Everywhere in this notes we will consider a lambda term as a tree.

1 Head Linear Reduction

This section introduces the *head linear reduction* (**HLR**) in a way that is similar to [?'Danos]. The head linear reduction is a reduction stratagy that performs a linear substitution by replacing a so called *head variable occurrence* (*hoc*) on each step. *hoc* can be found as a leaf on the left-most path in a tree. If this leaf is not a variable or represents a free variable then **HLR** get stuck and results in a *quasi-head normal form* (*ghn*).

To be more concrete let's present a HLR as the following **transition system**:

- A state is a tuple $\langle\ A[\underline{B}];\,\Gamma;\,\Delta\ \rangle$, where
  - $A[\underline{B}]$ is a λ-term A with an underline node B
  - Γ is an *environment (binding set)* that binds variables; an environment can be considered as a *set*; we will denote environment as follows:
    $$\Gamma\ =\ \{var \mapsto (t,\ \Gamma_1),\ \ldots\}$$
    where *var* is a variable, t is a λ-term, Γ_1 is a stored environment;
    NB : we are considered Γ as a *disjoint ordered set*
  - Δ (*pending applies*) is a **stack** of pairs $(t_1,\ \Gamma_1)$ where t_1 is a λ-term and Γ_1 is an environment
    $$\text{Denotation: } \Delta\ =\ [(t_1,\ \Gamma_1)\bullet\Delta_1]$$
    where Δ_1 is a rest of the stack Δ
- The *initial* state is $\langle\ \lambda - \textit{term with underlined root};\ \emptyset;\ [\ ]\ \rangle$ where $\emptyset$ denotes an empty set and $[\ ]$ — empty stack
- The *final* state is $\langle\ A[\underline{x}];\ \Gamma;\ \Delta\ \rangle$ where
  - $A[\underline{x}]$ is a λ-term with a variable x as an underlined node
  - $x \notin \Gamma$
- Denotation: $A_{\cancel{B}}[\cancel{\lambda x}\ .\ e]$ — is a term A where subterm B and a node "λx" are crossed over

1.1 Rules

$$\langle A\,[e_1\underline{@}e_2]\,;\ \Gamma;\ \Delta\rangle \longrightarrow \langle A\,[\underline{e_1}@e_2]\,;\ \Gamma;\ [(e_2,\Gamma)\bullet\Delta]\rangle \qquad [\textsc{App}]$$

$$\langle A\,[\underline{\lambda x}.e]\,;\ \Gamma;\ []\rangle \longrightarrow \langle A\,[\lambda x.\underline{e}]\,;\ \Gamma;\ []\rangle \qquad [\textsc{Lam-Non-Elim}]$$

$$\langle A\,[\underline{\lambda x}.e]\,;\ \Gamma;\ [(B,\Gamma')\bullet\Delta]\rangle \longrightarrow \langle A_{\cancel{B}}[\cancel{\lambda x}.\underline{e}]\,;\ \{x\mapsto(B,\Gamma'),\,\Gamma\};\ \Delta\rangle \qquad [\textsc{Lam-Elim}]$$

I.e. : variable x becomes bound by Γ with the top element of the stack of pending applies (B,Γ')

$$\langle A\,[\underline{x}]\,;\ \{x\mapsto(B,\Gamma'),\,\Gamma\};\ \Delta\rangle \longrightarrow \langle A\,[\underline{B}]\,;\ \Gamma';\ \Delta\rangle \qquad [\textsc{BVar}]$$

1.1.1 Example

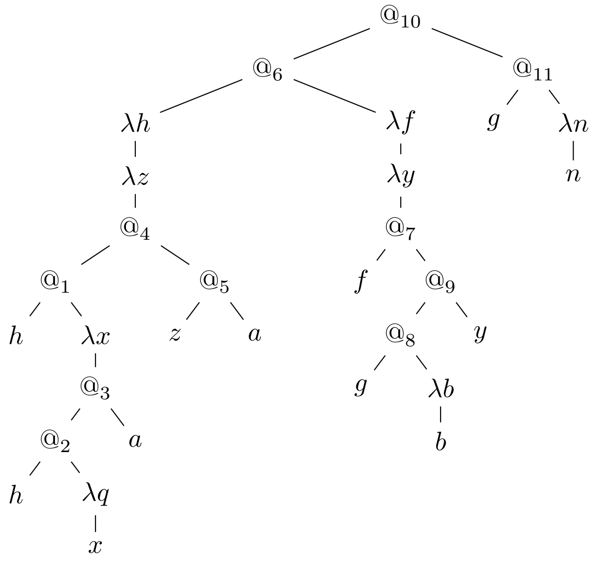
Consider the following input term $(\lambda\ x\ .\ x)\ @\ (\lambda\ y\ .\ y)$.

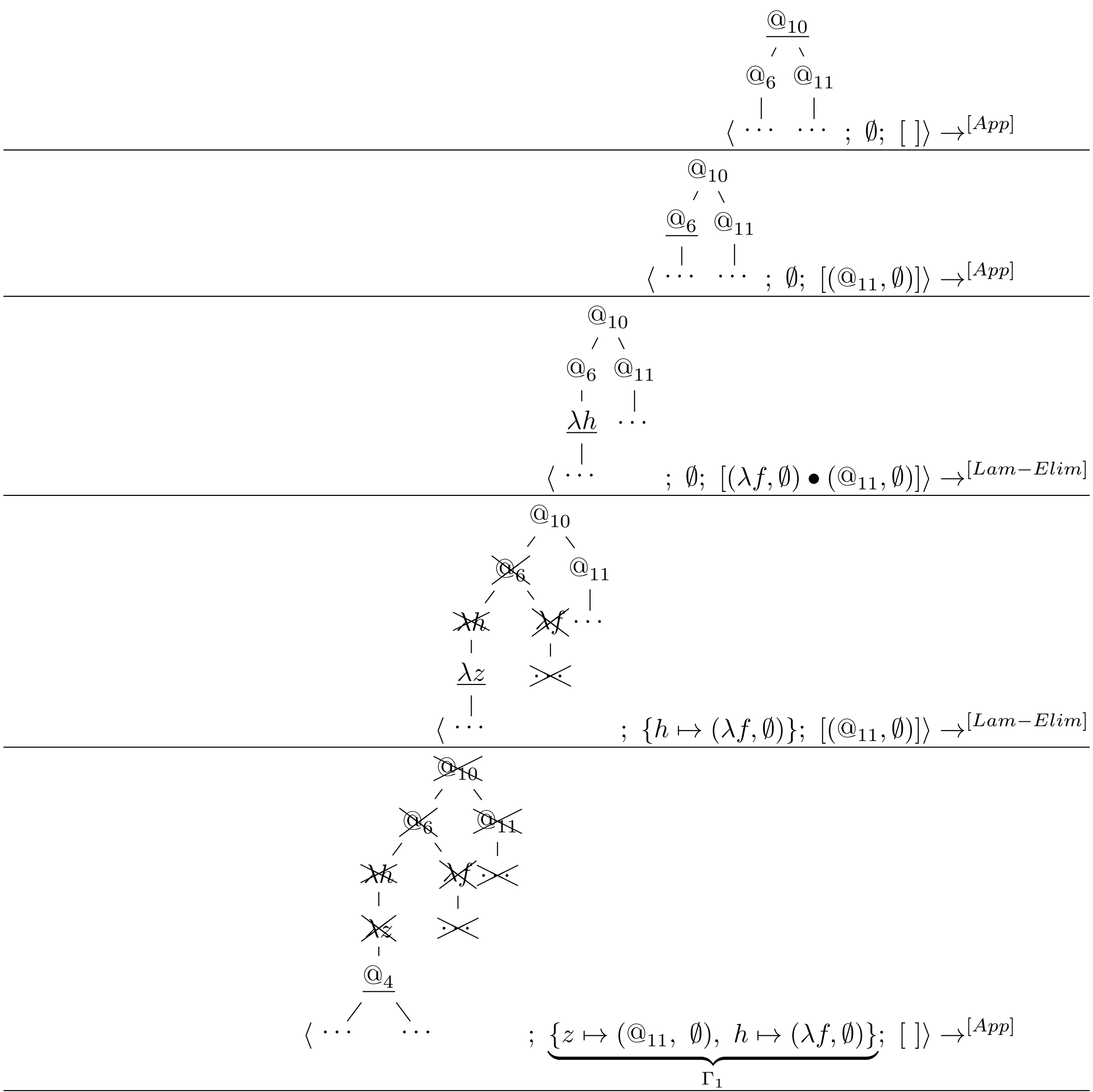
$$\begin{aligned}
&\langle(\lambda\ x\ .\ x)\ \underline{@}\ (\lambda\ y\ .\ y);\ \emptyset; && [\ ]\rangle \rightarrow^{[App]} && (1)\\
&\langle(\underline{\lambda\ x}\ .\ x)\ @\ (\lambda\ y\ .\ y);\ \emptyset; && [((\lambda\ y\ .\ y),\ \emptyset)]\rangle \rightarrow^{[Lam-Elim]} && (2)\\
&\langle(\cancel{\lambda x}\ .\ \underline{x})\ \cancel{@}\ \cancel{(\lambda\ y\ .\ y)};\ \{x\mapsto((\lambda\ y\ .\ y),\ \emptyset)\}; && [\ ]\rangle \rightarrow^{[BVar]} && (3)\\
&\langle(\cancel{\lambda x}\ .\ \underline{\lambda\ y}\ .y)\ \cancel{@}\ \cancel{(\lambda\ y\ .\ y)};\ \emptyset; && [\ ]\rangle \rightarrow^{[Lam-Non-Elim]} && (4)\\
&\langle(\cancel{\lambda x}\ .\ \lambda\ y\ .\underline{y})\ \cancel{@}\ \cancel{(\lambda\ y\ .\ y)};\ \emptyset; && [\ ]\rangle \not\rightarrow && (5)
\end{aligned}$$

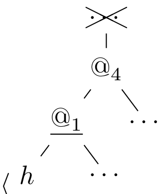
1.1.2 Example: $\langle\ N\ P\ R\ \rangle$

$$
\begin{aligned}
N &= \lambda h\ .\ \lambda z\ .\ h\ @\ (\lambda x\ .\ (h\ @\ (\lambda q\ .\ x)\ @\ a))\ @\ (z\ @\ a)\\
P &= \lambda f\ .\ \lambda y\ .\ f\ @\ ((g\ @\ (\lambda b\ .\ b))\ @\ y)\\
R &= g\ @\ (\lambda n\ .\ n)
\end{aligned}
$$

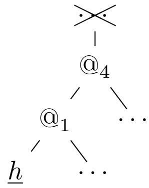
$$
\begin{aligned}
N\ P\ R\ &= ((\lambda h\ \lambda z\ .\ (h\ @_1\ (\lambda x\ .\ ((h\ @_2\ (\lambda q\ .\ x))\ @_3\ a)))\ @_4\ (z\ @_5\ a))\ @_6\ (\lambda f\ \lambda y\ .\ f\ @_7\ ((g\ @_8\ (\lambda h\ .\ h))\ @_9\ y)))\ @_{10}\\
&\quad (g\ @_{11}\ (\lambda n\ .\ n))
\end{aligned}
$$



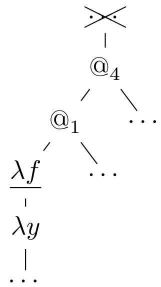




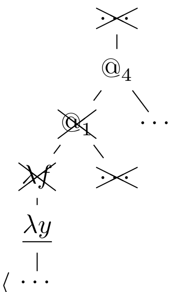
$\langle \ ; \ \Gamma_1; \ [(@_5, \ \Gamma_1)]\rangle \to^{[App]}$



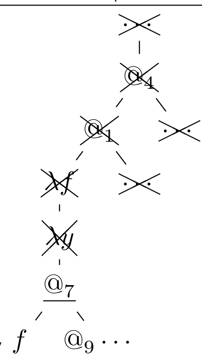
$\langle \ ; \ \Gamma_1; \ [(\lambda x, \ \Gamma_1) \bullet (@_5, \ \Gamma_1)]\rangle \to^{[BVar]}$



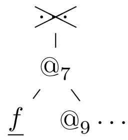
$\langle \ ; \ \emptyset; \ [(\lambda x, \ \Gamma_1) \bullet (@_5, \ \Gamma_1)]\rangle \to^{[Lam-Elim]}$



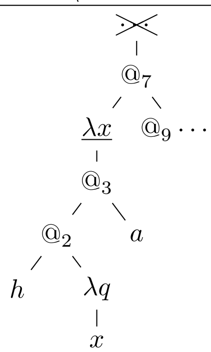
$\langle \ ; \ \{f \mapsto (\lambda x, \ \Gamma_1)\}; \ [(@_5, \ \Gamma_1)]\rangle \to^{[Lam-Elim]}$



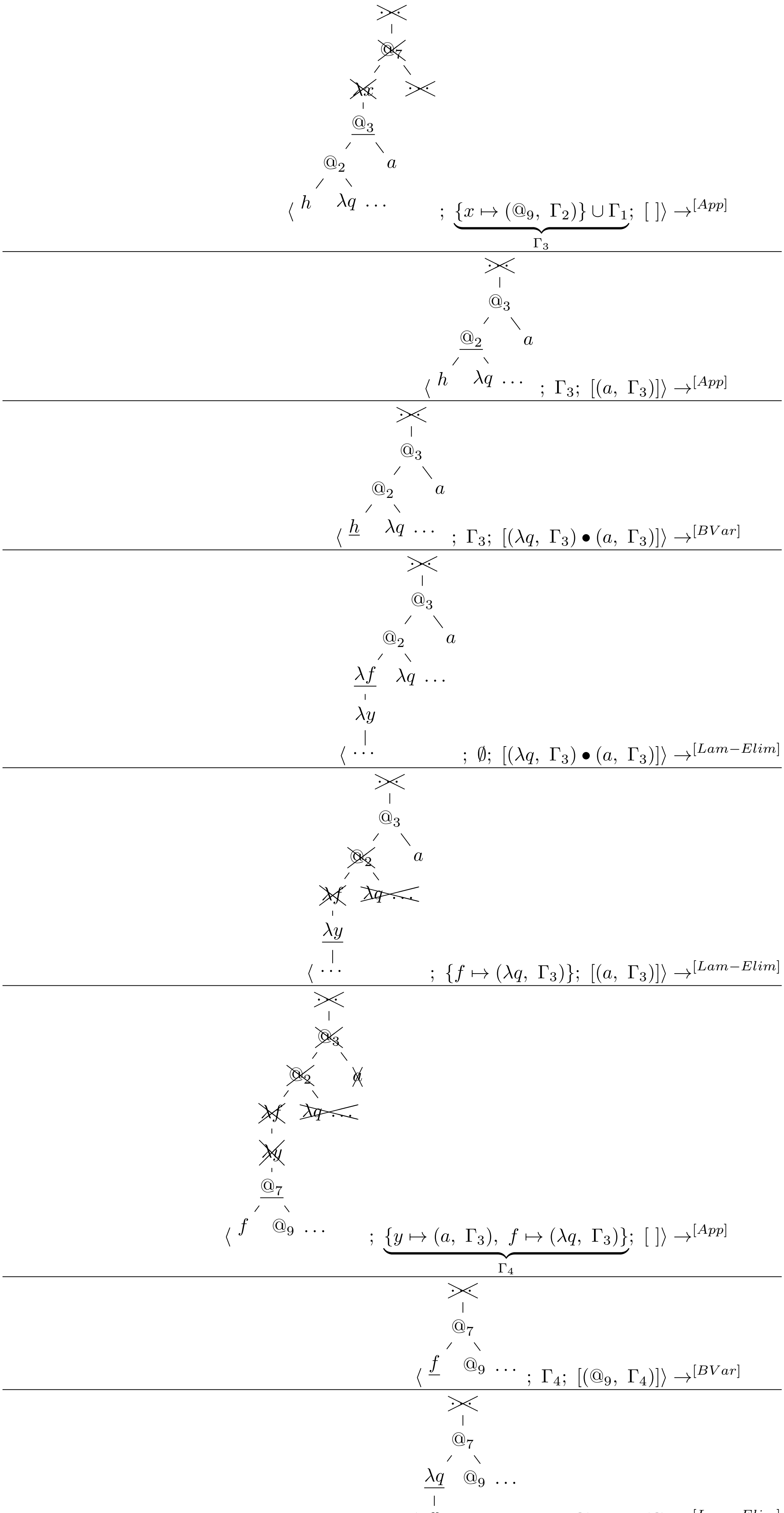
$\langle \ ; \ \underbrace{\{y \mapsto (@_5, \ \Gamma_1), \ f \mapsto (\lambda x, \ \Gamma_1)\}}_{\Gamma_2}; \ [\ ]\rangle \to^{[App]}$

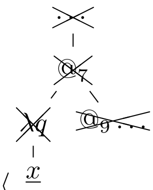


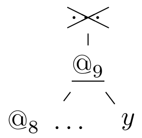
$\langle \ ; \ \Gamma_2; \ [(@_9, \ \Gamma_2)]\rangle \to^{[BVar]}$

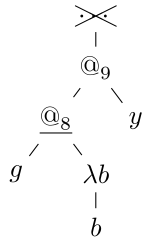


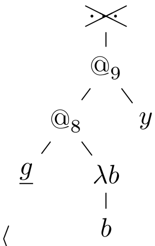
$\langle \ ; \ \Gamma_1; \ [(@_9, \ \Gamma_2)]\rangle \to^{[Lam-Elim]}$


; {x ↦ (@9, Γ2)} ∪ Γ1; [ ]⟩ →[App]
Γ3
; Γ3; [(a, Γ3)]⟩ →[App]
; Γ3; [(λq, Γ3) • (a, Γ3)]⟩ →[BVar]
; ∅; [(λq, Γ3) • (a, Γ3)]⟩ →[Lam−Elim]
; {f ↦ (λq, Γ3)}; [(a, Γ3)]⟩ →[Lam−Elim]
; {y ↦ (a, Γ3), f ↦ (λq, Γ3)}; [ ]⟩ →[App]
Γ4
; Γ4; [(@9, Γ4)]⟩ →[BVar]
; Γ3; [(@9, Γ4)]⟩ →[Lam−Elim]

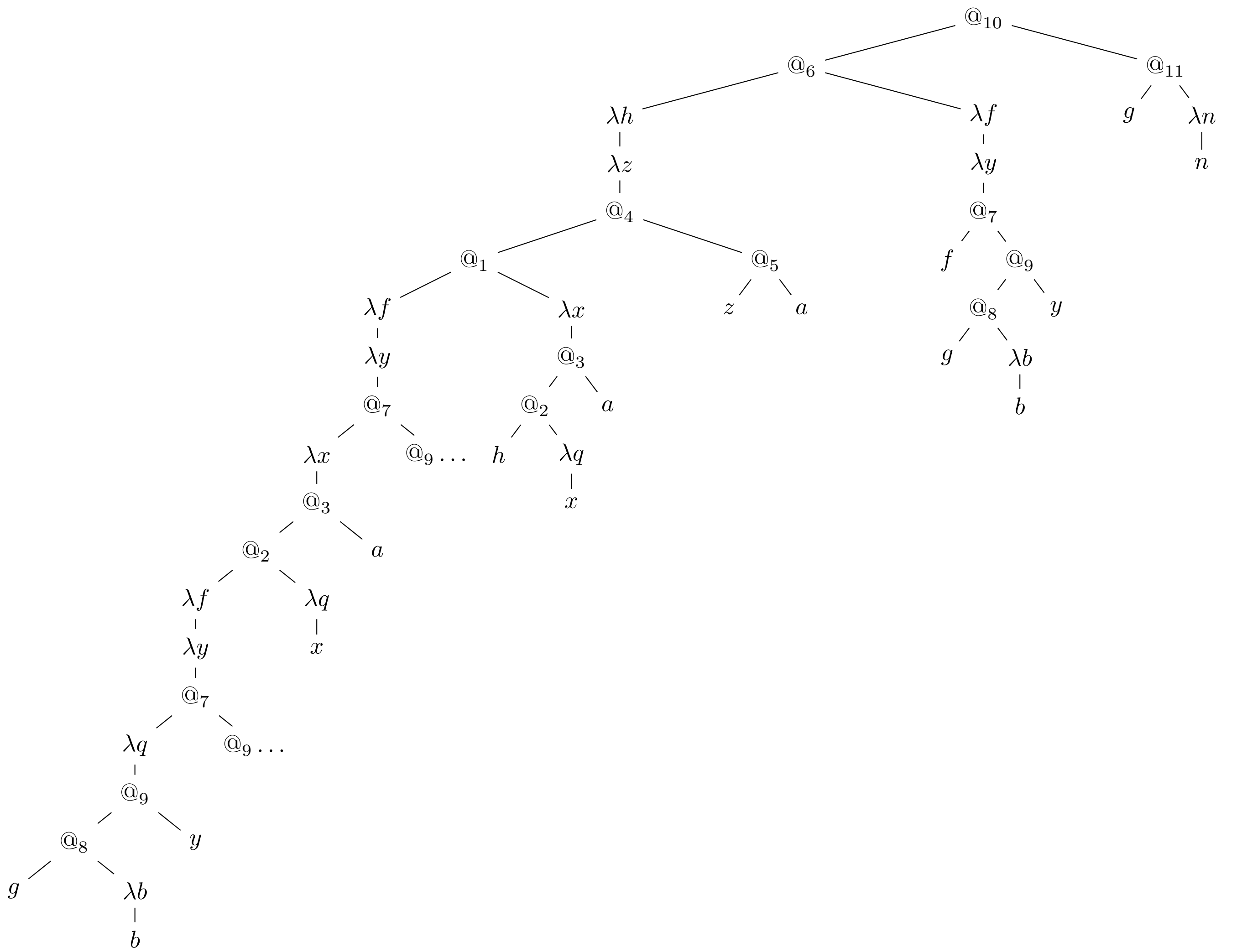
$\langle$  $;\ \{q \mapsto (@_9,\ \Gamma_4)\} \cup \Gamma_3;\ [\ ]\rangle \rightarrow^{[BVar]}$

$\langle$  $;\ \Gamma_2;\ [\ ]\rangle \rightarrow^{[App]}$

$\langle$  $;\ \Gamma_2;\ [(y,\ \Gamma_2)]\rangle \rightarrow^{[App]}$

$\langle$  $;\ \Gamma_2;\ [(\lambda b,\ \Gamma_2) \bullet (y,\ \Gamma_2)]\rangle \not\rightarrow$

NB: Note that the result of the head linear reduction (qhn form) is the first component of the final state (**elements that are crossed over also appears in qhn**). I.e. the result of the example under consideration is as follows:



1.2 Correctness proof

In this section correctness with respect to the **head reduction** is presented. To perform that we will define an auxiliary function "*expansion (exp)*".

Expansion function. *exp* by a given transition system state provides a λ-term. Informally speaking function *exp* performs head substition of all variables that are presented in the environments.

$$exp\ \langle M[\underline{A}];\ \{\Gamma,\ x \mapsto (B, \Gamma')\};\ \Delta\rangle = exp\ \langle M[\underline{A}[x/B[\Gamma']]];\ \Gamma;\ \Delta\rangle \tag{6}$$

$$exp\ \langle M_B[\underline{A}];\ \emptyset;\ [(B, \Gamma') \bullet \Delta]\rangle = exp\ \langle M_{B[\Gamma']}[\underline{A}];\ \emptyset;\ \Delta\rangle \tag{7}$$

$$exp\ \langle M;\ \emptyset;\ [\ ]\rangle = M' \tag{8}$$

$$\text{where } B[\Gamma'] = exp\ \langle \underline{B};\ \Gamma';\ [\ ]\rangle \tag{9}$$

$$M' \text{ is a term } M \text{ where all nodes that are crossed over are crosed out} \tag{10}$$

Note that (6) substitutes term $B[\Gamma']$ instead of all occurrence of variable x in the subtree of term M that has an underlined node as a root (i.e. subtree A). Each recursive call of (7) substitutes all variables with respect to the corresponding context exactly in one argument term (since it is assumed that all variables have diffenert names and substition is performed in only in arguments with respect to an environment no confusion may appear (no renaming is needed)).

Now consider TS for HLR. Note that:

- Only $[Lam - Elim]$ rule can change an expansion;
- The number of transitions without applying $[Lam - Elim]$ rule is limited since in the definition of context, the input term has a finite size, and only $[Lam - Elim]$ rule expands the context; Hereafter we will denote a sequence of transitions without applying $[Lam - Elim]$ rule by $\xrightarrow{*}$;
- The expansion function *exp* cannot change a path from the root of term to the underlined node since:
  - recursive call (6) can only change the subtree of the current term with underlined node as a root; **(I)**
  - recursive call (7) can only change arguments of the current term that are above the underlined node; Moreover, each application of (7) changes exactly one argument (that corresponds to the first element of the pair in Δ) leaving all other arguments and the subtree of the current term with underlined node as a root intact. **(II)**

1.2.1 Example

$$\begin{aligned}
exp((1)) &= exp(\langle(\lambda\ x\ .\ x)\ \underline{@}\ (\lambda\ y\ .\ y);\ \emptyset;\ [\ ]\rangle) = (\lambda\ x\ .\ x)\ @\ (\lambda\ y\ .\ y)\\
exp((2)) &= exp(\langle(\underline{\lambda\ x}\ .\ x)\ @\ (\lambda\ y\ .\ y);\ \emptyset;\ [((\lambda\ y\ .\ y),\ \emptyset)]\rangle)\ = exp(\langle(\underline{\lambda\ x}\ .\ x)\ @\ (\lambda\ y\ .\ y);\ \emptyset;\ [\ ]\rangle)\\
&= (\lambda\ x\ .\ x)\ @\ (\lambda\ y\ .\ y)\\
exp((3)) &= exp(\langle(\cancel{\lambda x}\ .\ \underline{x})\ \cancel{@}\ \cancel{(\lambda\ y\ .\ y)};\ \{x \mapsto ((\lambda\ y\ .\ y),\ \emptyset)\};\ [\ ]\rangle)\\
&= exp(\langle(\cancel{\lambda x}\ .\ \underline{\lambda\ y}\ .\ y)\ \cancel{@}\ \cancel{(\lambda\ y\ .\ y)};\ \emptyset;\ [\ ]\rangle)\ =\ \lambda\ y\ .\ y\\
exp((4)) &= exp(\langle(\cancel{\lambda x}\ .\ \underline{\lambda\ y}\ .y)\ \cancel{@}\ \cancel{(\lambda\ y\ .\ y)};\ \emptyset;\ [\ ]\rangle)\ =\ \lambda\ y\ .\ y\\
exp((5)) &= exp(\langle(\cancel{\lambda x}\ .\ \lambda\ y\ .\underline{y})\ \cancel{@}\ \cancel{(\lambda\ y\ .\ y)};\ \emptyset;\ [\ ]\rangle)\ =\ \lambda\ y\ .\ y
\end{aligned}$$

Correspondence with respect to Head Reduction (Informally; formally see in the following sections)
Expansion cannot be changed by any TS rules except $[Lam - Elim]$ rule. Thus, it is easy to see the correspondence between head linear reduction and usual head reduction. Each application of $[Lam - Elim]$ rule performs exactly one step of head reduction after applying the *exp*-function.

$$(\lambda\ x\ .\ x)\ @\ (\lambda\ y\ .\ y) \rightarrow \text{corresponds to first to steps}$$
$$\lambda\ y\ .\ y \text{ corresponds to last three steps}$$

1.2.2 Theorem

HLR and HR coincide as follows:

Suppose that we have some current TS state (denoted $\langle\dots\rangle$) with some expansion (denoted $\dots$) such that the next rule to be applied is $[Lam - Elim]$ and results in state $\langle M_i;\ \Gamma_i;\ \Delta_i\rangle$ whose expansion is some term M'_i. Then TS make several new steps without applying $[Lam - Elim]$ rule and results in some state $\langle A[\underline{\lambda x}.e];\ \Gamma;\ (B, \Gamma') \bullet \Delta\rangle$ whose expansion is still term M'_i. Then ***If*** $[Lam - Elim]$ *rule can be applied by extending environment* Γ *with new binding* $x \mapsto (B, \Gamma')$ *and expansion of the result is some term* M'_{i+1} ***then*** M'_{i+1} *can be obtained by one step of head reduction from the term* M'_i.

$$
\begin{tikzcd}
\langle \ldots \rangle \arrow[r, "{[Lam-Elim]}"] \arrow[d, "exp"] & \langle M_i;\ \Gamma_i;\ \Delta_i \rangle \arrow[r, "*"] \arrow[d, "exp"] & \langle A[\underline{\lambda x}.e];\ \Gamma;\ (B,\Gamma')\bullet\Delta \rangle \arrow[r, "{[Lam-Elim]}"] \arrow[dl, "exp"] & \langle A_{\cancel{B}}[\cancel{\lambda x}.\underline{e}];\ x \mapsto (B,\Gamma'),\Gamma;\ \Delta \rangle \arrow[d, "exp"] \\
\ldots \arrow[r, "HR"] & M'_i \arrow[rr, dotted, "HR"] & & M'_{i+1}
\end{tikzcd}
$$

Proof by induction on number of $[Lam - Elim]$ steps.

Base: trivial since first element being added to Γ is a head redex by definition.

Induction step. We know that after i^{th} application of rule $[Lam - Elim]$ we got some TS state $\langle M_i;\ \Gamma_i;\ \Delta_i \rangle$ whose expansion is some term M'_i. As was mentioned above $\overset{*}{\to}$ has not changed the expansion and the number of steps in $\overset{*}{\to}$ is finite. Thus, we have two possible cases:

first, TS got stuck and nothing to prove,

and second, we can apply the $[Lam - Elim]$ rule. In the second case we know the form of the state (following the $[Lam - Elim]$ definition), i.e. $\langle A[\underline{\lambda x}.e];\ \Gamma;\ (B,\Gamma')\bullet\Delta \rangle$ (let us denote it (i)), and by assumption the result of applying $[Lam - Elim]$ rule is a state $\langle A_{\cancel{B}}[\cancel{\lambda x}.\underline{e}];\ x \mapsto (B,\Gamma'),\Gamma;\ \Delta \rangle$ (let us denote it (ii)). Now we have to show that applying function *exp* to that state results in the same term as a step of head reduction from term M'_i. (i.e. in the term M'_{i+1})

Let see what will happend if we apply *exp* function to state (i):

$$
\begin{aligned}
& exp\ \langle A[\underline{\lambda x}.\ e];\ \Gamma;\ [(B,\Gamma')\bullet\Delta] \rangle && (11)\\
& \overset{exp}{\to}{}^{*}\ exp\ \langle A'[\underline{\lambda x}.\ e[\Gamma]];\ \emptyset;\ [(B,\Gamma')\bullet\Delta] \rangle \quad \text{by fully applying (6)} && (12)\\
& \overset{exp}{\to}\ exp\ \langle A'_{B[\Gamma']}[\underline{\lambda x}.\ e[\Gamma]];\ \emptyset;\ \Delta \rangle \quad \text{by one step of (7)} && (13)\\
& \overset{exp}{\to}\ \ldots && (14)
\end{aligned}
$$

to state (ii):

$$
\begin{aligned}
& exp\langle A_{\cancel{B}}[\cancel{\lambda x}.\ \underline{e}];\ \{x \mapsto (B,\Gamma'),\Gamma\};\ \Delta \rangle && (15)\\
& \overset{exp}{\to}{}^{*}\ exp\ \langle A_{\cancel{B}}[\cancel{\lambda x}.\ \underline{e}[\Gamma]];\ \{x \mapsto (B,\Gamma')\};\ \Delta \rangle \quad \text{by elliminating } \Gamma \text{ following (6)} && (16)\\
& \overset{exp}{\to}\ exp\ \langle A_{\cancel{B}}[\cancel{\lambda x}.\ \underline{e}[\Gamma][x/B[\Gamma']]];\ \emptyset;\ \Delta \rangle \quad \text{by one step of (6)} && (17)\\
& \overset{exp}{\to}\ \ldots && (18)
\end{aligned}
$$

It is easy to see that the term in state in (13), $A'_{B[\Gamma']}[\underline{\lambda x}.\ e[\Gamma]]$, has the head redex $(\lambda x,\ B)$ (since context is already empty). Thus, if we apply head reduction to this term, it will result exactly in the term $A_{\cancel{B}}[\cancel{\lambda x}.\ \underline{e}[\Gamma][x/B[\Gamma']]]$ (by head reduction definition) that is equal to the first component of state (17). Following **(I)** and **(II)**, if we will continue to apply *exp* function then elliminating the same Δ in both cases will lead no effect on argument B and underlined subtrees. Thus, $(\lambda x,\ B)$ is a head redex in M'_i and the expansion of state (ii) is M'_{i+1}. **Qed**

Consequences:

- if head linear reduction terminates then head reduction terminates;
- if head reduction terminates then head linear reduction terminates;
- Since expansion is not able to change a path from the root to the underlined node, the first component of the final state of TS for head linear reduction contains a term whose leftmost path is a part of the head normal form and the expansion of the final state is a node of the Boehm-tree of the input term;
  Thus, a repeated application of head linear reduction to all arguments (complete head linear reduction) leeds to the normal form.

2 Complete Head Linear Reduction

Complete Head Linear Reduction (CHLR) can be seen as usual head linear reduction that is repeatedly applied to the all arguments. In this section transition system for CHLR is presented. It is a simple extension of TS for HLR.

2.1 Transition System for Complete Head Linear Reduction

To perform serial application of HLR in all arguments three new rules $[FVar - *]$ are introduced. Informally, these three rules handle the situation when HLR results is some state with first component be a term with underlined unbouned (or free) variable.

- Stack of pairs Δ is extended by special marker \$;
- *Initial* state is $\langle \lambda - term\ with\ underlined\ root;\ \emptyset;\ [\,] \rangle$;
- *Final* state is $\langle M\,[\underline{x}]\,;\ \Gamma;\ [\,] \rangle$ where $x \notin dom(\Gamma)$;
- Changes (differences between TS for CHLR and TS for HLR) are colored;
- The result of CHLR is a term in normal form; it can be obtained from the first component of a final state by crossing out all element that are crossed over.

$$\langle A\,[e_1 \underline{@} e_2]\,;\ \Gamma;\ \Delta \rangle \longrightarrow \langle A\,\left[\underline{e_1} @ e_2\right]\,;\ \Gamma;\ [(e_2, \Gamma) \bullet \Delta] \rangle \quad [\textsc{App}]$$

$$\langle A\,[\underline{\lambda x}.e]\,;\ \Gamma;\ [\$ \bullet \Delta] \rangle \longrightarrow \langle A\,[\lambda x.\underline{e}]\,;\ \Gamma;\ [\$ \bullet \Delta] \rangle \quad [\textsc{Lam-Non-Elim}]$$

$$\langle A\,[\underline{\lambda x}.e]\,;\ \Gamma;\ [(B, \Gamma') \bullet \Delta] \rangle \longrightarrow \langle A_{\cancel{B}}\,[\cancel{\lambda x}.\underline{e}]\,;\ \{x \mapsto (B, \Gamma'),\, \Gamma\};\ \Delta \rangle \quad [\textsc{Lam-Elim}]$$

$$\langle A\,[\underline{x}]\,;\ \{x \mapsto (B, \Gamma'),\, \Gamma\};\ \Delta \rangle \longrightarrow \langle A\,[\underline{B}]\,;\ \Gamma';\ \Delta \rangle \quad [\textsc{BVar}]$$

$$\langle A\,[M\,[\underline{x}]\,@B]\,;\ \Gamma;\ [(B, \Gamma') \bullet \$ \bullet \Delta] \rangle \longrightarrow \langle A\,[M[x] @ \underline{B}]\,;\ \Gamma';\ [\$ \bullet \Delta] \rangle,\ x \notin dom(\Gamma) \quad [\textsc{FVar-0}]$$

Note that B here (and in all $[FVar - *] rules$) comes from the list of pending applies

It is an invariant that B has to be an argument of some application

$$\langle A\,[M\,[\underline{x}]\,@B]\,;\ \Gamma;\ [(B, \Gamma') \bullet C \bullet \Delta] \rangle \longrightarrow \langle A\,[M[x] @ \underline{B}]\,;\ \Gamma';\ [\$ \bullet C \bullet \Delta] \rangle,\ C \neq \$,\ x \notin dom(\Gamma) \quad [\textsc{FVar-1}]$$

$$\langle A\,[M\,[\underline{x}]\,@B]\,;\ \Gamma;\ [\$ \bullet (B, \Gamma') \bullet \Delta] \rangle \longrightarrow \langle A\,[M\,[x]\,@\underline{B}]\,;\ \Gamma';\ \Delta_1 \rangle,\ \ x \notin dom(\Gamma),\ \Delta_1 = \left\{ \begin{array}{l} \Delta, \text{if } \Delta = [\$ \bullet \ldots] \\ [\$ \bullet \Delta], \text{otherwise} \end{array} \right. \quad [\textsc{FVar-2}]$$

2.2 Expansion function

Expansion function for TS for CHLR is a simple extansion of the expansion function for TS state for HLR. Only one new case is added to handle \$ appearence on stack : (21).

$$exp\ \langle M[\underline{A}];\ \{\Gamma,\ x \mapsto (B, \Gamma')\};\ \Delta \rangle = exp\ \langle M[\underline{A}[x/B[\Gamma']]];\ \Gamma;\ \Delta \rangle \tag{19}$$

$$exp\ \langle M_B[\underline{A}];\ \emptyset;\ [(B, \Gamma') \bullet \Delta] \rangle = exp\ \langle M_{B[\Gamma']}[\underline{A}];\ \emptyset;\ \Delta \rangle \tag{20}$$

$$exp\ \langle M_B[\underline{A}];\ \emptyset;\ [\$ \bullet \Delta] \rangle = exp\ \langle M_{B[\Gamma']}[\underline{A}];\ \emptyset;\ \Delta \rangle \tag{21}$$

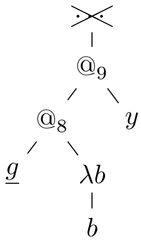
$$exp\ \langle M;\ \emptyset;\ [\,] \rangle = M' \tag{22}$$

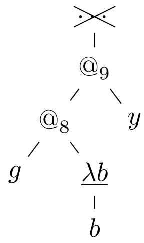
$$\text{where } B[\Gamma'] = exp\ \langle \underline{B};\ \Gamma';\ [\,] \rangle \tag{23}$$

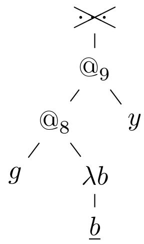
$$M' \text{ is a term } M \text{ where all nodes that are crossed over are crosed out} \tag{24}$$

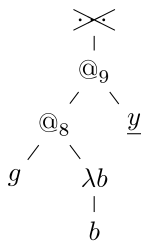
2.3 Example

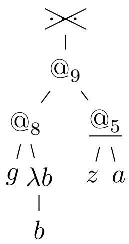
Let us return to example from section 1.1.2. TS for CHLR performs the same steps as TS for HLR with one exception: HLR stucks in qhn form while CHLR continues as follows:

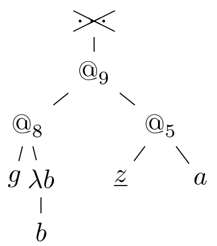
$\langle$  $;\ \Gamma_2;\ [(\lambda b,\ \Gamma_2) \bullet (y,\ \Gamma_2)]\rangle \to^{[FVar-1]}$

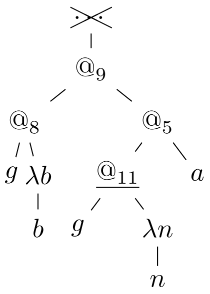
$\langle$  $;\ \Gamma_2;\ [\$ \bullet (y,\ \Gamma_2)]\rangle \to^{[Lam-Non-Elim]}$

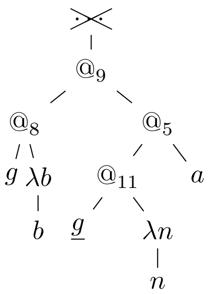
$\langle$  $;\ \Gamma_2;\ [\$ \bullet (y,\ \Gamma_2)]\rangle \to^{[FVar-2]}$

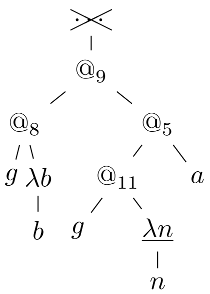
$\langle$  $;\ \Gamma_2;\ [\ ]\rangle \to^{[BVar]}$

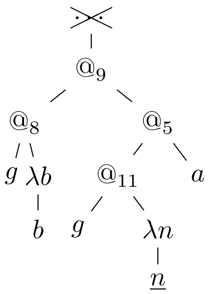
$\langle$  $;\ \Gamma_2;\ [\ ]\rangle \to^{[App]}$

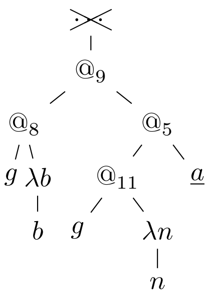
$\langle$  $;\ \Gamma_2;\ [(a,\ \Gamma_2)] \to^{[BVar]}$

$\langle$  $;\ \emptyset;\ [(a,\ \Gamma_2)]\rangle \rightarrow^{[App]}$

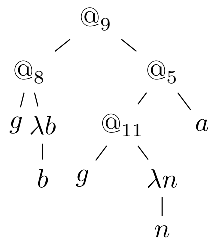
$\langle$  $;\ \emptyset;\ [(\lambda n,\ \emptyset) \bullet (a,\ \Gamma_2)]\rangle \rightarrow^{[FVar-1]}$

$\langle$  $;\ \emptyset;\ [\$ \bullet (a,\ \Gamma_2)]\rangle \rightarrow^{[Lam-Non-Elim]}$

$\langle$  $;\ \emptyset;\ [\$ \bullet (a,\ \Gamma_2)]\rangle \rightarrow^{[FVar-2]}$

$\langle$  $;\ \Gamma_2;\ [\,]\rangle \not\rightarrow$

The result (normal form):



3 Transition System for UNP

This section contains TS for UNP. This TS has the same number of rules that the TS for CHLR. Moreover, it is easy to see that rules with the same name corresponds to each other.

There are two pointer types:

- Binder pointer (denoted as $\rightarrow$); from the bound variables binder pointer points to the dynamic binder in the history and for all other nodes it points points to the parent; (this kind of pointer is omitted in these notes since they can be easly defined by recursive procedure)
- Pointer to the last pending application; there are two subtypes of this pointer:
  - Thick brown pointer can binds either lambda node and application (in this case they arrise as *spine redexes*) or any other node and some application;
  - Vioilet pointer has the same meaning (pointer to the last pending application) together with \$ in TS for CHLR;
  - Dashed red pointer is used when the real colour (brown or violet) is not the metter.

Rules to construct a traversal:

- $(BVar)$

$$\dots\ @\ \dots\ \lambda x\ \dots\ \overset{i}{x} \longrightarrow \text{where } e \text{ is is the right child of } @ \longrightarrow\ \dots\ @\ \dots\ \lambda x\ \dots\ \overset{i}{x}\ =\ e$$

  NB: the colour of the added (red) pointer has to be the same as the colour of the first "red"pointer

- $(Lam - Non - Elim)$

$$\dots\ @\ \dots\ \lambda x \longrightarrow \text{where } e \text{ is the right child of } @ \longrightarrow\ \dots\ @\ \dots\ \lambda x\ \ e$$

- $(Lam - Elim)$

$$\dots\ @_1\ \dots\ @_2\ \dots\ \lambda x \longrightarrow \text{where } e \text{ is a child of } \lambda x \longrightarrow\ \dots\ @_1\ \dots\ @_2\ \dots\ \lambda x\ \ e$$

  NB: the colour of the added (red) pointer has to be the same as the colour of the first "red"pointer

- (App)

$$\dots\ @ \longrightarrow \text{where } e \text{ is the left child of } @ \longrightarrow\ \dots\ @\ e$$

- $(FVar - 0)$

$$\dots\ @_1\ \dots\ @_2\ \dots\ y \longrightarrow \text{where } e \text{ is the right child of } @_2 \longrightarrow\ \dots\ @_1\ \dots\ @_2\ \dots\ y\ \ ||\ \ e$$

- $(FVar - 1)$

$$\dots\ @_1\ \dots\ @_2\ \dots\ y \longrightarrow \text{where } e \text{ is the right child of } @_2 \longrightarrow\ \dots\ @_1\ \dots\ @_2\ \dots\ y\ \ ||\ \ e$$

- $(FVar-2)$

$$\ldots\ @_1\ \ldots\ @_2\ \ldots\ y\ \longrightarrow \text{ where } e \text{ is the right child of } @_2 \longrightarrow\ \ \ldots\ @_1\ \ldots\ @_2\ \ldots\ y\ ||\ e$$

NB: the colour of the "red"pointer is not the metter in this case

Note that Δ and Γ can be easly defined from UNP TS state as follows:

- Δ can be constructed by following pending application pointer and storing right childs of all reachable applications in reverse order with corresponding contexts (see next item how to define context from traversal);
- Γ can be constructed by following binder pointers, and when faced with a lambda–token λx that has a brown pointer to some application @, store $x \mapsto (e, \Gamma')$ where e is the right child of apllication @ and Γ' is a context (which is defined recursively from token @).

3.1 Example

Consider again NPR example.

$$@_{10}$$

$$\to^{(App)}$$

$$@_{10}\ \ @_6$$

$$\to^{(App)}$$

$$@_{10}\ \ @_6\ \ \lambda h$$

$$\to^{(Lam-Elim)}$$

$$@_{10}\ \ @_6\ \ \lambda h\ \ \lambda z$$

$$\to^{(Lam-Elim)}$$

$$@_{10}\ \ @_6\ \ \lambda h\ \ \lambda z\ \ @_4$$

$$\to^{(App)}$$

$$@_{10}\ \ @_6\ \ \lambda h\ \ \lambda z\ \ @_4\ \ @_1$$

$$\to^{(App)}$$

$$@_{10}\ \ @_6\ \ \lambda h\ \ \lambda z\ \ @_4\ \ @_1\ \ h$$

$$\to^{(BVar)}$$

$$@_{10}\ \ @_6\ \ \lambda h\ \ \lambda z\ \ @_4\ \ @_1\ \ h\ = \ \lambda f$$

$$\to^{(Lam-Elim)}$$

$$@_{10}\ \ @_6\ \ \lambda h\ \ \lambda z\ \ @_4\ \ @_1\ \ h\ = \ \lambda f\ \ \lambda y$$

$$\to^{(Lam-Elim)}$$

$$@_{10}\ @_6\ \lambda h\ \lambda z\ @_4\ @_1\ h = \lambda f\ \lambda y\ @_7$$

$$\to^{(App)}$$

$$@_{10}\ @_6\ \lambda h\ \lambda z\ @_4\ @_1\ h = \lambda f\ \lambda y\ @_7\ f$$

$$\to^{(BVar)}$$

$$@_{10}\ @_6\ \lambda h\ \lambda z\ @_4\ @_1\ h = \lambda f\ \lambda y\ @_7\ f = \lambda x$$

$$\to^{(Lam-Elim)}$$

$$@_{10}\ @_6\ \lambda h\ \lambda z\ @_4\ @_1\ h = \lambda f\ \lambda y\ @_7\ f = \lambda x\ @_3$$

$$\to^{(App)}$$

$$@_{10}\ @_6\ \lambda h\ \lambda z\ @_4\ @_1\ h = \lambda f\ \lambda y\ @_7\ f = \lambda x\ @_3\ @_2$$

$$\to^{(App)}$$

$$@_{10}\ @_6\ \lambda h\ \lambda z\ @_4\ @_1\ h = \lambda f\ \lambda y\ @_7\ f = \lambda x\ @_3\ @_2\ h$$

$$\to^{(BVar)}$$

$$@_{10}\ @_6\ \lambda h\ \lambda z\ @_4\ @_1\ h = \lambda f\ \lambda y\ @_7\ f = \lambda x\ @_3\ @_2\ h = \lambda f$$

$$\to^{(Lam-Elim)}$$

$$@_{10}\ @_6\ \lambda h\ \lambda z\ @_4\ @_1\ h = \lambda f\ \lambda y\ @_7\ f = \lambda x\ @_3\ @_2\ h = \lambda f\ \lambda y$$

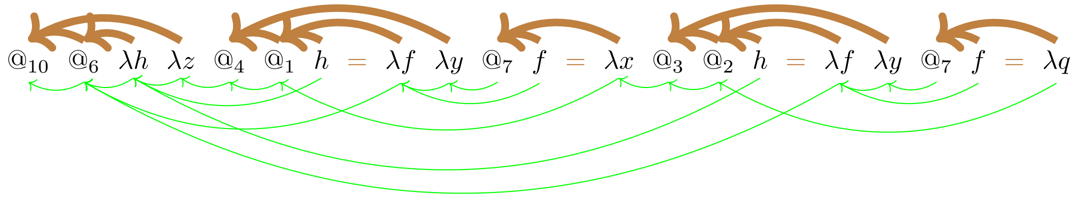
$$\to^{(Lam-Elim)}$$

$$@_{10}\ @_6\ \lambda h\ \lambda z\ @_4\ @_1\ h = \lambda f\ \lambda y\ @_7\ f = \lambda x\ @_3\ @_2\ h = \lambda f\ \lambda y\ @_7$$

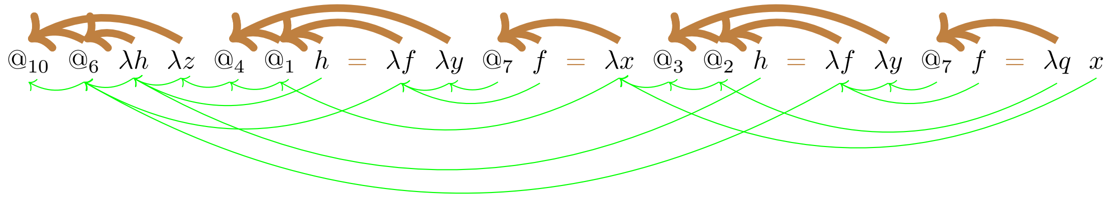
$$\to^{(App)}$$

$$@_{10}\ @_6\ \lambda h\ \lambda z\ @_4\ @_1\ h = \lambda f\ \lambda y\ @_7\ f = \lambda x\ @_3\ @_2\ h = \lambda f\ \lambda y\ @_7\ f$$

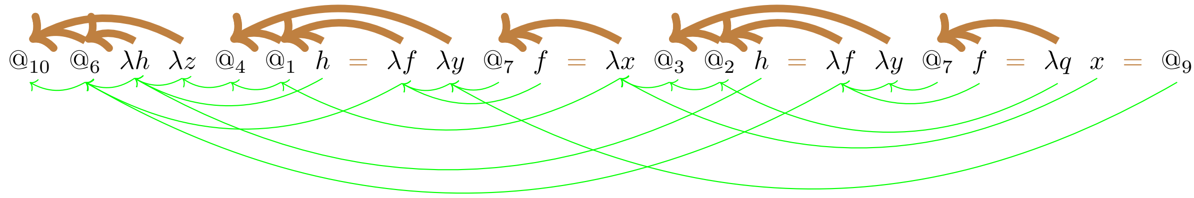
$\rightarrow^{(BVar)}$



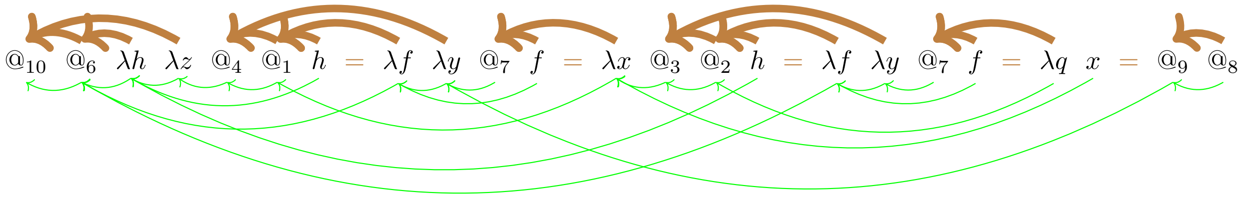
$\rightarrow^{(Lam-Elim)}$



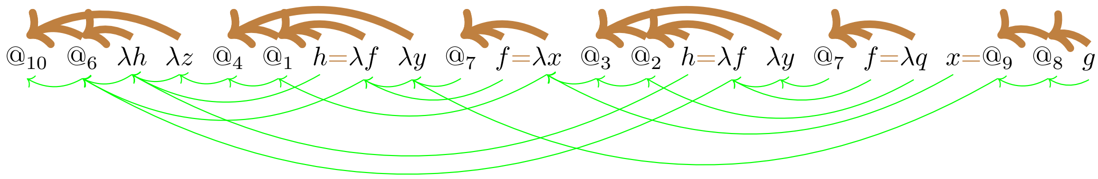
$\rightarrow^{(BVar)}$



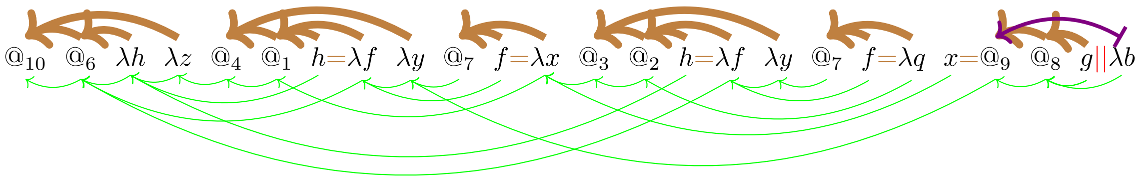
$\rightarrow^{(App)}$



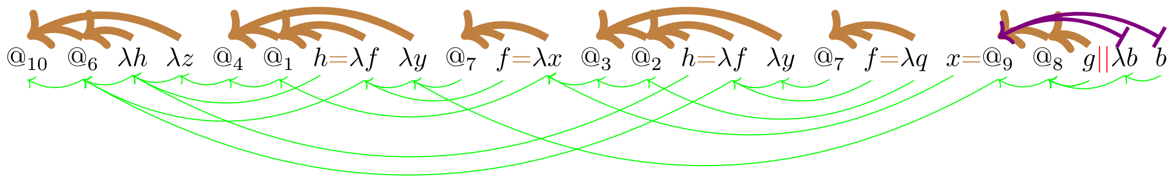
$\rightarrow^{(App)}$



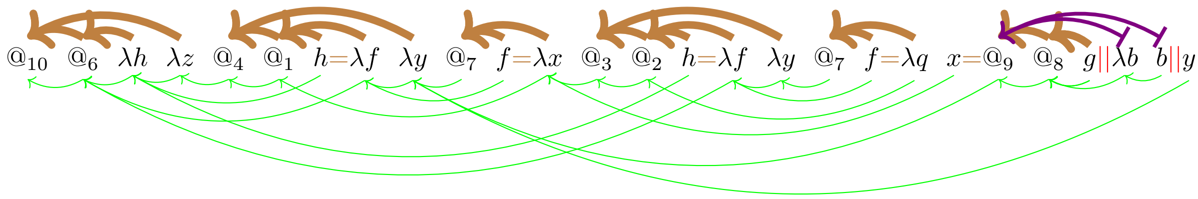
$\rightarrow^{(FVar-1)}$



$\rightarrow^{(Lam-Non-Elim)}$



$\rightarrow^{(FVar-2)}$



$$\to^{(BVar)}$$

$$@_{10}\quad @_6\quad \lambda h\quad \lambda z\quad @_4\quad @_1\quad h{=}\lambda f\quad \lambda y\quad @_7\quad f{=}\lambda x\quad @_3\quad @_2\quad h{=}\lambda f\quad \lambda y\quad @_7\quad f{=}\lambda q\quad x{=}@_9\quad @_8\quad g||\lambda b\quad b||y{=}@_5$$

$$\to^{(App)}$$

$$@_{10}\quad @_6\quad \lambda h\quad \lambda z\quad @_4\quad @_1\quad h{=}\lambda f\quad \lambda y\quad @_7\quad f{=}\lambda x\quad @_3\quad @_2\quad h{=}\lambda f\quad \lambda y\quad @_7\quad f{=}\lambda q\quad x{=}@_9\quad @_8\quad g||\lambda b\quad b||y{=}@_5\quad z$$

$$\to^{(BVar)}$$

$$@_{10}\quad @_6\quad \lambda h\quad \lambda z\quad @_4\quad @_1\quad h{=}\lambda f\quad \lambda y\quad @_7\quad f{=}\lambda x\quad @_3\quad @_2\quad h{=}\lambda f\quad \lambda y\quad @_7\quad f{=}\lambda q\quad x{=}@_9\quad @_8\quad g||\lambda b\quad b||y{=}@_5\quad z{=}@_{11}$$

$$\to^{(App)}$$

$$@_{10}\quad @_6\quad \lambda h\quad \lambda z\quad @_4\quad @_1\quad h{=}\lambda f\quad \lambda y\quad @_7\quad f{=}\lambda x\quad @_3\quad @_2\quad h{=}\lambda f\quad \lambda y\quad @_7\quad f{=}\lambda q\quad x{=}@_9\quad @_8\quad g||\lambda b\quad b||y{=}@_5\quad z{=}@_{11}\quad g$$

$$\to^{(FVar-1)}$$

$$@_{10}\quad @_6\quad \lambda h\quad \lambda z\quad @_4\quad @_1\quad h{=}\lambda f\quad \lambda y\quad @_7\quad f{=}\lambda x\quad @_3\quad @_2\quad h{=}\lambda f\quad \lambda y\quad @_7\quad f{=}\lambda q\quad x{=}@_9\quad @_8\quad g||\lambda b\quad b||y{=}@_5\quad z{=}@_{11}\quad g||\lambda n$$

$$\to^{(Lam-Non-Elim)}$$

$$@_{10}\quad @_6\quad \lambda h\quad \lambda z\quad @_4\quad @_1\quad h{=}\lambda f\quad \lambda y\quad @_7\quad f{=}\lambda x\quad @_3\quad @_2\quad h{=}\lambda f\quad \lambda y\quad @_7\quad f{=}\lambda q\quad x{=}@_9\quad @_8\quad g||\lambda b\quad b||y{=}@_5\quad z{=}@_{11}\quad g||\lambda n\quad n$$

$$\to^{(FVar-2)}$$

$$@^{1}_{10}\ @^{2}_{6}\ \lambda h^{3}\ \lambda z^{4}\ @^{5}_{4}\ @^{6}_{1}\ h^{7}=\lambda f^{8}\ \lambda y^{9}\ @^{10}_{7}\ f^{11}=\lambda x^{12}\ @^{13}_{3}\ @^{14}_{2}\ h^{15}=\lambda f^{16}\ \lambda y^{17}\ @^{18}_{7}\ f^{19}=\lambda q^{20}\ x^{21}=@^{22}_{9}\ @^{23}_{8}\ g^{24}||\lambda b^{25}\ b^{26}||y^{27}=@^{28}_{5}\ z^{29}=@^{30}_{11}\ g^{31}||\lambda n^{32}\ n^{33}||a^{34}$$

3.2 Display

It is easy to extend rules $(Lam - Elim)$ and $(BVar)$ to automatically crossing over nodes that will not appear in normal form:

$(Lam - Elim)$

$$\ldots\ @_1\ \ldots\ @_2\ \ldots\ \lambda x \longrightarrow \text{where } e \text{ is a child of } \lambda x \longrightarrow \quad \ldots\ @_1\ \ldots\ \cancel{@_2}\ \ldots\ \cancel{\lambda x}\ e$$

$(BVar)$

$$\ldots\ \cancel{@}\ \ldots\ \cancel{\lambda x}\ \ldots\ \overset{i}{x} \longrightarrow \text{where } e \text{ is is the right child of } @ \longrightarrow \quad \ldots\ \cancel{@}\ \ldots\ \cancel{\lambda x}\ \ldots\ \overset{i}{\cancel{x}} = e$$

Thus, the resulting traversal has to be as follows:

$$\cancel{@^{1}_{10}}\ \cancel{@^{2}_{6}}\ \cancel{\lambda h^{3}}\ \cancel{\lambda z^{4}}\ \cancel{@^{5}_{4}}\ \cancel{@^{6}_{1}}\ \cancel{h^{7}}=\cancel{\lambda f^{8}}\ \cancel{\lambda y^{9}}\ \cancel{@^{10}_{7}}\ \cancel{f^{11}}=\cancel{\lambda x^{12}}\ \cancel{@^{13}_{3}}\ \cancel{@^{14}_{2}}\ \cancel{h^{15}}=\cancel{\lambda f^{16}}\ \cancel{\lambda y^{17}}\ \cancel{@^{18}_{7}}\ \cancel{f^{19}}=\cancel{\lambda q^{20}}\ \cancel{x^{21}}=@^{22}_{9}\ @^{23}_{8}\ g^{24}||\lambda b^{25}\ b^{26}||\cancel{y^{27}}=@^{28}_{5}\ \cancel{z^{29}}=@^{30}_{11}\ g^{31}||\lambda n^{32}\ n^{33}||a^{34}$$

And the residual (infix notation) term is $@_9\ (@_8\ g\ (\lambda b\ .\ b)\ (@_5\ (@_{11}\ g\ (\lambda n\ .\ n))))\ a$