A link between lambda calculus and maps

Noam Zeilberger[1]

MSR-Inria Joint Centre

SpecFun seminar
29 September 2014

[1] based on a joint article with Alain Giorgetti: arxiv.org/abs/1408.5028

Part 1: Lambda calculus and its refinements

A Turing-complete language based on just two operations:

$$t(u) \qquad \text{(function application)}$$
$$\lambda x.t \qquad \text{(function abstraction)}$$

("Everything is a function.")

Consider terms up to renaming of variables, e.g.:

$$\lambda x.\lambda y.\lambda z.z(yx) = \lambda a.\lambda b.\lambda c.c(ba)$$
$$\lambda x.\lambda y.y = \lambda y.\lambda x.x$$
$$\lambda x.xx = \lambda y.yy$$

The main/only rule of computation is β*-reduction*:

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

where $t[u/x]$ is the "capture-avoiding" substitution of u for x in t.

$$
\begin{aligned}
(\lambda x.xx)(\lambda y.y) &\rightsquigarrow xx[(\lambda y.y)/x] \\
&= (\lambda y.y)(\lambda y.y) \\
&\rightsquigarrow y[(\lambda y.y)/y] \\
&= \lambda y.y
\end{aligned}
$$

$$
\begin{aligned}
((\lambda x.\lambda y.x)w)(\lambda z.z) &\rightsquigarrow (\lambda y.x[w/x])(\lambda z.z) \\
&= (\lambda y.w)(\lambda z.z) \\
&\rightsquigarrow w[(\lambda z.z)/y] \\
&= w
\end{aligned}
$$

$$
\begin{aligned}
(\lambda x.xx)(\lambda x.xx) &\rightsquigarrow xx[(\lambda x.xx)/x] \\
&= (\lambda x.xx)(\lambda x.xx) \\
&\rightsquigarrow \ldots
\end{aligned}
$$

We can use *typing* to isolate various interesting fragments of lambda calculus…

The simply-typed terms

A *simple type* τ is either the atomic type $\tau = o$ or a function type $\tau = \tau_1 \to \tau_2$.

Let Γ and Δ range over ordered lists of assumptions $x_1 : \tau_1, \ldots, x_i : \tau_i$ (all x_k distinct). Consider the following rules:

$$\frac{}{\Gamma, x : \tau, \Delta \vdash x : \tau} \qquad \frac{\Gamma \vdash t : \tau_1 \to \tau_2 \quad \Gamma \vdash u : \tau_1}{\Gamma \vdash t(u) : \tau_2} \qquad \frac{x : \tau_1, \Gamma \vdash t : \tau_2}{\Gamma \vdash \lambda x.t : \tau_1 \to \tau_2}$$

Definition

A term t is **simply-typed** if there exists a derivation (using the above rules) of $\Gamma \vdash t : \tau$ for some Γ and τ.

$\lambda x.\lambda y.x$ is simply-typed:

$$\dfrac{\dfrac{\overline{y : o, x : o \vdash x : o}}{x : o \vdash \lambda y.x : o \to o}}{\vdash \lambda x.\lambda y.x : o \to (o \to o)}$$

$\lambda x.xx$ is **not** simply-typed:

$$\dfrac{\dfrac{x : \tau_1 \vdash \overset{??}{x : \tau_1 \to \tau_2} \quad \overline{x : \tau_1 \vdash x : \tau_1}}{x : \tau_1 \vdash xx : \tau_2}}{\vdash \lambda x.xx : \tau_1 \to \tau_2}$$

All terms

Intuitively, the pure lambda calculus is based on a universal type $U \cong U \to U$.

This motivates the following type system, here Γ and Δ ranging over lists of assumptions $x_1 : U, \ldots, x_i : U$.

$$\frac{}{\Gamma, x : U, \Delta \vdash x : U} \qquad \frac{\Gamma \vdash t : U \quad \Gamma \vdash u : U}{\Gamma \vdash t(u) : U} \qquad \frac{x : U, \Gamma \vdash t : U}{\Gamma \vdash \lambda x.t : U}$$

Now for *any* term t with free variables $x_1, \ldots, x_i$ there is a (unique) derivation of $x_1 : U, \ldots, x_i : U \vdash t : U$.

Effectively, this type system defines the “fragment” of pure lambda calculus including *everything*.[2]

[2] It is not a trivial definition, though, since it relates lambda terms to certain kinds of trees.

Planar terms

But suppose we restrict the variable and application rules:

$$\frac{}{x : P \vdash x : P} \qquad \frac{\Gamma \vdash t : P \quad \Delta \vdash u : P}{\Gamma, \Delta \vdash t(u) : P} \qquad \frac{x : P, \Delta \vdash t : P}{\Delta \vdash \lambda x.t : P}$$

This defines a much smaller fragment of terms, enforcing the following discipline: every variable is used *exactly once* and in *last-in, first-out order*.

We refer to such terms as **planar**.

$\lambda x.\lambda y.yx$ is planar:

$$\dfrac{\dfrac{\dfrac{\overline{y : P \vdash y : P} \quad \overline{x : P \vdash x : P}}{y : P, x : P \vdash yx : P}}{x : P \vdash \lambda y.yx : P}}{\vdash \lambda x.\lambda y.yx : P}$$

$\lambda x.\lambda y.(\lambda z.zy)x$ is planar:

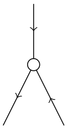
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{z : P \vdash z : P} \quad \overline{y : P \vdash y : P}}{z : P, y : P \vdash zy : P}}{y : P \vdash \lambda z.zy : P} \qquad \overline{x : P \vdash x : P}}{y : P, x : P \vdash (\lambda z.zy)x : P}}{x : P \vdash \lambda y.(\lambda z.zy)x : P}}{\vdash \lambda x.\lambda y.(\lambda z.zy)x : P}$$

$\lambda x.\lambda y.y(\lambda z.xz)$ is **not** planar:

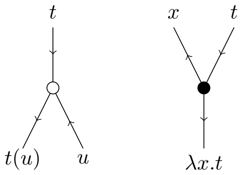
$$
\dfrac{\dfrac{\dfrac{\dfrac{}{y : P \vdash y : P} \qquad \dfrac{\overset{??}{z : P, x : P \vdash xz : P}}{x : P \vdash \lambda z.xz : P}}{y : P, x : P \vdash y(\lambda z.xz) : P}}{x : P \vdash \lambda y.y(\lambda z.xz) : P}}{\vdash \lambda x.\lambda y.y(\lambda z.xz) : P}
$$

Planar diagrams for planar lambda terms

A *string diagram* is a way of representating a morphism in a monoidal category. One can define string diagrams for **linear** lambda terms, built out of two basic connectors

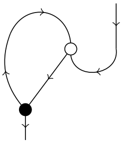


which intuitively may be annotated like so:



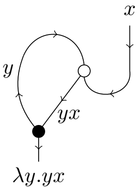
In particular, the wire annotated by *x* is hooked up directly to its *unique occurrence* in *t*.

The term $\lambda y.yx$ (with one free variable x) corresponds to the following string diagram:

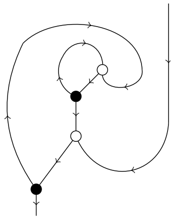


(A)

We can annotate the diagram to see this more clearly:

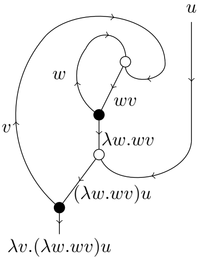


The term $\lambda v.(\lambda w.wv)u$ (with free variable u) corresponds to

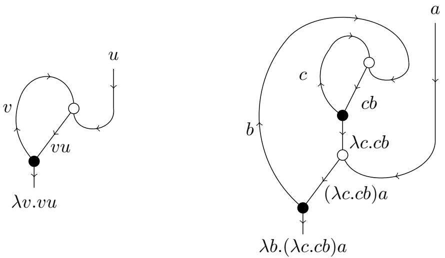


(B)

Again we can annotate the diagram to see this better:



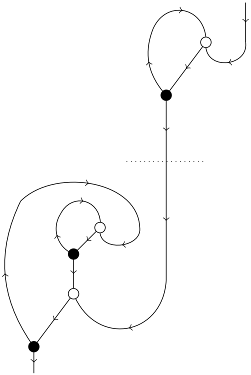
Since terms are always considered up to variable renaming, the following are also perfectly good annotations of (A) and (B):



So, by considering the underlying diagrams themselves, we can represent just the essential structure of lambda terms.

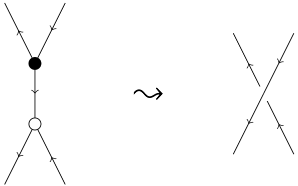
Similarly, we can represent the essential structure of substitution by just "plugging" one diagram into another:

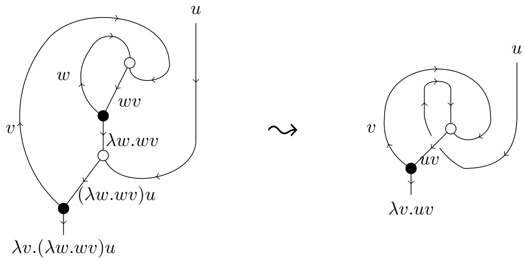
$$\lambda v.(\lambda w.wv)u[(\lambda y.yx)/u] = \lambda v.(\lambda w.wv)(\lambda y.yx)$$



Planarity and β-reduction

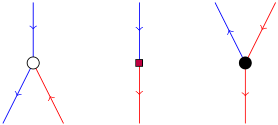
Observe that planarity (no crossing of wires) enforces the last-in, first-out (LIFO) discipline on the use of variables. However, planarity is **not** preserved by β-reduction:



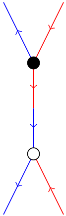


One “response” is to restrict to β-**normal** planar terms.[3]

Normality can be enforced by a simple coloring discipline:

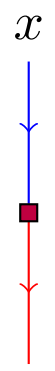


Since there is no coercion red ⇒ blue,
it is impossible to color a β-redex.

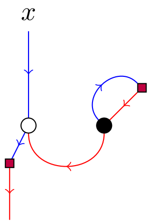


[3] Another possible response is to switch from a LIFO discipline to a FIFO discipline, which is preserved under β-reduction. However, even in that case it is still interesting to isolate the β-normal terms.

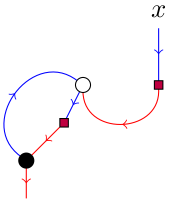
A short catalogue of normal planar terms (with one free variable)[4]



x

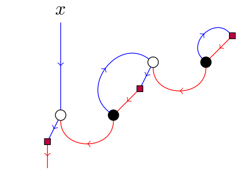


$x(\lambda y.y)$

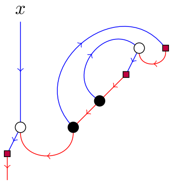


$\lambda y.yx$

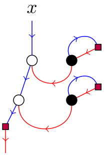
[4]Note all of these diagrams have one blue incoming wire and one red outgoing wire.



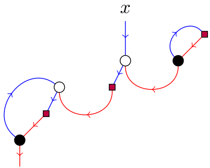
$x(\lambda y.y(\lambda z.z))$



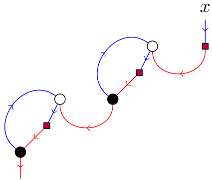
$x(\lambda y.\lambda z.zy)$



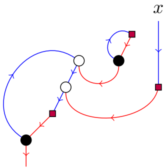
$(x(\lambda y.y))(\lambda z.z)$



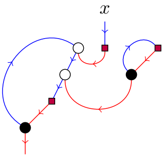
$\lambda y.y(x(\lambda z.z))$



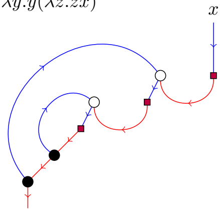
$\lambda y.y(\lambda z.zx)$



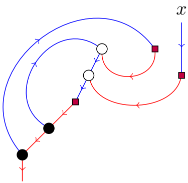
$\lambda y.(y(\lambda z.z))x$



$\lambda y.(yx)(\lambda z.z)$



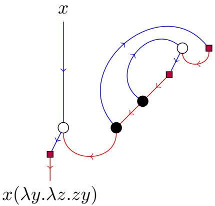
$\lambda y.\lambda z.z(yx)$



$\lambda y.\lambda z.(zy)x$

Part 2: A curious coincidence

Let the *size* of a normal planar lambda term be defined as the number of purple boxes in its string diagram.



(size 3)

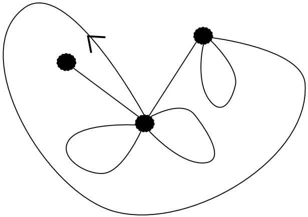
Count the normal planar lambda terms[5] of size $n + 1$:

$$1, 2, 9, 54, 378, 2916, 24057$$

This coincides with the first seven entries of A000168, counting the number of rooted planar maps with *n* edges!

[5] With no free variables, or equivalently with one free variable.

A rooted planar map is essentially a connected graph embedded in the plane, with one edge marked and assigned an orientation:



They were first counted by Tutte in the 1960s. Let's begin by proving that the two sequences really coincide, and then go on to establish a bijection.

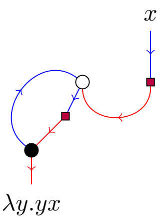
An inductive definition of normal (and neutral) planar terms

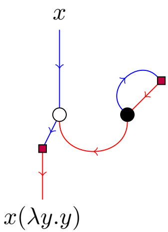
The coloring scheme corresponds to a refinement of the type system for planarity:

$$\frac{}{x : A \vdash x : A} \qquad \frac{\Gamma \vdash t : A \quad \Delta \vdash u : B}{\Gamma, \Delta \vdash t(u) : A}\ a$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : B}\ s \qquad \frac{x : A, \Delta \vdash t : B}{\Delta \vdash \lambda x.t : B}\ \ell$$

A = blue = "neutral"
B = red = normal
size = # of applications of the s rule in a derivation





$$\dfrac{\dfrac{\dfrac{\dfrac{}{y : A \vdash y : A} \qquad \dfrac{\dfrac{}{x : A \vdash x : A}}{x : A \vdash x : B}\ \boxed{s}}{y : A, x : A \vdash yx : A}\ a}{y : A, x : A \vdash yx : B}\ \boxed{s}}{x : A \vdash \lambda y.yx : B}\ \ell$$

$$\dfrac{\dfrac{\dfrac{}{x : A \vdash x : A} \qquad \dfrac{\dfrac{\dfrac{}{y : A \vdash y : A}}{y : A \vdash y : B}\ \boxed{s}}{\vdash \lambda y.y : B}\ \ell}{x : A \vdash x(\lambda y.y) : A}\ a}{x : A \vdash x(\lambda y.y) : B}\ \boxed{s}$$

From typing to generating functions

$$\frac{}{x : A \vdash x : A} \qquad \frac{\Gamma \vdash t : A \quad \Delta \vdash u : B}{\Gamma, \Delta \vdash t(u) : A}\ a$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : B}\ s \qquad \frac{x : A, \Delta \vdash t : B}{\Delta \vdash \lambda x.t : B}\ \ell$$

Define families of generating functions $A_i(z)$ and $B_i(z)$, counting neutral and normal planar terms (with i free variables) by size:

$$A_i(z) = [i = 1] + \sum_{j+k=i} A_j(z)B_k(z)$$
$$B_i(z) = zA_i(z) + B_{i+1}(z)$$

$$A_i(z) = [i = 1] + \sum_{j+k=i} A_j(z)B_k(z)$$
$$B_i(z) = zA_i(z) + B_{i+1}(z)$$

Next, formally aggregate these families

$$A(z,x) \stackrel{\text{def}}{=} \sum_{i \geq 0} A_i(z)x^i \qquad B(z,x) \stackrel{\text{def}}{=} \sum_{i \geq 0} B_i(z)x^i$$

to define a single pair of GFs counting terms along both size and number of free variables.

Unfolding definitions, we derive the following equations:

$$A(z,x) = x + A(z,x)B(z,x)$$
$$B(z,x) = zA(z,x) + \frac{1}{x}(B(z,x) - B_0(z))$$

$$A(z,x) = x + A(z,x)B(z,x)$$
$$B(z,x) = zA(z,x) + \frac{1}{x}(B(z,x) - B_0(z))$$

Proposition

The generating function $B_0(z)$ satisfies

$$B_0(z) = -\frac{1}{54z}\left(1 - 18z - (1-12z)^{3/2}\right)$$

(Proof by quadratic method.)

$$B_0(z) = -\frac{1}{54z}\left(1 - 18z - (1 - 12z)^{3/2}\right)$$
$$B_1(z) = B_0(z)$$

Corollary

The number of rooted planar maps with n edges is equal to the number of closed normal planar lambda terms of size $n + 1$, and to the number of normal planar lambda terms with one free variable of size $n + 1$.

Number of normal planar terms of size n with i free variables.

$i \backslash n$	1	2	3	4	5	6	7	8
1	1	2	9	54	378	2916	24057	208494
2	0	1	6	40	295	2346	19739	173426
3	0	0	2	20	175	1526	13587	123978
4	0	0	0	5	70	756	7602	74964
5	0	0	0	0	14	252	3234	36828
6	0	0	0	0	0	42	924	13728

(Row $i = 1$ is **A000168**.)

Number of neutral planar terms of size *n* with *i* free variables.

i \ n	0	1	2	3	4	5	6	7
1	1	1	3	14	83	570	4318	35068
2	0	1	4	20	120	820	6152	49448
3	0	0	2	15	105	770	5985	49014
4	0	0	0	5	56	504	4368	38136
5	0	0	0	0	14	210	2310	23100
6	0	0	0	0	0	42	792	10296

(Row $i = 1$ is **A220910**.)

Part 3: Tutte decomposition

Definition

A **(topological) map** is an embedding of a connected graph (possibly with loops and multiple edges) on a compact oriented surface X. A map is **planar** if $X =$ sphere.

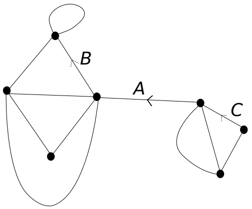
Definition

A **dart** is an edge together with an orientation. A **rooting** of a map is a choice of dart, but the vertex map (with no edges) is also considered rooted by convention. A **rooted planar map** is a planar map equipped with a rooting, treated up to root-preserving homeomorphism.

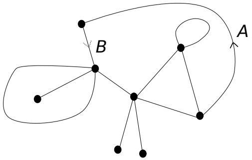
Trichotomy

Any rooted planar map *M* falls in one of three possible classes:

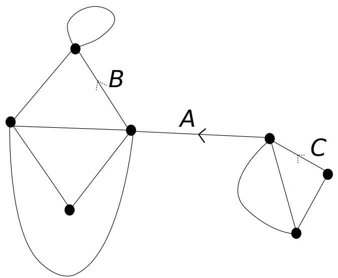
1. *M* is the vertex map
2. *M* has an isthmic root:



3. *M* has a non-isthmic root:

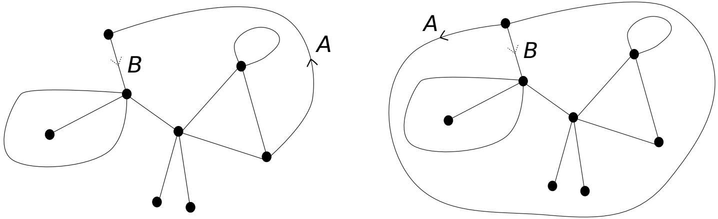


The isthmic case



Removing A yields two RPMs M_1 and M_2 (with roots B and C). Moreover, this operation is reversible: there is a binary operation $c_I(-,-)$, which given two RPMs M_1 and M_2, constructs $M = c_I(M_1, M_2)$.

The non-isthmic case



Removing A yields a RPM M_1 (with root B). Conversely, there is a *family* of operations $c_N^{(k)}(-)$ which given a RPM M_1 of outer face degree $\geq k$, constructs a RPM by starting at the root vertex x of M_1, walking k darts along the outer face, and adding a root pointing back to x.

(Above, $M_L = c_N^{(8)}(M_1)$ and $M_R = c_N^{(0)}(M_1)$.)

Theorem (Tutte 1968)

Let M be a rooted planar map with $e(M)$ edges and outer face degree $o(M)$. Then exactly one of the following cases must hold:

(i) *M is the vertex map and $e(M) = o(M) = 0$.*

(ii) *$M = c_I(M_1, M_2)$ for some M_1 and M_2 such that $e(M) = 1 + e(M_1) + e(M_2)$ and $o(M) = 2 + o(M_1) + o(M_2)$.*

(iii) *$M = c_N^{(k)}(M_1)$ for some M_1 and $0 \leq k \leq o(M_1)$ such that $e(M) = 1 + e(M_1)$ and $o(M) = k + 1$.*

Part 4: Replaying Tutte in lambda calculus

Most natural to consider normal planar lambda terms with one free variable (NPTVs).

Definition

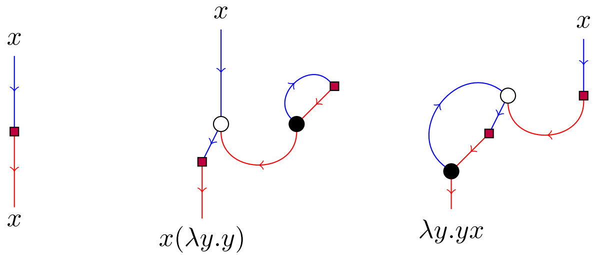
Let t be a NPTV with free variable x. We say that t is **the variable term** if $t = x$, that it is **function-open** if x is applied to some (normal) subterm of t, and that it is **argument-open** if x is passed as an argument to some (neutral) subterm of t.

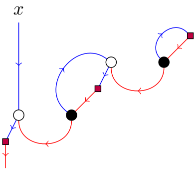
Proposition (*Trichotomy*)

Every NPTV is either the variable term, function-open, or argument-open (mutually exclusively).

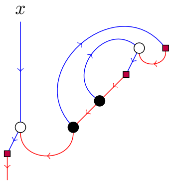
It's easiest to see this on the string diagram f of a NPTV t:

1. t is the variable term iff f is the basic diagram s.
2. t is function-open iff the incoming wire of f runs directly to an a-node.
3. t is argument-open iff the incoming wire of f runs directly to an s-node, which then connects directly to an a-node.

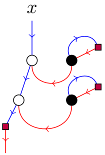

x
x
x
$x(\lambda y.y)$
x
$\lambda y.yx$



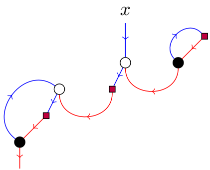
$x(\lambda y.y(\lambda z.z))$



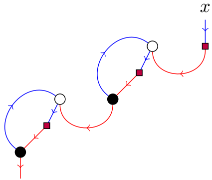
$x(\lambda y.\lambda z.zy)$



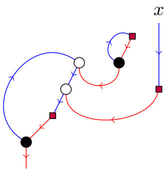
$(x(\lambda y.y))(\lambda z.z)$



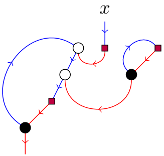
$\lambda y.y(x(\lambda z.z))$



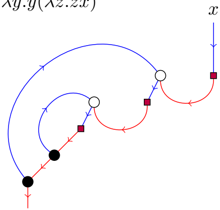
$\lambda y.y(\lambda z.zx)$



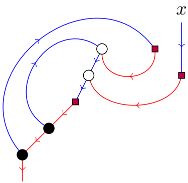
$\lambda y.(y(\lambda z.z))x$



$\lambda y.(yx)(\lambda z.z)$



$\lambda y.\lambda z.z(yx)$



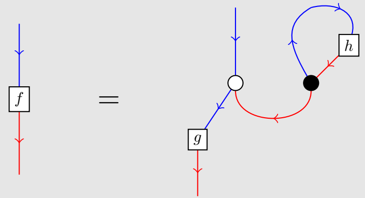
$\lambda y.\lambda z.(zy)x$

Now let's further decompose the function-open and argument-open cases...

The function-open case

Proposition

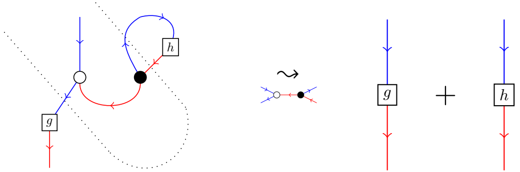
The diagram f of any function-open NPTV factors as



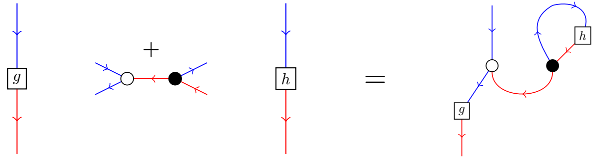
for some g and h.

The function-open case

Surgery on a function-open NPTV extracts a pair of NPTVs:

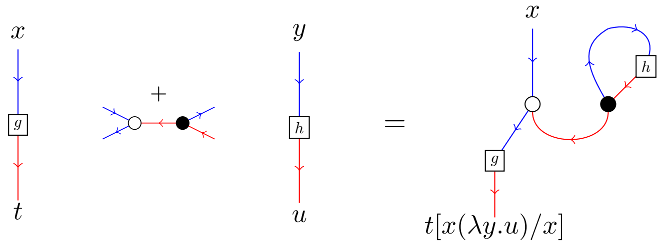


Moreover this surgery is reversible:

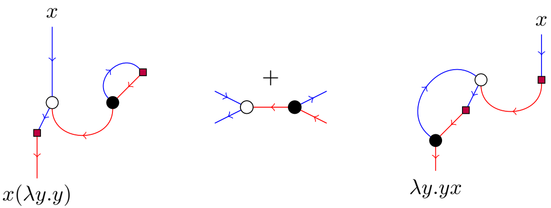


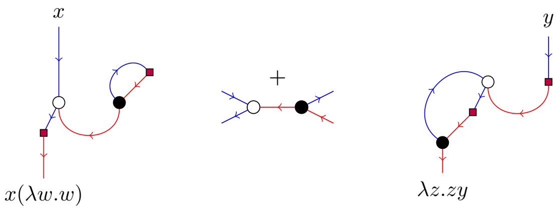
The function-open case

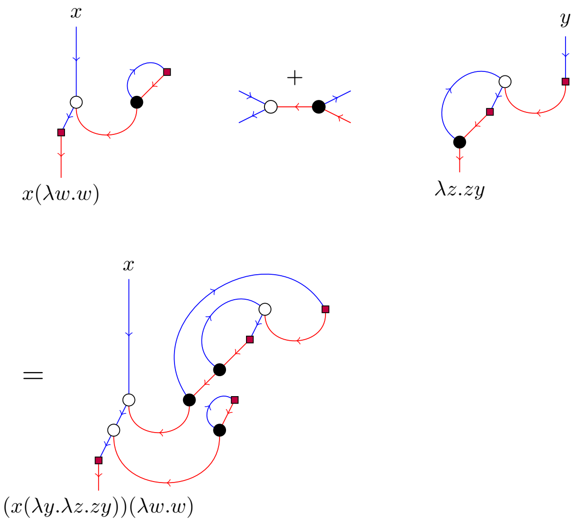
With annotations:



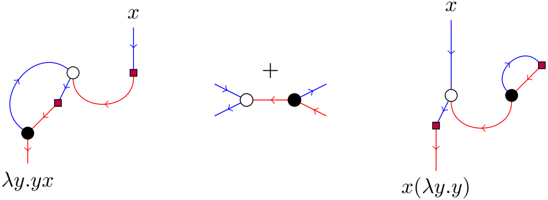
For example, consider combining $x(\lambda y.y)$ and $\lambda y.yx$ in either order...


x
$x(\lambda y.y)$
$+$
x
$\lambda y.yx$

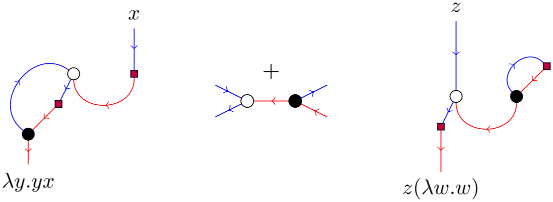

x
$x(\lambda w.w)$
$+$
y
$\lambda z.zy$

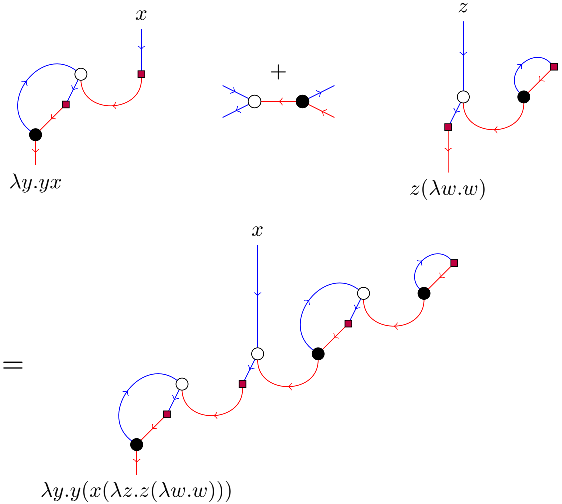


x
$x(\lambda w.w)$
$+$
y
$\lambda z.zy$
$=$
x
$(x(\lambda y.\lambda z.zy))(\lambda w.w)$



x
λy.yx
+
x
x(λy.y)

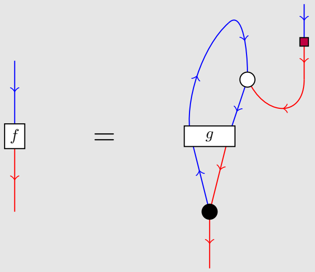

x
$\lambda y.yx$
$+$
z
$z(\lambda w.w)$


x
z
$+$
$\lambda y.yx$
$z(\lambda w.w)$
$=$
x
$\lambda y.y(x(\lambda z.z(\lambda w.w)))$

The argument-open case

Proposition

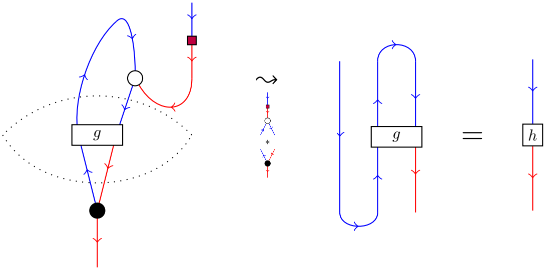
The diagram f of any argument-open NPTV factors as



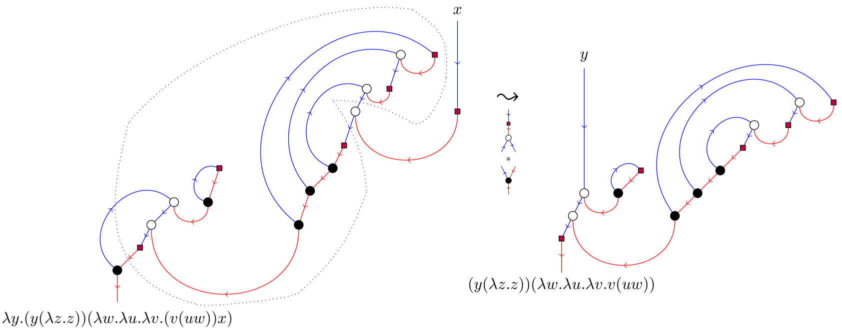
for some g.

The argument-open case

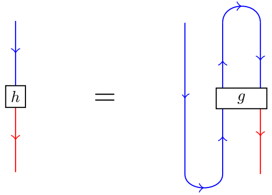
Surgery on an argument-open NPTV recovers a NPTV:



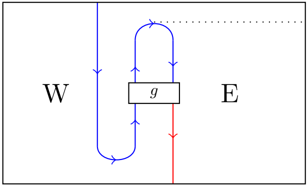
For example, here is a demonstration of surgery on the diagram of an argument-open NPTV of size 6 (yielding a function-open NPTV of size 5)...

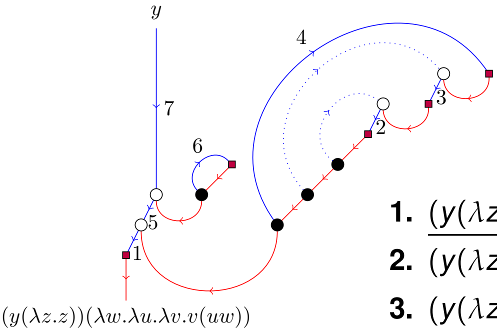

x
y
λy.(y(λz.z))(λw.λu.λv.(v(uw))x)
(y(λz.z))(λw.λu.λv.v(uw))

Argument-open surgery is only reversible given extra information: we have to *factor h* along a *valence wire*...



The **valence wires** of a diagram are the blue wires reachable from the East:





1. $\underline{(y(\lambda z.z))(\lambda w.\lambda u.\lambda v.v(uw))}$
2. $(y(\lambda z.z))(\lambda w.\lambda u.\lambda v.\underline{v(uw)})$
3. $(y(\lambda z.z))(\lambda w.\lambda u.\lambda v.v(\underline{uw}))$
4. $(y(\lambda z.z))(\lambda w.\lambda u.\lambda v.v(u\underline{w}))$
5. $(\underline{y(\lambda z.z)})(\lambda w.\lambda u.\lambda v.v(uw))$
6. $(y(\lambda z.\underline{z}))(\lambda w.\lambda u.\lambda v.v(uw))$
7. $(\underline{y}(\lambda z.z))(\lambda w.\lambda u.\lambda v.v(uw))$

Performing reverse surgery while focused on any of these subterms yields a different argument-open NPTV:

$$\underline{(y(\lambda z.z))(\lambda w.\lambda u.\lambda v.v(uw))} \rightsquigarrow \lambda y.((y(\lambda z.z))(\lambda w.\lambda u.\lambda v.v(uw)))x$$

$$(y(\lambda z.z))(\lambda w.\lambda u.\lambda v.\underline{v(uw)}) \rightsquigarrow \lambda y.(y(\lambda z.z))(\lambda w.\lambda u.\lambda v.(v(uw))x)$$

$$(y(\lambda z.z))(\lambda w.\lambda u.\lambda v.v(\underline{uw})) \rightsquigarrow \lambda y.(y(\lambda z.z))(\lambda w.\lambda u.\lambda v.v((uw)x))$$

$$(y(\lambda z.z))(\lambda w.\lambda u.\lambda v.v(u\underline{w})) \rightsquigarrow \lambda y.(y(\lambda z.z))(\lambda w.\lambda u.\lambda v.v(u(wx)))$$

$$(\underline{y(\lambda z.z)})(\lambda w.\lambda u.\lambda v.v(uw)) \rightsquigarrow \lambda y.((y(\lambda z.z))x)(\lambda w.\lambda u.\lambda v.v(uw))$$

$$(y(\lambda z.\underline{z}))(\lambda w.\lambda u.\lambda v.v(uw)) \rightsquigarrow \lambda y.(y(\lambda z.zx))(\lambda w.\lambda u.\lambda v.v(uw))$$

$$(\underline{y}(\lambda z.z))(\lambda w.\lambda u.\lambda v.v(uw)) \rightsquigarrow \lambda y.((yx)(\lambda z.z))(\lambda w.\lambda u.\lambda v.v(uw))$$

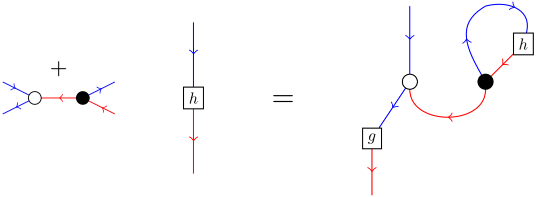
Bijection

Theorem

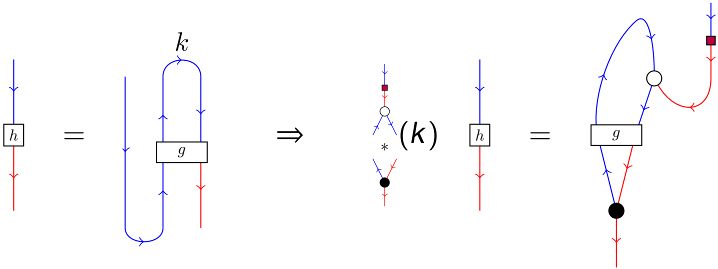
Let t be a NPTV of size $|\pi|$ and valence $v(\pi)$. Then exactly one of the following cases must hold:

- **(i)** *t is the variable term and $|\pi| = v(\pi) = 1$.*
- **(ii)** *$t = c_F(t_1, t_2)$ for some t_1 and t_2 (and π_1 and π_2) such that $|\pi| = |\pi_1| + |\pi_2|$ and $v(\pi) = 1 + v(\pi_1) + v(\pi_2)$.*
- **(iii)** *$t = c_A^{(k)}(t_1)$ for some t_1 (and π_1) and $1 \leq k \leq v(\pi_1)$ such that $|\pi| = 1 + |\pi_1|$ and $v(\pi) = k + 1$.*

Verify size and valence constraints by inspection...



$$|\pi| = |\pi_1| + |\pi_2| \quad \text{and} \quad v(\pi) = 1 + v(\pi_1) + v(\pi_2)$$



$$|\pi| = 1 + |\pi_1| \quad \text{and} \quad v(\pi) = k + 1$$

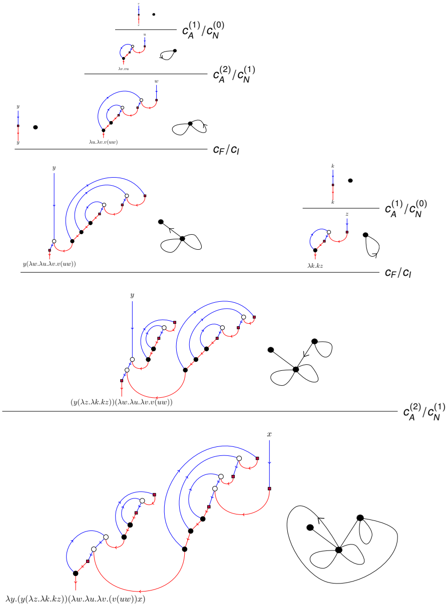
Corollary

There is a bijection between rooted planar maps with n edges and outer face degree d, and NPTVs of size $n+1$ and valence $d+1$.

Corollary

The following sets are all in one-to-one correspondence:

- *rooted planar maps*
- *normal planar lambda terms with one free variable*
- *closed normal planar lambda terms*



$c_A^{(1)}/c_N^{(0)}$
$\lambda v.vu$
$c_A^{(2)}/c_N^{(1)}$
$\lambda u.\lambda v.v(uw)$
c_F/c_I
$c_A^{(1)}/c_N^{(0)}$
$y(\lambda w.\lambda u.\lambda v.v(uw))$
$\lambda k.kz$
c_F/c_I
$(y(\lambda z.\lambda k.kz))(\lambda w.\lambda u.\lambda v.v(uw))$
$c_A^{(2)}/c_N^{(1)}$
$\lambda y.(y(\lambda z.\lambda k.kz))(\lambda w.\lambda u.\lambda v.(v(uw))x)$