1 Labelled Transition System for Traversals

- *Input*: λ–term $M \in \Lambda$ where $\Lambda @ \Lambda \mid \Lambda\ x\ .\ \Lambda \mid x$;
- *State space* is a set of chains of the following view $n_1, \ \ldots, \ n_m, \ \ldots,$ where $\forall i, \ n_i$ is a token (a tree node) of M;
- *Transition labels* (optional) is a node to be added in the traversal on current state.

Some notes about traversals:

- There are two different kinds of pointers. Note that any traversal element has both of them.
  - First kind is either:
    * A pointer to the *last unfinished application*. I.e. a pointer to the last apllication within one run of *head linear reduction* (in other words, this pointer can not to get over $||$ sing) whose left had side is being under consideration or has been consedered yet while right hand side (argument of application) has not consedered and has not bound by some (Lam)–node. On traversal diagrams this kind of pointer is denoted as $\rightarrow$.
    * A pointer to the *last unfinished application* that is between nodes in different *head linear reduction* runs (in other words, this pointer has to get over at least one $||$ sing). On traversal diagrams this kind of pointer is denoted as $\rightarrow$.
    * A pointer that binds (Lam)–node with its argument. (for example, for λx node this pointer point to the application whose argument has to be substituted instead of x variable occurence in the future). On traversal diagrams this kind of pointer is denoted as $\rightarrow$.
  - Note that pointers described above can points only to some application in current history.
  - The second pointer is a *binder pointer* that for:
    * Bound variables points to the corresponding binder;
    * Free variables points to nowhere;
    * Application nodes and lambda nodes it points to the parent in scence of tree structure of input term.
  - On traversal diagrams binder pointer is denoted as $\rightarrow$.
  - A pointer - - > (dotted binder pointer) denotes "there exists a path between this to nodes by the chain of binder pointers".
  - $\rightarrow$ denotes either $\rightarrow$ or $\rightarrow$.

1.1 Rules

1. (BVars)
   - (BVar – Lam)

$$\begin{array}{ccccccc} & & & & & & i \\ \ldots & @_1 & \ldots & \lambda x & \ldots & x & \end{array}$$

$\longrightarrow^{\lambda y}$

$$\begin{array}{ccccccccc} & & & & & & i & & \\ \ldots & @_1 & \ldots & \lambda x & \ldots & x & = & \lambda y \end{array}$$

   - (BVar – App)

$$\begin{array}{ccccccc} & & & & & & i \\ \ldots & @_1 & \ldots & \lambda x & \ldots & x & \end{array}$$

$\longrightarrow^{@_2}$

$$\begin{array}{ccccccccc} & & & & & & i & & \\ \ldots & @_1 & \ldots & \lambda x & \ldots & x & = & @_2 \end{array}$$

   - (BVar – BVar)

$$\begin{array}{ccccccc} & & & & & & i \\ \ldots & @_1 & \ldots & \lambda x & \ldots & x & \end{array}$$

$\longrightarrow^{y}$, where $\exists\ \lambda y$ in history such that there is a chain of green pointers from $@_1$ to this λy

$$\begin{array}{ccccccccc} & & & & & & i & & \\ \lambda y & \ldots & @_1 & \ldots & \lambda x & \ldots & x & = & y \end{array}$$

- (BVar – FVar)

$$i$$
$$\ldots \quad @_1 \quad \ldots \quad \lambda x \quad \ldots \quad x$$

$\longrightarrow^{y}$, where $\exists\ \lambda y$ in history such that there is a chain of green pointers from $@_1$ to this λy

$$i$$
$$\ldots \quad @_1 \quad \ldots \quad \lambda x \quad \ldots \quad x \;=\; y$$

2. (FVars) and (BVars) without arguments

- (FVar – Not-FVar)

$$@_0 \quad \ldots \quad @_1 \quad \ldots \quad y$$

$\longrightarrow^{n}$, where n is a right child of $@_1$ and $n \neq$ (FVar) && $n \neq$ (BVar)

$$@_0 \quad \ldots \quad @_1 \quad \ldots \quad y \;||\; n$$

- (FVar – FVar)

$$@_0 \quad \ldots \quad @_1 \quad \ldots \quad y$$

$\longrightarrow^{x}$, where n is a right child of $@_1$ and $n =$ (FVar)

$$@_0 \quad \ldots \quad @_1 \quad \ldots \quad y \;||\; x$$

- (FVar – BVar)

$$\lambda x$$
$$@_0 \quad \ldots \quad @_1 \quad \ldots \quad y$$

$\longrightarrow^{x}$, where $@_1 = \ldots @x$ (BVar)

$$\lambda x$$
$$@_0 \quad \ldots \quad @_1 \quad \ldots \quad y \;||\; x$$

3. (Apps)

- (App – BVar)

$$\ldots \quad \lambda x \quad \ldots \quad @_1$$

$\longrightarrow^{x}$, where $@_1 = x@\ldots$ (BVar)

$$\ldots \quad \lambda x \quad \ldots \quad @_1 \quad x$$

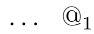
- (App – FVar)

$$\ldots \quad \ldots \quad @_1$$

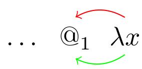
$\longrightarrow^{y}$, such that $\nexists \lambda y$ in tarversal: $@_1 -- > \lambda y$

$$\ldots \quad \ldots \quad @_1 \quad y$$

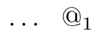
- (App – Lam)



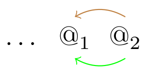
$\longrightarrow^{\lambda x}$



- (App – App)

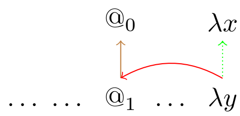


$\longrightarrow^{@_2}$

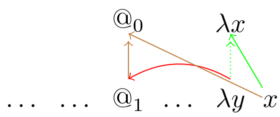


4. (Lam-Reds)

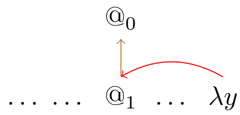
- (Lam-Red – BVar)



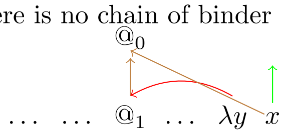
$\longrightarrow^{x}$



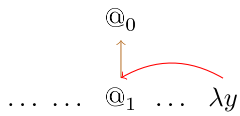
- (Lam-Red – FVar)



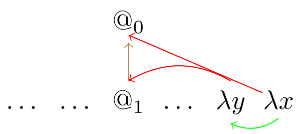
$\longrightarrow^{x}$, where $\nexists\ \lambda x$ in history such that there is no chain of binder (dreen) pointers from λy to λx



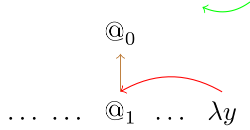
- (Lam-Red – Lam)



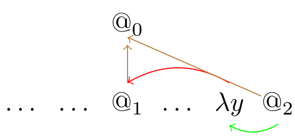
$\longrightarrow^{\lambda x}$



- (Lam-Red – App)

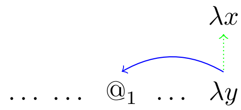


$\longrightarrow^{@_2}$

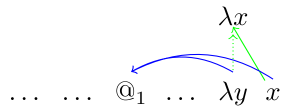


5. (Lam-Browns) and (Lam-Violet)

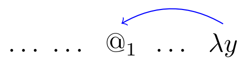
- (Lam-Blue – BVar)



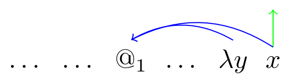
$\longrightarrow^{x}$



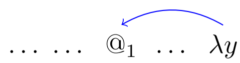
- (Lam-Blue – FVar)



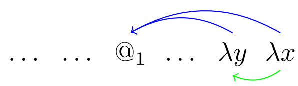
$\longrightarrow^{x}$, where $\nexists\ \lambda x$ in history such that there is no chain of binder (green) pointers from λy to λx



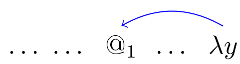
- (Lam-Blue – Lam)



$\longrightarrow^{\lambda x}$



- (Lam-Blue – App)



$\longrightarrow^{@_2}$