}_\rho, \vec{\mathrm{id}}_\sigma) & = & \mathrm{id}_{\mu\alpha\,(\overrightarrow{\mathsf{c}:\kappa_{\vec{\rho},\vec{\sigma}}(\alpha)})} \end{array}$$

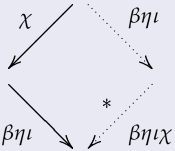
We want to add the following rules:

$$\begin{aligned} \mathbf{Cp}_{\vec{g},\vec{g'}}(\mathbf{Cp}_{\vec{f},\vec{f'}}t) &\longrightarrow_{\chi_\circ} \mathbf{Cp}_{\overrightarrow{g\circ f},\overrightarrow{f'\circ g'}}t \\ \mathbf{Cp}_{\vec{\mathrm{id}},\vec{\mathrm{id}}}t &\longrightarrow_{\chi_{\mathrm{id}}} t \end{aligned}$$

Adjournment

Definition (Adjournment)

Given two binary relations $\beta\eta\iota$ *and* χ, χ *is adjournable w.r.t* $\beta\eta\iota$ *if* $\chi;\beta\eta\iota \;\subseteq\; \beta\eta\iota;(\beta\eta\iota\chi)^*$ *i.e, if the following diagram can be closed:*



Lemma (Adjournment Lemma)

Given two strongly normalising relations $\beta\eta\iota$ *and* χ, $\beta\eta\iota\chi$ *is strongly normalising if* χ *is adjournable w.r.t* $\beta\eta\iota$.

Example

$$\mathrm{Map}(f) ::= (\!|\mathrm{nil}, \lambda xy.(fx)::y|\!)$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)t) \longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)t$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)(x)) \longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)(x)$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)(x))$$

Example

$$\mathrm{Map}(f) ::= (\!|\mathrm{nil}, \lambda xy.(fx)::y|\!)$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)t) \longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)t$$

$$\begin{aligned}\mathrm{Map}(g)(\mathrm{Map}(f)(x)) &\longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)(x)\\ &\longrightarrow_{\beta\eta\iota} \mathrm{Map}(h)(a::l)\\ \mathrm{Map}(g)(\mathrm{Map}(f)(x)) &\end{aligned}$$

Example

$$\mathrm{Map}(f) ::= (\!|\mathrm{nil}, \lambda xy.(fx)::y|\!)$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)t) \longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)t$$

$$\begin{aligned}\mathrm{Map}(g)(\mathrm{Map}(f)(x)) &\longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)(x) \\ &\longrightarrow_{\beta\eta\iota} \mathrm{Map}(h)(a::l) \\ \mathrm{Map}(g)(\mathrm{Map}(f)(x)) &\longrightarrow_{\beta\eta\iota} ???\end{aligned}$$

Definition

The set of Inductive Type together with the terms of type $\mu \to \mu'$ (where μ and μ' are inductive type) defined inductively by:

$$\mathcal{I}_1 \ni f, f' ::= \lambda x^{\mu}.x \mid (\!| \overrightarrow{t} |\!) \mid f \circ f'$$

forms a category $\mathcal{I}$.

Example

$$\mathrm{Map}(f) ::= (\!|\mathrm{nil}, \lambda xy.(fx)::y|\!)$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)t) \longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)t$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)(a::l)) \longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)(a::l)$$

$$\longrightarrow \quad g(fa)::\mathrm{Map}(g \circ f)l$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)(a::l))$$

Example

$$\mathrm{Map}(f) ::= (\!|\mathrm{nil}, \lambda xy.(fx)::y|\!)$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)t) \longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)t$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)(a::l)) \longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)(a::l)$$

$$\longrightarrow \quad g(fa)::\mathrm{Map}(g \circ f)l$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)(a::l)) \longrightarrow_{\beta\iota} g(fa)::\mathrm{Map}(g)(\mathrm{Map}(f)l)$$

Example

$$\text{Map}(f) ::= (\!|\text{nil}, \lambda xy.(fx)::y|\!)$$

$$\text{Map}(g)(\text{Map}(f)t) \longrightarrow_{\chi_\circ} \text{Map}(g \circ f)t$$

$$\text{Map}(g)(\text{Map}(f)(a::l)) \longrightarrow_{\chi_\circ} \text{Map}(g \circ f)(a::l)$$

$$\longrightarrow \quad g(fa)::\text{Map}(g \circ f)l$$

$$\begin{aligned}\text{Map}(g)(\text{Map}(f)(a::l)) &\longrightarrow_{\beta\iota} g(fa)::\text{Map}(g)(\text{Map}(f)l) \\ &\longrightarrow_{\chi_\circ} g(fa)::\text{Map}(g \circ f)l\end{aligned}$$

Example

$$\mathrm{Map}(f) ::= (\!|\mathrm{nil}, \lambda xy.(fx)::y|\!)$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)t) \longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)t$$

$$\begin{aligned} \mathrm{Map}(g)(\mathrm{Map}(f)(a::l)) &\longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)(a::l) \\ &\longrightarrow_{\iota} ((\lambda xy.((g \circ f)x)::y)a(\mathrm{Map}(g \circ f)l)) \\ &\longrightarrow g(fa)::\mathrm{Map}(g \circ f)l \end{aligned}$$

$$\begin{aligned} \mathrm{Map}(g)(\mathrm{Map}(f)(a::l)) &\longrightarrow_{\beta\iota} g(fa)::\mathrm{Map}(g)(\mathrm{Map}(f)l) \\ &\longrightarrow_{\chi_\circ} g(fa)::\mathrm{Map}(g \circ f)l \end{aligned}$$

Example

$$\mathrm{Map}(f) ::= (\!|\mathrm{nil}, \lambda xy.(fx)::y|\!)$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)t) \longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)t$$

$$\begin{aligned}
\mathrm{Map}(g)(\mathrm{Map}(f)(a::l)) &\longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)(a::l) \\
&\longrightarrow_{\iota} ((\lambda xy.((g \circ f)x)::y)a(\mathrm{Map}(g \circ f)l)) \\
&\longrightarrow_{\beta} (g \circ f)a::\mathrm{Map}(g \circ f)l \\
&\longrightarrow g(fa)::\mathrm{Map}(g \circ f)l
\end{aligned}$$

$$\begin{aligned}
\mathrm{Map}(g)(\mathrm{Map}(f)(a::l)) &\longrightarrow_{\beta\iota} g(fa)::\mathrm{Map}(g)(\mathrm{Map}(f)l) \\
&\longrightarrow_{\chi_\circ} g(fa)::\mathrm{Map}(g \circ f)l
\end{aligned}$$

Example

$$\mathrm{Map}(f) ::= (\!|\mathrm{nil}, \lambda xy.(fx)::y|\!)$$

$$\mathrm{Map}(g)(\mathrm{Map}(f)t) \longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)t$$

$$\begin{aligned}
\mathrm{Map}(g)(\mathrm{Map}(f)(a::l)) &\longrightarrow_{\chi_\circ} \mathrm{Map}(g \circ f)(a::l) \\
&\longrightarrow_{\iota} ((\lambda xy.((g \circ f)x)::y)a(\mathrm{Map}(g \circ f)l)) \\
&\longrightarrow_{\beta} (g \circ f)a::\mathrm{Map}(g \circ f)l \\
&\longrightarrow_{\beta} g(fa)::\mathrm{Map}(g \circ f)l
\end{aligned}$$

$$\begin{aligned}
\mathrm{Map}(g)(\mathrm{Map}(f)(a::l)) &\longrightarrow_{\beta\iota} g(fa)::\mathrm{Map}(g)(\mathrm{Map}(f)l) \\
&\longrightarrow_{\chi_\circ} g(fa)::\mathrm{Map}(g \circ f)l
\end{aligned}$$

Modified ι-reductions

$$(\!|\ \vec{t}\ |\!)^{\mu,\tau}(c_i{}^{\mu}\,\vec{p}\,\vec{u}) \quad \longmapsto_\iota \quad t_i\ \vec{p}\ \overrightarrow{((\!|\ \vec{t}\ |\!)^{\mu,\tau} \circ u)}$$

Definition

$$\overrightarrow{(\!|\lambda\vec{x}\,\vec{y}.t|\!)}c_i\,\vec{p}\,\vec{r} \quad \longmapsto_{\iota 2} \quad t_i\{\vec{p}\,/\,\vec{x}\}\langle\overrightarrow{(\!|\lambda\vec{x}\,\vec{y}.t|\!)\bullet r}\,/\,\vec{y}\rangle$$

$$\mathcal{I}t(\vec{y}) \ni t ::= y_i\ \vec{t}\ \mid x \mid \lambda z.t \mid tt \mid (\!|\ \vec{t}\ |\!) \mid c_i$$

$$\begin{aligned} y_i\ \vec{t}\ \langle\overrightarrow{u\bullet r}\,/\,\vec{y}\rangle &::= u(r\ \vec{t}\ ) \\ x\langle\overrightarrow{u\bullet r}\,/\,\vec{y}\rangle &::= x \end{aligned}$$

Example

$$\text{Map}(g \circ f)(a{::}l) \longrightarrow_\iota (g \circ f)a{::}\text{Map}(g \circ f)l$$

Convergence

Theorem (Convergence of modified ι)

The system equipped with the modififed ι-reduction is convergent and generate the same conversion relation as with the standard ι-conversion.

Proof.

$\beta\eta\iota 2$ is embeddable in $\beta\eta\iota$, i.e. for each reduction in $\beta\eta\iota 2$ there exist a sequence of reduction in $\beta\eta\iota$, hence $\beta\eta\iota 2$ is strongly normalizing.

The set of normal form for $\beta\eta\iota$ an $\beta\eta\iota 2$ are the same. Moreover $\xrightarrow{+}_{\beta\eta\iota 2}$ is a subrelation of $\xrightarrow{+}_{\beta\eta\iota}$, hence the set of normal form of a term t are the same in the two systems, hence $\beta\eta\iota 2$ is confluent. □

Definition (The Copy function on the category of functions defined by iterators)

$\mathbf{Cp}(I, \overrightarrow{f}, \overrightarrow{f'}) := (\!|\overrightarrow{t}|\!)$ *with* $\overrightarrow{t} = t_1, \ldots, t_n$, $I(\mathrm{c_k}) = \mathrm{c}'_k$, *and*

$$t_k = \lambda \overrightarrow{x}\, \overrightarrow{y_0}\, \overrightarrow{y_1} \cdot c'_k \; \overrightarrow{\circ_x(f)} \; \overrightarrow{y_0} \; \overrightarrow{\lambda \overrightarrow{z}.y_1 \overrightarrow{\circ_z(f')}} \; .$$

where the function $\circ_x(f_k)$ *is defined by induction on the structure of* f_k*(resp.* f'_k*):*

- $\circ_x(\lambda x^{\mu}.x) := x$
- $\circ_x((\!|\overrightarrow{t}|\!)) := (\!|\overrightarrow{t}|\!)x$
- $\circ_x(f \circ f') := \circ_x(f)\{^{\circ_x(f')}/x\}$

Example

$$\mathrm{Map}(g \circ f) := (\!|\mathrm{nil}, \lambda xy.(g(fx))::y|\!) \neq (\!|\mathrm{nil}, \lambda xy.((g \circ f)x))::y|\!)$$

Theorem

The χ-reductions restricted to the category of iterator $\mathcal{I}$ is convergent.

Proof.

- SN by adjournment.
- Local confluence by case analysis.

□

Conclusion

- Other subcategory of the system.
- Generalize to the whole underlying category.
- Other method to prove strong normalization.

D. Chemouil.
Types inductifs, isomorphismes et récriture extensionnelle.
Thèse de doctorat, Université Paul Sabatier, Toulouse, septembre 2004.

D. Chemouil and S. Soloviev.
Remarks on isomorphisms of simple inductive types.
Electronic Notes in Theoretical Computer Science 85, 7, juillet 2003. Elsevier,.

M. Okada and P. J. Scott.
A note on rewriting theory on for uniqueness of iteration.
Theory and Applications of Categories, 6(4):47–64, 1999.

V. Vene.
Categorical Programming with Inductive and Coinductive Types.
PhD thesis (Diss. Math. Univ. Tartuensis 23), Dept. of Computer Science, Univ. of Tartu, Aug. 2000.