Inductive Type Schemas as Functors

Freiric Barral[1] and Sergei Soloviev[2]

[1] LMU München, Institut für Informatik,
Oettingenstraße 67, 80 538 München, Germany
barral@tcs.informatik.uni-muenchen.de
[2] IRIT, Université Paul Sabatier,
118 route de Narbonne, 31062 Toulouse Cedex 4, France
soloviev@irit.fr

Abstract. Parametric inductive types can be seen as functions taking type parameters as argument and returning the instantiated inductive types. Given functions between parameters one can construct function between the instantiated inductive types representing the change of parameters along these functions. It is well known that it is not a functor w.r.t intensional equality based on standard reductions. We investigate a simple type system with inductive types and iteration and show by modular rewriting techniques that new reductions can be safely added to make this construction a functor, while the decidability of the internal conversion relation based on the strong normalization and confluence properties is preserved. Possible applications: new categorical and computational structures on λ-calculus, certified computation.

A Proofs

Proof (proof of theorem 2 (a)). The embedding is the identity and one $\beta\eta\iota 2$-reduction is simulated by one ι-reduction followed by several β-reductions.

Proof (proof of theorem 2 (b)). $\beta\eta\iota 2$ is embeddable in $\beta\eta\iota$, i.e. for each reduction in $\beta\eta\iota 2$ there exists a sequence of reduction in $\beta\eta\iota$, hence $\beta\eta\iota 2$ is strongly normalizing.
If t is a normal form in $\beta\eta\iota$, then t is a normal form in $\beta\eta\iota 2$ (the reductions of $\beta\eta\iota 2$ apply on redexes of $\beta\eta\iota$ under some restrictions). If t is a normal form in $\beta\eta\iota 2$, then t is a normal form in $\beta\eta\iota$. 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 $\beta\eta\iota$, hence the set of normal form of a term t are the same in the two systems, hence $\beta\eta\iota 2$ is confluent.

Proof (proof of lemma 1). Let's suppose that there is an infinite RS-derivation beginning from a term t. As R and S are strongly normalizing, this derivation consists of an alternation of finite R- and S-derivation. In particular the derivation is of the form

$$t \longrightarrow_X t' \longrightarrow_S \longrightarrow_R \xrightarrow{\infty}_{RS}$$

with an initial fragment $X = R^*$ or $X = R * S^+$. but we can adjourn the derivation following t' to obtain a derivation

$$t \longrightarrow_X t' \longrightarrow_R \xrightarrow{\infty}_{RS}$$

The iteration of this process will indefinitely increase the number of R-reductions in the beginning preserving the infinite "tail" and we shall have a contradiction with the assumption that R is SN.

To ease the exposition we will need to single out a particular occurrence of a subterm t' of a term t, we will then use the notation $C[t']$ for the term t, in this notation C is called a context.

Proof (proof of theorem 3). Since $\chi_\circ$ is strongly normalizing (by simple argument on the size of the term), it remains to show that $\chi_\circ$ is adjournable w.r.t $\beta\eta\iota$. It is adjournable iff it is adjournable w.r.t all infinite derivations $d = \longrightarrow_{\chi_\circ} \longrightarrow_{\beta\eta\iota} d'$. We consider the reduction $\longrightarrow_{\beta\eta\iota}$ following the first $\longrightarrow_{\chi_\circ}$-segment in d. Let consider a $\chi_\circ$-reduction followed by a ι-reduction (difficult case).

$$C[\mathbf{Cp}_{\vec{g},\vec{g'}}(\mathbf{Cp}_{\vec{f},\vec{f'}}\mathsf{c_k}\,\vec{p}\,\vec{r})] \longrightarrow_{\chi_\circ} C[\mathbf{Cp}_{\overrightarrow{g\circ f},\overrightarrow{f'\circ g'}}\mathsf{c_k}\,\vec{p}\,\vec{r}]$$
$$\longrightarrow_{\iota 2} C[\mathsf{c_k}\overrightarrow{\circ_x(g\circ f)\{^p/_x\}}\overrightarrow{\lambda\vec{z}.(\overrightarrow{y\circ_z(f'\circ g')})}\langle\overrightarrow{^{\mathbf{Cp}_{\overrightarrow{g\circ f},\overrightarrow{f'\circ g'}}\bullet r}}/\vec{y}\rangle]$$
$$\equiv C[\mathsf{c_k}\overrightarrow{\circ_p(g\circ f)}\overrightarrow{\lambda z.\mathbf{Cp}_{\overrightarrow{g\circ f},\overrightarrow{f'\circ g'}}(\overrightarrow{r\circ_z(f'\circ g')})}]$$

It can be adjourned as follows:

$$
\begin{aligned}
& C[\mathbf{Cp}_{\vec{g},\vec{g'}}(\mathbf{Cp}_{\vec{f},\vec{f'}}\mathsf{c_k}\,\overrightarrow{p}\,\overrightarrow{r})] \\
& \longrightarrow_{\iota 2} C[\mathbf{Cp}_{\vec{g},\vec{g'}}(\mathsf{c_k}\,\overrightarrow{\circ_x(f)\{^p/_x\}}\,\overrightarrow{\lambda\vec{z}.(y\circ_z\overrightarrow{(f')})}\langle\overrightarrow{^{\mathbf{Cp}_{\vec{f},\vec{f'}}\bullet r}/_{\vec{y}}}\rangle)] \\
& \equiv C[\mathbf{Cp}_{\vec{g},\vec{g'}}(\mathsf{c_k}\,\overrightarrow{\circ_p(f)}\,\overrightarrow{\lambda\vec{z}.(\mathbf{Cp}_{\vec{f},\vec{f'}}(r\overrightarrow{\circ_z(f')}))})] \\
& \longrightarrow_{\iota 2} C[\mathsf{c_k}\,\overrightarrow{\circ_x(g)\{^{\circ_p(f)}/_x\}}\,\overrightarrow{\lambda\vec{z}.y\overrightarrow{\circ_z(g')}}\langle\overrightarrow{^{\mathbf{Cp}_{\vec{g},\vec{g'}}\bullet(\lambda\vec{z}.(\mathbf{Cp}_{\vec{f},\vec{f'}}(r\overrightarrow{\circ_z(f')})))}/_{\vec{y}}}\rangle] \\
& \equiv C[\mathsf{c_k}\,\overrightarrow{\circ_p(g\circ f)}\,\overrightarrow{\lambda\vec{z}.\mathbf{Cp}_{\vec{g},\vec{g'}}((\lambda\vec{z}.(\mathbf{Cp}_{\vec{f},\vec{f'}}(r\overrightarrow{\circ_z(f')})))\overrightarrow{\circ_z(g')})}] \\
& \longrightarrow_{\beta} C[\mathsf{c_k}\,\overrightarrow{\circ_p(g\circ f)}\,\overrightarrow{\lambda\vec{z}.\mathbf{Cp}_{\vec{g},\vec{g'}}(\mathbf{Cp}_{\vec{f},\vec{f'}}(r\overrightarrow{\circ_z(f'\circ g')}))}] \\
& \longrightarrow_{\chi_\circ} C[\mathsf{c_k}\,\overrightarrow{\circ_p(g\circ f)}\,\overrightarrow{\lambda z.\mathbf{Cp}_{\overrightarrow{g\circ f},\overrightarrow{f'\circ g'}}(r\overrightarrow{\circ_z(f'\circ g')})}]
\end{aligned}
$$

For the other cases, ordinary adjournment method works well. (We will abbreviate in the following $\mathbf{Cp}_{\vec{g},\vec{g'}}(\mathbf{Cp}_{\vec{f},\vec{f'}}t)$ and $\mathbf{Cp}_{\overrightarrow{g\circ f},\overrightarrow{f'\circ g'}}t$ by $L_{\chi_\circ}(t)$ and $R_{\chi_\circ}(t)$):

1. For β-conversion:
   (a) For $C \equiv C[(\lambda x \cdot p[L_{\chi_\circ}(t)])\ q]$, we have

   $$C \longrightarrow_{\chi_\circ} C[(\lambda x \cdot p[R_{\chi_\circ}(t)])\ q] \longrightarrow_{\beta_\rightarrow} C[p[R_{\chi_\circ}(t)]\{^q/_x\}]$$

   In this case, we can adjourn $\chi_\circ$ by

   $$C \longrightarrow_{\beta_\rightarrow} C[p[L_{\chi_\circ}(t)]\{^q/_x\}] \longrightarrow_{\chi_\circ} C[p[R_{\chi_\circ}(t)]\{^q/_x\}]$$

   (b) For $C \equiv C[(\lambda x \cdot p)\ q[L_{\chi_\circ}(t)]]$, we have

   $$C \longrightarrow_{\chi_\circ} C[(\lambda x \cdot p)\ q[R_{\chi_\circ}(t)]] \longrightarrow_{\beta_\rightarrow} C[p\{^{q[R_{\chi_\circ}(t)]}/_x\}]$$

   In this case, we can adjourn $\chi_\circ$ by

   $$C \longrightarrow_{\beta_\rightarrow} C[p\{^{q[L_{\chi_\circ}(t)]}/_x\}] \stackrel{*}{\longrightarrow}_{\chi_\circ} C[p\{^{q[R_{\chi_\circ}(t)]}/_x\}]$$

2. For η-conversion, there is no interesting overlap with $\chi_\circ$-conversion as we have the following facts about a $\chi_\circ$-redex $L_{\chi_\circ}()$:
   - $\mathbf{Cp}_{\vec{f},\vec{f'}}$ and $\mathbf{Cp}_{\vec{g},\vec{g'}}$ are iterators ;
   - and t and $L_{\chi_\circ}(t)$ inhabit an *inductive* type.

Proof (proof of theorem 4). As $\beta\eta\iota\chi_\circ$-conversion is strongly normalizing, it is enough to show that $\beta\eta\iota\chi$-conversion is locally confluent, by Newman's Lemma. As $\beta\eta\iota$- and $\chi_\circ$-conversions are both confluent, the proof is by verification (case analysis) that $\longleftarrow_{\chi_\circ};\longrightarrow_{\beta\eta\iota} \subseteq \stackrel{*}{\longleftarrow}_{\beta\eta\iota\chi_\circ};\stackrel{*}{\longrightarrow}_{\beta\eta\iota\chi_\circ}$.

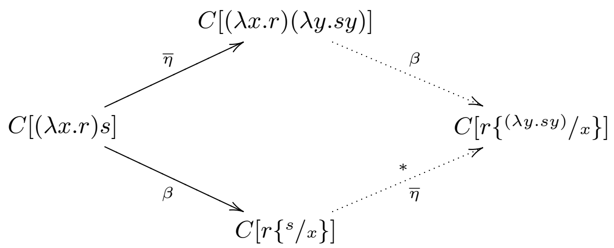
Proof (proof of lemma 2). As T is strongly normalizing, d contains infinitely many $R\backslash T$-reductions (possibly interleaved with finite sequences of T-reductions). As T is insertable in R there exists a relation $\mathcal{S}$ with $(t,t') \in T \subseteq \mathcal{S}$, and we can construct a derivation d' where every reduction $R \setminus T$ of d are repercussed along $\mathcal{S}$ by a R^+-reduction in d'. Hence R^+ contains infinitely many R^+-reductions (possibly interleaved with finite sequences of T^*-reductions) and is therefore infinite

Proof (proof of lemma 3). Given a derivation $d = t \longrightarrow_S \longrightarrow_R \overset{\infty}{\longrightarrow}_{RS}$ beginning with t,

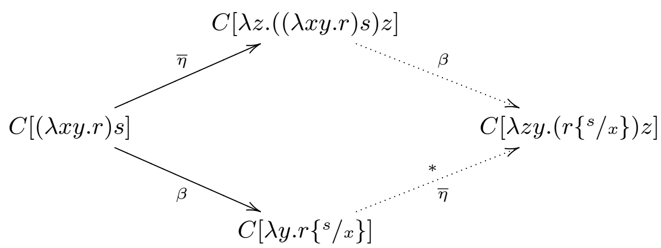
- either $\mathcal{P}(t)$ and then S is adjournable w.r.t R in d
- or as T realizes $\mathcal{P}$, $\exists t', t \rightarrow_T^+ t' \wedge \mathcal{P}(t')$. But T is insertable and strongly normalizing so by lemma 2, there exists an infinite derivation from t'. Hence, as $T \subseteq R$ S is adjournable w.r.t R.

Proof (proof of lemma 5). Take the transitive reflexive closure of $\overline{\eta}$ as relation $\mathcal{S}$.
By lemma 4 is enough to show $\longleftarrow_{\overline{\eta}}; R\backslash \longrightarrow_{\eta} \subseteq R^*; R\backslash \longrightarrow_{\eta}; R^*; \longleftarrow_{\overline{\eta}}^*$ and $\longleftarrow_{\overline{\eta}}; \longrightarrow_{\eta} \subseteq R^*; \longleftarrow_{\overline{\eta}}^*$
The proof is fairly simple and we will only discuss the critical cases (why we take the reflexive transitive closure of $\overline{\eta}$ instead of merely η for $\mathcal{S}$).

- for β, there are two non trivial cases:

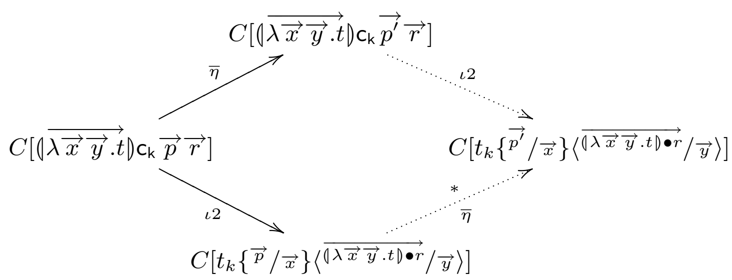


As the subterm s can be substituted in an applicative position, this case justify to take $\overline{\eta}$ instead of merely η for the relation $\mathcal{S}$.

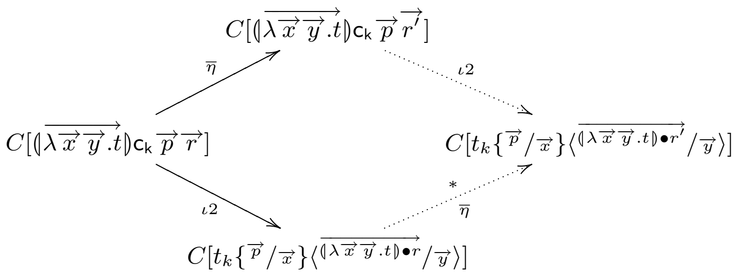


This case illustrates the need for the $\overline{\eta}$-expansion to be applicable to a subterm in abstraction form.

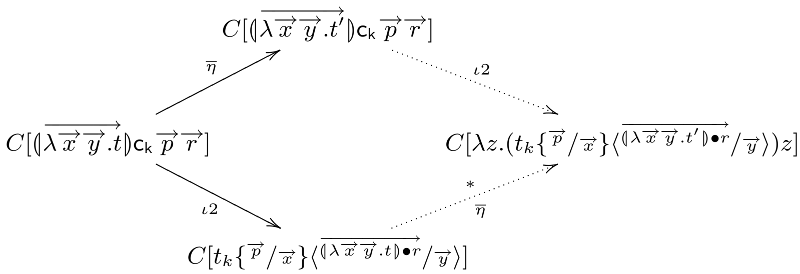
- for $\iota 2$ the situation is similar to β, the $\overline{\eta}$ can take place in a parameter argument: for $\overrightarrow{p} = p_0, \ldots, p_i, \ldots, p_n$ $(i \in [0, \ldots, n])$ we note $\overrightarrow{p'} := p_0, \ldots, \lambda x.p_i x, \ldots, p_n$.



or in a recursive argument: for $\overrightarrow{r} = r_0, \ldots, r_i, \ldots, r_n$ $(i \in [0, \ldots, n])$ we note $\overrightarrow{r'} := r_0, \ldots, \lambda x.r_i x, \ldots, r_n$.



or in the iterator: for $\overrightarrow{t} = t_0, \ldots, t_i, \ldots, t_n$ $(i \in [0, \ldots, n])$ we note $\overrightarrow{t'} := t_0, \ldots, \lambda z.t_i z, \ldots, t_n$. The non trivial case is $i = k$:



Proof (proof of theorem 5). We will abbreviate $\mathbf{Cp}_{\overrightarrow{\mathsf{id}}, \overrightarrow{\mathsf{id}}} t$ by $L_{\chi_{\mathsf{id}}}(t)$

1. For ι-conversion:
   (a) For $C \equiv C[\mathbf{Cp}_{\overrightarrow{\mathsf{id}}, \overrightarrow{\mathsf{id}}}(\mathsf{c}_k\ \overrightarrow{p}\ \overrightarrow{r})]$, by η-insertion we can put the terms $\overrightarrow{r}$ in full externally eta-expansed form. We will write $\overline{r}$ for a term r fully η-expansed externally. We can then always adjourn

   $$C[\mathbf{Cp}_{\overrightarrow{\mathsf{id}}, \overrightarrow{\mathsf{id}}}(\mathsf{c}_k\ \overrightarrow{p}\ \overrightarrow{\overline{r}})] \longrightarrow_{\chi_{\mathsf{id}}} C[\mathsf{c}_k\ \overrightarrow{p}\ \overrightarrow{\overline{r}}] \longrightarrow_{\beta\eta\iota} \cdots$$

   by

   $$\begin{array}{lll} C[\mathbf{Cp}_{\overrightarrow{\mathsf{id}}, \overrightarrow{\mathsf{id}}}(\mathsf{c}_k\ \overrightarrow{p}\ \overrightarrow{\overline{r}})] & \longrightarrow_{\iota 2} & C[(\mathsf{c}_k\, \overrightarrow{x}\, \overrightarrow{\lambda \overrightarrow{z}.y\, \overrightarrow{z}})\{\overrightarrow{p}/\overrightarrow{x}\}\langle \overrightarrow{\mathbf{Cp}_{\overrightarrow{\mathsf{id}}, \overrightarrow{\mathsf{id}}}\bullet \overline{r}}/\overrightarrow{y}\rangle] \\ & \equiv & C[\mathsf{c}_k\, \overrightarrow{p}\, \overrightarrow{\lambda \overrightarrow{z}.\mathbf{Cp}_{\overrightarrow{\mathsf{id}}, \overrightarrow{\mathsf{id}}}(\overline{r}\, \overrightarrow{z})}] \\ & \longrightarrow_{\beta}^{*}\longrightarrow_{\chi_{\mathsf{id}}} & C[\mathsf{c}_k\, \overrightarrow{p}\, \overrightarrow{\overline{r}}] \end{array}$$

   (b) For $C \equiv C[(\!|\ldots, \lambda \overrightarrow{x}\, \overrightarrow{y}.t_r[L_{\chi_{\mathsf{id}}}(s)], \ldots|\!)\ (\mathsf{c_k}\ \overrightarrow{p}\, \overrightarrow{r})]$ and $\overrightarrow{t} \in It(y^{\overrightarrow{\overrightarrow{\sigma_i} \to \tau}})$ (one remarks that $It(y^{\overrightarrow{\overrightarrow{\sigma_i} \to \tau}})$ is stable under $\longleftarrow_{\chi_{\mathsf{id}}}$), we have three possibilities:

i. Either $r \neq k$ and then we have

$$\begin{array}{ccc}
& C[(\!|\ldots,\lambda\overrightarrow{x}\,\overrightarrow{y}.t_r[L_{\chi_{\mathsf{id}}}(s)],\ldots|\!)\ (\mathsf{c_k}\ \ \overrightarrow{p}\,\overrightarrow{r})] & \\
\swarrow_{\chi_{\mathsf{id}}} & & \searrow^{\iota} \\
C[(\!|\ldots,\lambda\overrightarrow{x}\,\overrightarrow{y}.t_r[s],\ldots|\!)\ (\mathsf{c_k}\ \ \overrightarrow{p}\,\overrightarrow{r})] & & C[t_k\ \{\overrightarrow{p}/\overrightarrow{x}\}\ \langle\overrightarrow{((\!|\ldots,\lambda\overrightarrow{x}\,\overrightarrow{y}.t_r[L_{\chi_{\mathsf{id}}}(s)],\ldots|\!)\bullet r_i^R)}/\overrightarrow{y}\rangle] \\
\searrow_{\iota} & & \swarrow^{*}_{\chi_{\mathsf{id}}} \\
& C[t_k\ \{\overrightarrow{p}/\overrightarrow{x}\}\ \langle\overrightarrow{((\!|\ldots,\lambda\overrightarrow{x}\,\overrightarrow{y}.t_r[s],\ldots,|\!)\bullet r_i^R)}/\overrightarrow{y}\rangle] &
\end{array}$$

with as many χ_{id}-conversions as there are variables $\overrightarrow{y}$ occurring in t_k.

ii. or $r = k$, and then
- either $L_{\chi_{\mathsf{id}}}(s)$ is a strict subterm in $t_r[L_{\chi_{\mathsf{id}}}(s)]$:

$$\begin{array}{ccc}
& C[(\!|\ldots,\lambda\overrightarrow{x}\,\overrightarrow{y}.t_r[L_{\chi_{\mathsf{id}}}(s)],\ldots|\!)\ (\mathsf{c_k}\ \ \overrightarrow{p}\,\overrightarrow{r})] & \\
\swarrow_{\chi_{\mathsf{id}}} & & \searrow^{\iota} \\
C[(\!|\ldots,\lambda\overrightarrow{x}\,\overrightarrow{y}.t_r[s],\ldots|\!)\ (\mathsf{c_k}\ \ \overrightarrow{p}\,\overrightarrow{r})] & & \begin{array}{c} C[t_r[L_{\chi_{\mathsf{id}}}(s)]\ \{\overrightarrow{p}/\overrightarrow{x}\} \\ \langle\overrightarrow{((\!|\ldots,\lambda\overrightarrow{x}\,\overrightarrow{y}.t_r[L_{\chi_{\mathsf{id}}}(s)],\ldots|\!)\bullet r_i^R)}/\overrightarrow{y}\rangle] \end{array} \\
\searrow_{\iota} & & \swarrow^{+}_{\chi_{\mathsf{id}}} \\
& C[t_r[s]\ \{\overrightarrow{p}/\overrightarrow{x}\}\ \langle\overrightarrow{((\!|\ldots,\lambda\overrightarrow{x}\,\overrightarrow{y}.t_r[s],\ldots|\!)\bullet r_i^R)}/\overrightarrow{y}\rangle] &
\end{array}$$

  with as many χ_{id}-conversions as there are variable $\overrightarrow{y}$ occurring in t_k g plus one for the $L_{\chi_{\mathsf{id}}}(s)$ occurring already in t_r before $\iota 2$-reduction.
- or $L_{\chi_{\mathsf{id}}}(s)$ is a not strict subterm and is the whole term itself. Let us therefore rewrite the original term as $C[(\!|\ldots,\lambda\overrightarrow{x}\,\overrightarrow{y}.L_{\chi_{\mathsf{id}}}(s),\ldots|\!)\ \mathsf{c_k}\,\overrightarrow{p}\,\overrightarrow{r}]$. Then we have

$$C \longrightarrow_{\chi_{\mathsf{id}}} C[(\!|\ldots,\lambda\overrightarrow{x}\,\overrightarrow{y}.s,\ldots|\!)\ \mathsf{c_k}\,\overrightarrow{p}\,\overrightarrow{r}] \longrightarrow_{\iota 2} C[s]$$

  (because $L_{\chi_{\mathsf{id}}}(s)$ inhabits necessarily an inductive type, therefore can't be of functional type and accept arguments, and hence $\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{p}, \overrightarrow{r}$ are the empty lists). In this case, we can adjourn χ_{id} by

$$C \longrightarrow_{\iota 2} C[L_{\chi_{\mathsf{id}}}(s)] \longrightarrow_{\chi_{\mathsf{id}}} C[s]\ \ .$$

(c) For $C \equiv C[(\!|\ldots,\lambda\overrightarrow{x}\,\overrightarrow{y}.t_k,\ldots|\!)\ (\mathsf{c_k}\ \ \ldots p_r[L_{\chi_{\mathsf{id}}}(s)]\ldots\overrightarrow{r})]$ (writing a for the number of parameter arguments), we have:

$$\begin{array}{rl} C \longrightarrow_{\chi_{\mathsf{id}}} & C[(\!|\ldots,\lambda\overrightarrow{x}\,\overrightarrow{y}.t_k,\ldots|\!)\ (\mathsf{c_k}\ \ \ldots p_r[s]\ldots\overrightarrow{r})] \\ \longrightarrow_{\iota 2} & C[t_k\ \{^{\cdots p_r[s]\cdots}/\overrightarrow{x}\}\ \langle\overrightarrow{((\!|\lambda\overrightarrow{x}\,\overrightarrow{y}.\overrightarrow{t}\,|\!)\bullet r_i)}/\overrightarrow{y}\rangle\ \ . \end{array}$$

In this case, we can adjourn χ_{id} by

$$C \longrightarrow_{\iota 2} C[t_k \ \{^{\cdots p_r[L_{\chi_{\mathsf{id}}}(s)]\cdots}/_{\overrightarrow{x}}\} \ \langle^{\overrightarrow{(\llparenthesis \lambda \overrightarrow{x}\, \overrightarrow{y}\,.\, \overrightarrow{t}\, \rrparenthesis \bullet r_i^R)}}/_{\overrightarrow{y}}\rangle$$
$$\longrightarrow_{\chi_{\mathsf{id}}} C[t_k \ \{^{\cdots p_r[s]\cdots}/_{\overrightarrow{x}}\} \ \langle^{\overrightarrow{(\llparenthesis \lambda \overrightarrow{x}\, \overrightarrow{y}\,.\, \overrightarrow{t}\, \rrparenthesis \bullet r_i^R)}}/_{\overrightarrow{y}}\rangle \ .$$

(d) For $C \equiv C[\llparenthesis \ldots, \lambda \overrightarrow{x}\, \overrightarrow{y}.t_k, \ldots \rrparenthesis \ (\mathsf{c_k} \ \overrightarrow{p} \ldots r_r[L_{\chi_{\mathsf{id}}}(s)] \ldots)]$ (writing a for the number of recursive arguments), we have:

$$C \longrightarrow_{\chi_{\mathsf{id}}} C[\llparenthesis \ldots, \lambda \overrightarrow{x}\, \overrightarrow{y}.t_k, \ldots \rrparenthesis \ (\mathsf{c_k} \ \overrightarrow{p} \ldots r_r[L_{\chi_{\mathsf{id}}}(s)] \ldots)]$$
$$\longrightarrow_{\iota 2} C[t_k \ \{^{\overrightarrow{p}}/_{\overrightarrow{x}}\} \ \langle^{\cdots(\llparenthesis \lambda \overrightarrow{x}\, \overrightarrow{y}\,.\, \overrightarrow{t}\, \rrparenthesis \bullet r_r[s]) \ \cdots}/_{\overrightarrow{y}}\rangle \ .$$

In this case, we can adjourn χ_{id} by

$$C \longrightarrow_{\iota 2} C[t_k \ \{^{\overrightarrow{p}}/_{\overrightarrow{x}}\}\langle^{\cdots(\llparenthesis \lambda \overrightarrow{x}\, \overrightarrow{y}\,.\, \overrightarrow{t}\, \rrparenthesis \bullet r_r[L_{\chi_{\mathsf{id}}}(s)]) \ \cdots}/_{\overrightarrow{y}}\rangle$$
$$\longrightarrow_{\chi_{\mathsf{id}}} C[t_k \ \{^{\overrightarrow{p}}/_{\overrightarrow{x}}\} \ \langle^{\cdots(\llparenthesis \overrightarrow{t}\, \rrparenthesis \bullet r_r[s]) \ \cdots}/_{\overrightarrow{y}}\rangle \ .$$

2. For β-conversion:
   (a) For $C \equiv C[(\lambda x{\cdot}p[L_{\chi_{\mathsf{id}}}(t)]) \ q]$, we have $C \longrightarrow_{\chi_{\mathsf{id}}} C[(\lambda x{\cdot}p[t]) \ q] \longrightarrow_{\beta_\to} C[p[t]\{^q/_x\}]$. In this case, we can adjourn χ_{id} by $C \longrightarrow_{\beta_\to} C[p[L_{\chi_{\mathsf{id}}}(t)]\{^q/_x\}] \longrightarrow_{\chi_{\mathsf{id}}} C[p[t]\{^q/_x\}]$.
   (b) For $C \equiv C[(\lambda x{\cdot}p) \ q[L_{\chi_{\mathsf{id}}}(t)]]$, we have $C \longrightarrow_{\chi_{\mathsf{id}}} C[(\lambda x{\cdot}p) \ q[t]] \longrightarrow_{\beta_\to} C[p\{^{q[t]}/_x\}]$. In this case, we can adjourn χ_{id} by $C \longrightarrow_{\beta_\to} C[p\{^{q[L_{\chi_{\mathsf{id}}}(t)]}/_x\}] \overset{*}{\longrightarrow}_{\chi} C[p\{^{q[t]}/_x\}]$.
3. For η-conversion, there is no interesting overlap with χ_{id}-conversion as we have the following facts about a χ_{id}-redex $L_{\chi_{\mathsf{id}}}(t)$:
   - $L_{\chi_{\mathsf{id}}}$ is a recursor ;
   - and t and $L_{\chi_{\mathsf{id}}}(t)$ inhabit an *inductive* type.
4. For $\chi_\circ$-conversion The only non trivial case is:

$$C[\mathbf{Cp}_{\overrightarrow{f},\overrightarrow{f'}}(\mathbf{Cp}_{\overrightarrow{\mathsf{id}},\overrightarrow{\mathsf{id}}}\,t)] \longrightarrow_{\chi_{\mathsf{id}}} C[\mathbf{Cp}_{\overrightarrow{f},\overrightarrow{f'}}(t)]$$

In this case we can adjourn by

$$C[\mathbf{Cp}_{\overrightarrow{f},\overrightarrow{f'}}(\mathbf{Cp}_{\overrightarrow{\mathsf{id}},\overrightarrow{\mathsf{id}}}\,t)] \longrightarrow_{\chi_\circ} C[\mathbf{Cp}_{\overrightarrow{f},\overrightarrow{f'}}\,t]$$

Proof (proof of theorem 6). As $\beta\eta\iota\chi$-conversion is strongly normalizing, it is enough, by Newman's Lemma to show that $\beta\eta\iota\chi$-conversion is locally confluent. As $\beta\eta\iota\chi_\circ$- and χ_{id}-conversions are both confluent, The proof is a verification by case analysis that $\longleftarrow_{\chi_{\mathsf{id}}};\longrightarrow_{\beta\eta\iota\chi_\circ} \subseteq \overset{*}{\longrightarrow}_{\beta\eta\iota\chi};\overset{*}{\longleftarrow}_{\beta\eta\iota\chi}$.