Proof-Nets and λ-calculus

Beniamino Accattoli

Ecole Polytechnique, LIX

Outline

Outline

1. Introduction
   - General motivations
   - MELL Proof-Nets and Girard's translation
   - Proof-Nets
2. Bisimulation
   - Statics
   - Dynamics
3. The structural λ-calculus (Accattoli, Kesner)
4. Properties

Outline

Premise

Proof-Nets	***Terms***
Visual and parallel syntax	***Understood by everyone***
Difficult to manage formally	***Easy inductive reasoning***
Non-inductive characterizations	***Too rigid tree structure***
Explicit name identification	***Name identification for free***
Nameless	***α-equivalence***

Goal: take the best of the two worlds!

PN perspective: an algebraic language

- PN are as ***nice to see*** as ***dreadful to manage***.
- Limited to a ***tiny community***.
- ***Idea***: try to find an ***elegant*** algebraic language for PN.
- ***Graphical intuitions*** $\Rightarrow$ ***algebraic theorems***.
- ***First Aim***: to increase the ***proving power*** of the PN-specialist.
- ***Second Aim***: PN technology to a ***wider community***.
- ***Third Aim***: to work with PN and write ***graph-free papers***!

Term persective: a syntactic model

- λ-calculus Proof-Nets are related to ***explicit substitutions*** (ES).
- ***No canonical ES-calculus*** despite 20 years of research.
- ***Problem***: ***too much freedom*** in designing the rules.
- ***In PN***: correctness and parallelism ***limit*** the possible choices.
- ***Idea***: re-design ES using PN as ***syntactic model***.

Outline

Multiplicative Exponential Linear Logic (MELL)

Identity rules:

$$\frac{}{\vdash A^{\perp}, A}\ \text{ax} \qquad \frac{\vdash \Gamma, A \qquad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}\ \text{cut}$$

Multiplicative rules:

$$\frac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\ \otimes \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\ \parr$$

Exponential rules:

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}\ ?\text{d} \qquad \frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}\ !$$

$$\frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\ ?\text{c} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?A}\ ?\text{w}$$

λ-calculus actually requires a ***tiny intuitionistic fragment*** of MELL.

Intuitionistic MELL

Identity group:

$$\frac{}{\vdash A^{\perp}, A}\ \text{ax} \qquad \frac{\vdash \Gamma, A \qquad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}\ \text{cut}$$

Linear implication:

$$\frac{\Gamma \vdash A \qquad \Delta, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C}\ \multimap_L \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\ \multimap_R$$

Exponentials:

$$\frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B}\ ?\text{d} \qquad \frac{!\Gamma \vdash A}{!\Gamma \vdash !A}\ !$$

$$\frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}\ ?\text{c} \qquad \frac{\Gamma \vdash B}{\Gamma, !A \vdash B}\ ?\text{w}$$

Girard's translation in sequent calculus

Formulas: $X^\circ = X$ and $(A \to B)^\circ = (!A^\circ) \multimap B^\circ$

Sequents: $(\Gamma \vdash A)^\circ = !\Gamma^\circ \vdash A^\circ$

Proofs:

$$\dfrac{}{A \vdash A}\,\text{ax} \quad\longrightarrow\quad \dfrac{\dfrac{}{A^\circ \vdash A^\circ}\,\text{ax}}{!A^\circ \vdash A^\circ}\,?\text{d}$$

$$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\Rightarrow_I \quad\longrightarrow\quad \dfrac{!\Gamma^\circ, !A^\circ \vdash B^\circ}{!\Gamma^\circ \vdash !A^\circ \multimap B^\circ}\multimap_R$$

$$\dfrac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B}\Rightarrow_E \quad\longrightarrow\quad \dfrac{!\Gamma^\circ \vdash !A^\circ \multimap B^\circ \qquad \dfrac{\dfrac{!\Gamma^\circ \vdash A^\circ}{!\Gamma^\circ \vdash !A^\circ}\,! \qquad \dfrac{}{B^\circ \vdash B^\circ}\,\text{ax}}{!\Gamma, !A^\circ \multimap B^\circ \vdash B}\multimap_L}{!\Gamma \vdash B}\,\text{cut}$$

Remark: every $\Rightarrow_E$ introduces a ***cut***.

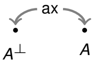
Untyped λ-calculus requires a ***recursive*** type $o = !o \multimap o = ?o \parr o$.

Outline

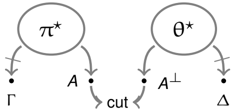
1. Introduction
   - General motivations
   - MELL Proof-Nets and Girard's translation
   - Proof-Nets
2. Bisimulation
   - Statics
   - Dynamics
3. The structural λ-calculus (Accattoli, Kesner)
4. Properties

MELL Proof-Nets 1: multiplicatives

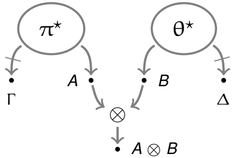
$$\left(\frac{}{\vdash A^{\perp}, A}\ \mathsf{ax}\right)^{\star} =$$



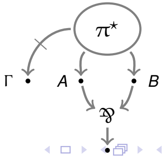
$$\left(\frac{\begin{array}{c}\vdots\\ \pi \\ \vdots \\ \vdash \Gamma, A\end{array} \qquad \begin{array}{c}\vdots\\ \theta \\ \vdots \\ \vdash A^{\perp}, \Delta\end{array}}{\vdash \Gamma, \Delta}\ \mathsf{cut}\right)^{\star} =$$



$$\left(\frac{\begin{array}{c}\vdots\\ \pi \\ \vdots \\ \vdash \Gamma, A\end{array} \qquad \begin{array}{c}\vdots\\ \theta \\ \vdots \\ \vdash \Delta, B\end{array}}{\vdash \Gamma, \Delta, A \otimes B}\ \otimes\right)^{\star} =$$

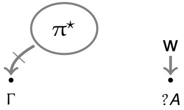


$$\left(\frac{\begin{array}{c}\vdots\\ \pi \\ \vdots \\ \vdash \Gamma, A, B\end{array}}{\vdash \Gamma, A \parr B}\ \parr\right)^{\star} =$$

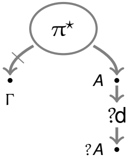


MELL Proof-Nets 2: exponentials

$$\left(\dfrac{\begin{array}{c} \vdots \\ \pi \\ \vdots \\ \vdash \Gamma \end{array}}{\vdash \Gamma, ?A}\, \mathsf{w} \right)^{\star} =$$

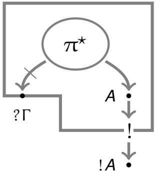


$$\left(\dfrac{\begin{array}{c} \vdots \\ \pi \\ \vdots \\ \vdash \Gamma, A \end{array}}{\vdash \Gamma, ?A}\, ?\mathsf{d} \right)^{\star} =$$

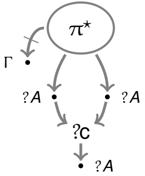


MELL Proof-Nets 2: exponentials

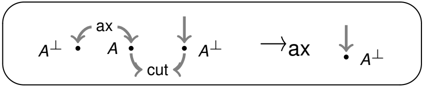
$$\left(\dfrac{\begin{array}{c} \vdots \\ \pi \\ \vdots \\ \vdash ?\Gamma, A \end{array}}{\vdash ?\Gamma, !A} \; ! \right)^{\star} =$$

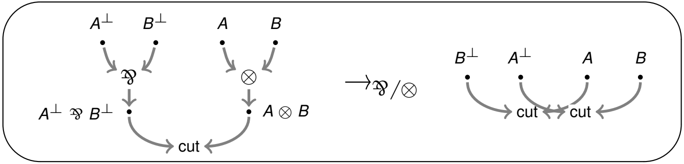


$$\left(\dfrac{\begin{array}{c} \vdots \\ \pi \\ \vdots \\ \vdash \Gamma, ?A, ?A \end{array}}{\vdash \Gamma, ?A} \; ?\mathsf{c} \right)^{\star} =$$

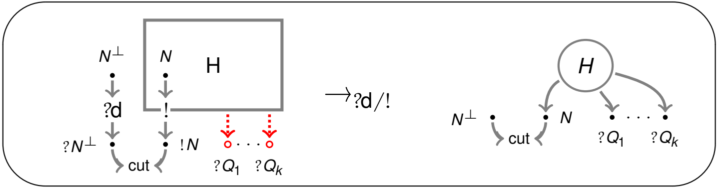


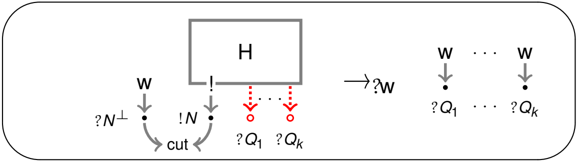
Multiplicative rules



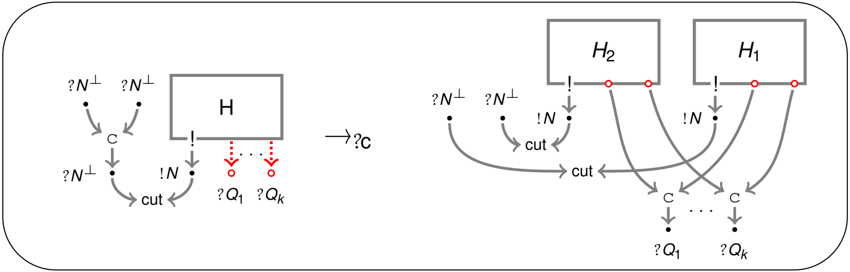


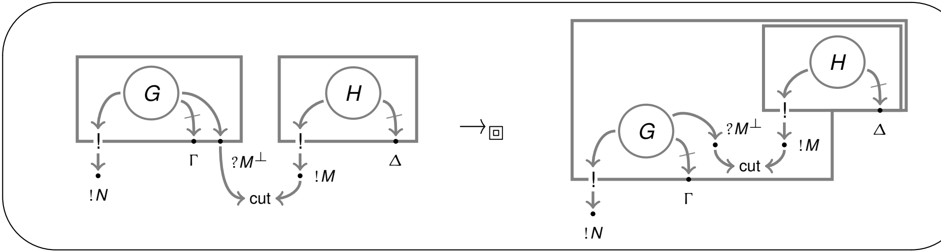
Exponential rules 1



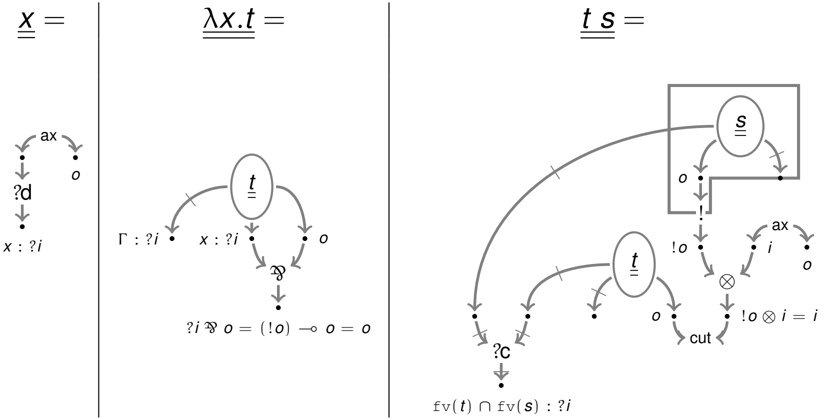


Exponential rules 1





Girard's translation on Proof-Nets



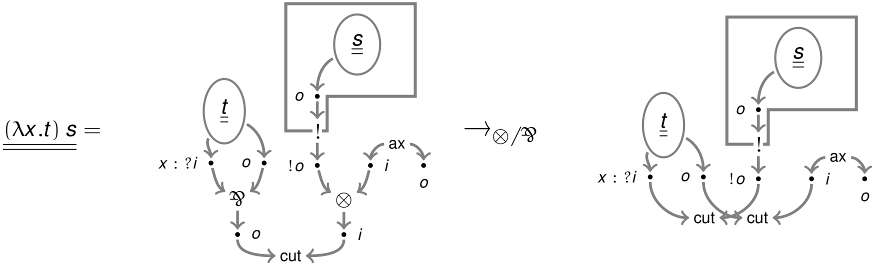
The translation of $t = (x\ y)\ z$ ***has an axiom cut*** while t is ***normal***.

Remark: the translation of a term can only have ***multiplicative cuts***.

Outline

A more relevant mismatch

Let's reduce a ***graphical*** β***-redex***:



We get a net with an ***exponential cut***, *i.e.* ***not a*** λ***-term***.

The further elimination of the exponential cut gets a λ-term.

Simulation of λ-calculus in PN

The ***operational relation*** between terms and graphs is as follows:

$$\begin{array}{ccc} t & \to_\beta & t' \\ \downarrow_{\doteq} & & \downarrow_{\doteq} \\ G_t & \to^+ & G_{t'} \end{array}$$

In the other sense:

$$G_t \;\to\; G' \quad\Rightarrow\quad \begin{array}{ccccc} t & & \to & & t' \\ \downarrow_{\doteq} & & & & \downarrow_{\doteq} \\ G_t & \to & G' & \to^* & G_{t'} \end{array}$$

To simplify we want to represent the ***intermediary steps*** on terms.

Bisimulation

Diagrammatically we want:

$$\begin{array}{ccc} t & & \\ \downarrow_{\dot{=}} & & \\ G_t & \rightarrow & G' \end{array} \quad \Rightarrow \quad \exists\, t' \text{ s.t. } \quad \begin{array}{ccc} t & \rightarrow & t' \\ \downarrow_{\dot{=}} & & \downarrow_{\dot{=}} \\ G_t & \rightarrow & G' \end{array}$$

And

$$\begin{array}{ccc} t & \rightarrow & t' \\ \downarrow_{\dot{=}} & & \\ G_t & & \end{array} \quad \Rightarrow \quad \exists\, G' \text{ s.t. } \quad \begin{array}{ccc} t & \rightarrow & t' \\ \downarrow_{\dot{=}} & & \downarrow_{\dot{=}} \\ G_t & \rightarrow & G' \end{array}$$

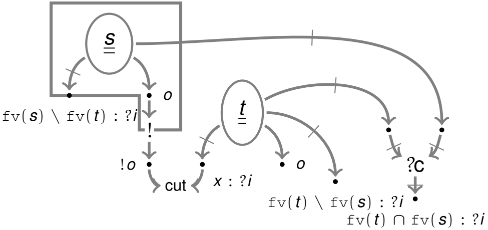
We want the translation to be a ***strong bisimulation***.

Outline

1. Introduction
   - General motivations
   - MELL Proof-Nets and Girard's translation
   - Proof-Nets
2. Bisimulation
   - Statics
   - Dynamics
3. The structural λ-calculus (Accattoli, Kesner)
4. Properties

Let's introduce ***explicit substitutions*** $t[x/s]$:

$\underline{\underline{t[x/s]}} =$



Now ***any graph*** can be mapped to a ***term***.

Various technical issues

- It is necessary to work ***modulo axiom cuts***.

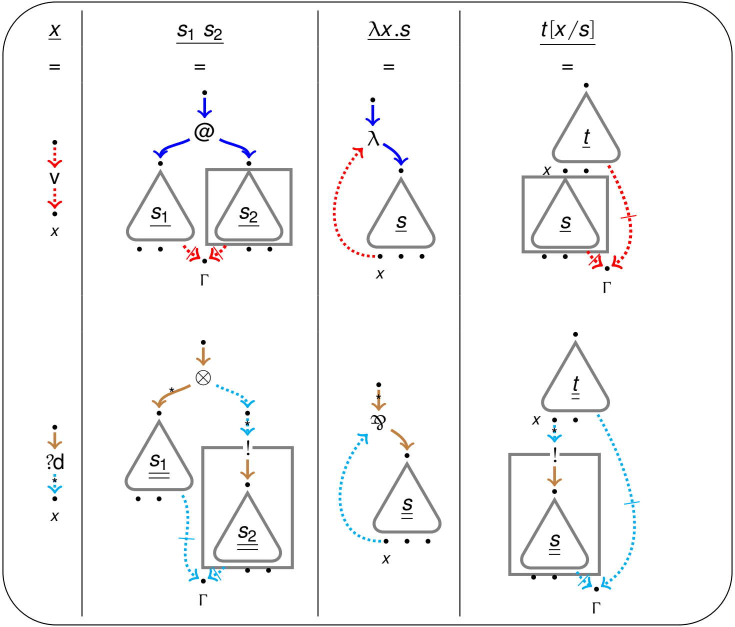
  $\Rightarrow$ ***We collapse axioms and cuts***.

- ***Variable representation*** presents some other technical issues:
  - Associativity and commutativity of contractions.
  - Permutation of contractions and weakenings with promotions.
  - On the fly removal of contractions with weakenings

  $\Rightarrow$ ***We collapse contractions***.

- Moreover, here we use PN to understand terms:

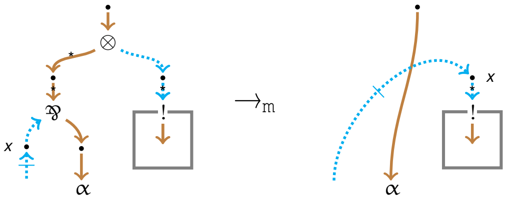
  $\Rightarrow$ ***We draw PN as syntax trees of terms***.



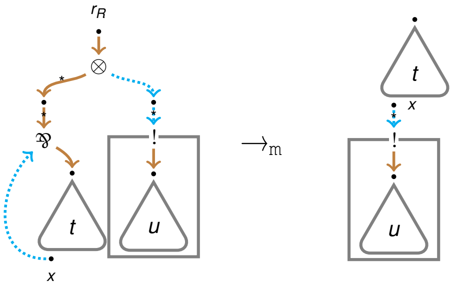
x
=
v
x
?d
x
s1 s2
=
@
s1
s2
Γ
⊗
!
λx.s
=
λ
s
x
⅋
t[x/s]
=
t
s
x
!
Γ

Outline

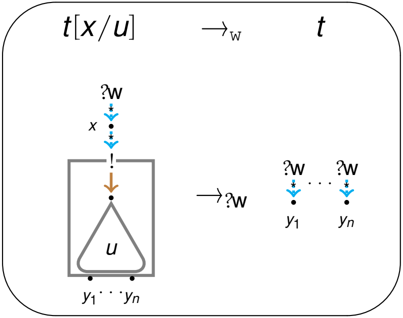
New β-rule



$$(\lambda x.t)\ u \quad \rightarrow \quad t[x/u]$$



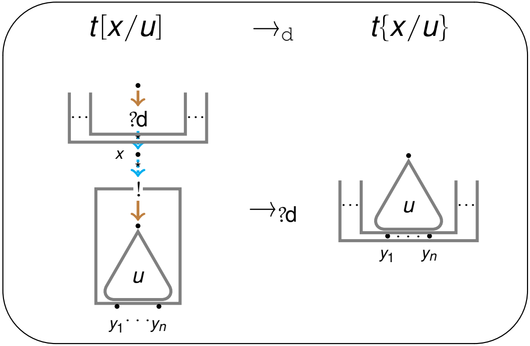
Garbage Collection



If $x \notin \mathtt{fv}(t)$ then $t[x/u]$ as a cut ***as this one***.

Thus it reduces to t, since u is ***simply erased***.

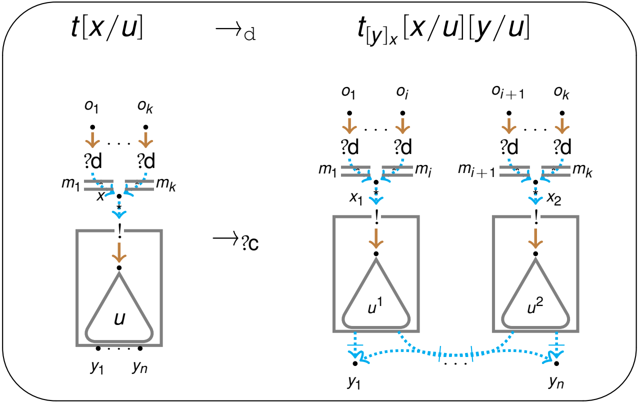
Linear Substitution



If $x \in \mathtt{fv}(t)$ and $|t|_x = 1$ then x is ***replaced*** by u.

Remark: reduction ***through*** box borders.

Duplication

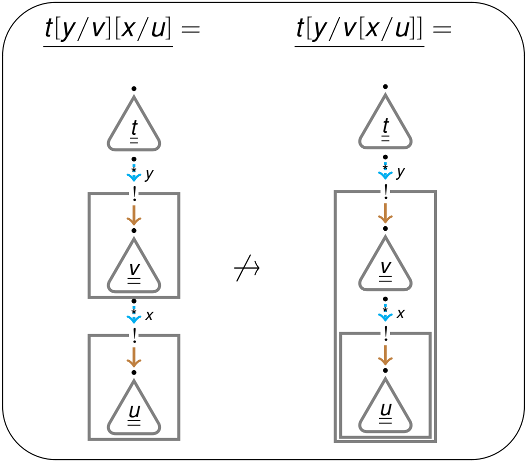


If $x \in \mathtt{fv}(t)$ and $|t|_x > 1$ then ***some occurrences*** of x give raise to a ***new variable***, substituted by a copy of u.

No commutative cut!

Main feature*: *no commutative cut!

This rule ***is not*** part of the operational semantics:



General Picture

Minimal LJ Big-steps cut-elimination	$\Leftrightarrow$	λ***-calculus*** β-reduction.
Minimal LJ Small-steps cut-elimination ***With*** commutative reductions	$\Leftrightarrow$	λ***-calculus*** With Explicit Substitutions
Minimal LJ Small-steps cut-elimination. ***Without*** commutative reductions	$\Leftrightarrow$	***Structural*** λ***-calculus*** λj $\simeq$ Proof-Nets

Outline

Structure vs Multiplicity

Traditionally, ES are introduced as follows:

$$(\lambda x.t)\ u \rightarrow_{\mathrm{B}} t[x/u]$$

Rules act by induction on the ***structure*** of terms by ***proximity***.

$$
\begin{array}{lcl}
x[x/u] & \rightarrow & u \\
y[x/u] & \rightarrow & y \\
(\lambda y.t)[x/u] & \rightarrow & \lambda y.t[x/u] \\
(t\ v)[x/u] & \rightarrow & t[x/u]\ v[x/u] \\
t[y/v][x/u] & \rightarrow & t[x/u][y/v[x/u]]
\end{array}
$$

Structure vs Multiplicity

The $\lambda\mathtt{j}$-calculus acts by induction on ***multiplicities*** at a ***distance***.

$$t[x/u] \rightarrow \ldots \quad \text{if } |t|_x = 0$$

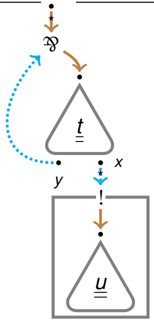
$$t[x/u] \rightarrow \ldots \quad \text{if } |t|_x = 1$$

$$t[x/u] \rightarrow \ldots \quad \text{if } |t|_x > 1$$

Modular: new constructors do not need new rules!

The translation is ***not injective***, in particular:

$$\underline{\underline{(\lambda y.t)[x/u]}} = \underline{\underline{\lambda y.t[x/u]}} =$$



Now consider $(\lambda y.t)[x/u]\ v = (\lambda y.t[x/u])\ v$. ***Both have a redex!***

Operational Semantics of $\lambda\mathtt{j}$: a new $\mathtt{B}$

- Rule ***at a distance***:

$$(\lambda x.t)[\cdot/\cdot]\ldots[\cdot/\cdot]\ u \rightarrow_{\mathtt{B-distance}} t[x/u][\cdot/\cdot]\ldots[\cdot/\cdot]$$

- Traditionally a configuration like:

$$(\lambda x.t)[y/v]\ u$$

is not a redex, as it is ***blocked*** by $[y/v]$.

- In $\lambda\mathtt{j}$, instead, ***it is a redex***.

Operational Semantics of $\lambda \mathtt{j}$: jumps

- ES rules by induction on ***multiplicities***:

$$
\begin{array}{llll}
(\lambda x.t)L\, u & \rightarrow_{\mathtt{B-distance}} & t[x/u]L & \\
t[x/u] & \rightarrow_{\mathtt{weakening}} & t & |t|_x = 0 \\
t[x/u] & \rightarrow_{\mathtt{dereliction}} & t\{x/u\} & |t|_x = 1 \\
t[x/u] & \rightarrow_{\mathtt{contraction}} & t_{[y]_x}[x/u][y/u] & |t|_x > 1 \ \&\ y \text{ fresh}
\end{array}
$$

Distance on terms = ***Locality on graphs***.

Example

$((\lambda z.(\lambda x.x\ x))\ z')\ y$	$\rightarrow_{\texttt{B-distance}}$
$(\lambda x.x\ x)[z/z']\ y$	$\rightarrow_{\texttt{B-distance}}$
$(x\ x)[x/y][z/z']$	$\rightarrow_{\texttt{weakening}}$
$(x\ x)[x/y]$	$\rightarrow_{\texttt{contraction}}$
$(x_1\ x_2)[x_1/y][x_2/y]$	$\rightarrow_{\texttt{dereliction}}$
$(y\ x_2)[x_2/y]$	$\rightarrow_{\texttt{dereliction}}$
$y\ y$	

Outline

$\equiv_\circ$-equivalence

- The translation on graphs induces a quotient:

$$\begin{array}{rcll}(\lambda y.t)[u/x] & \equiv & \lambda y.(t[u/x]) & \text{if } y \notin \mathtt{fv}(u)\\ (t[u/x])\ v & \equiv & (t\ v)[u/x] & \text{if } x \notin \mathtt{fv}(v)\\ t[x/u][y/v] & \equiv & t[y/v][x/u] & \text{if } y \notin \mathtt{fv}(u)\ \&\ x \notin \mathtt{fv}(v)\end{array}$$

- Which is a strong bisimulation by construction:

$$\begin{array}{ccc} t & \rightarrow & t' \\ \downarrow_{\cdot} & & \downarrow_{\cdot} \\ G & \rightarrow & G' \\ \uparrow_{\cdot} & & \uparrow_{\cdot} \\ s & \rightarrow & s' \end{array}$$