8 March 2007

2-Categories from Typed λ-calculi

Let P be a typed λ-calculus. We've seen how to get a cartesian closed category out of it. Now let's start trying to get a 2-category out of it, where what were <u>equations</u> between morphism become <u>2-morphisms</u>. These represent processes of computation. For example, if $P = \lambda Th(Calc)$ we have terms and equations between them – the product rule, chain rule, etc. – but now we'll treat them as "rewrite rules", e.g.

$$D(fg) \Longrightarrow (Df)g + f(Dg)$$

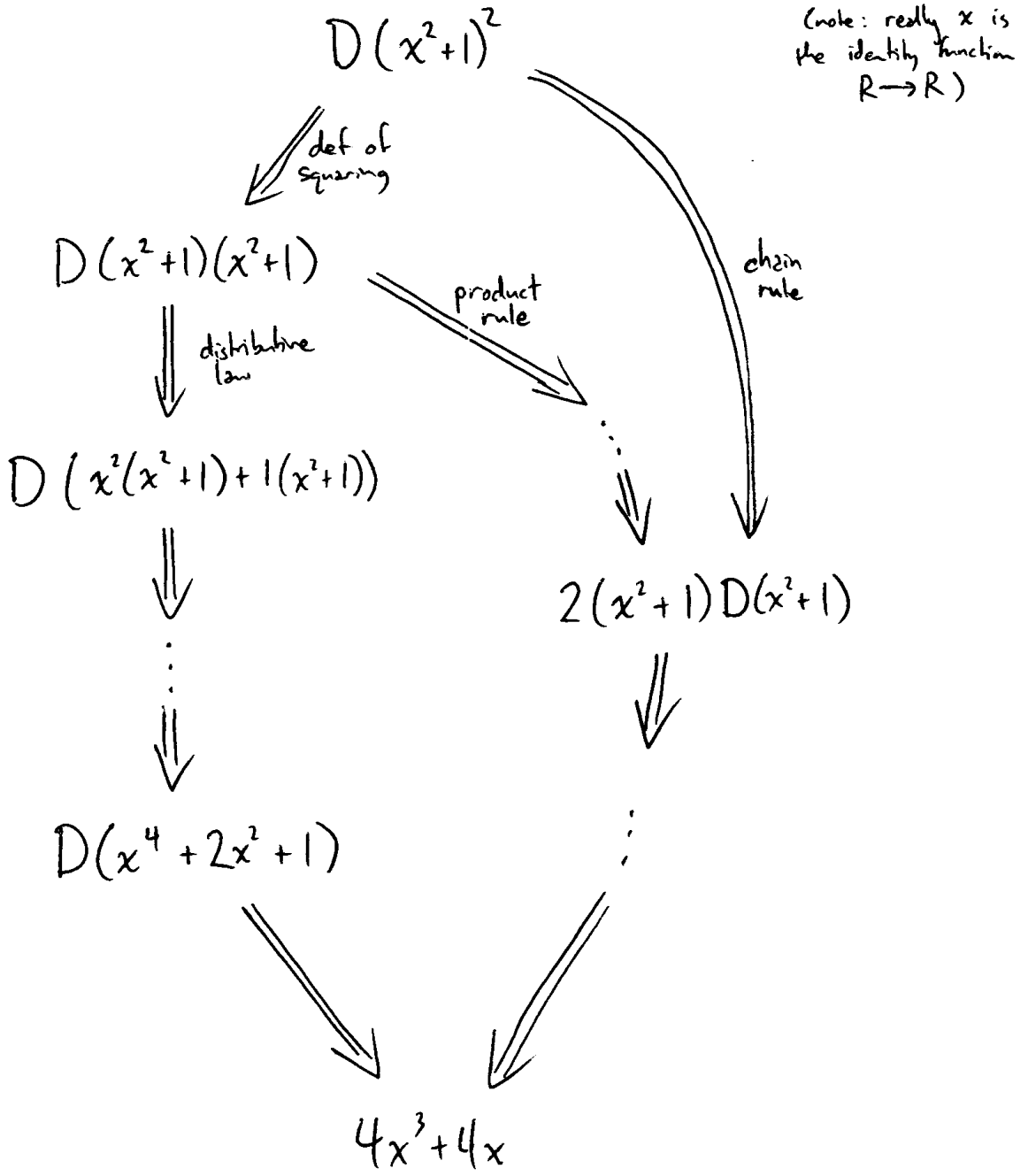
(in human-friendly form), or

$$D(x \mapsto \cdot(f(x), g(x))) \Longrightarrow x \mapsto +\left(\cdot((Df)(x), g(x)), \cdot(f(x), (Dg)(x))\right)$$

(according to official rules)

Then for any type A, we get not a set of terms of that type but a category freely generated by these rewrite rules.

For example, if $A = \hom(R, R)$ then we have



Part of the implicit faith in high school calculus is that these categories (one for each type) are terminating and confluent.

This says you can just keep applying rules however you want and wind up with "the right answer." Actually this isn't quite true:

$$x+1 \Downarrow 1+x \Downarrow x+1 \Downarrow \vdots$$

not terminating!

To fix this you could modify the category so the objects were certain equivalence classes of terms by taking <u>some</u> rewrite rules as <u>equations</u>. For any typed λ-calculus P, Lambek & Scott define a <u>poset</u> of terms of a given type — this is a category with at most one morphism between any two objects s.t. given $f: x \to y$, $g: y \to x$ we have $x = y$. Their poset is freely generated <u>as a poset</u> by these rewrite rules:

1) $\pi_1(a,b) \Longrightarrow a$ $\quad a \in A,\ b \in B$

In C_P both these terms yield morphisms; now we're forming a 2-category $\widetilde{C}_P$ where there's a 2-morphism

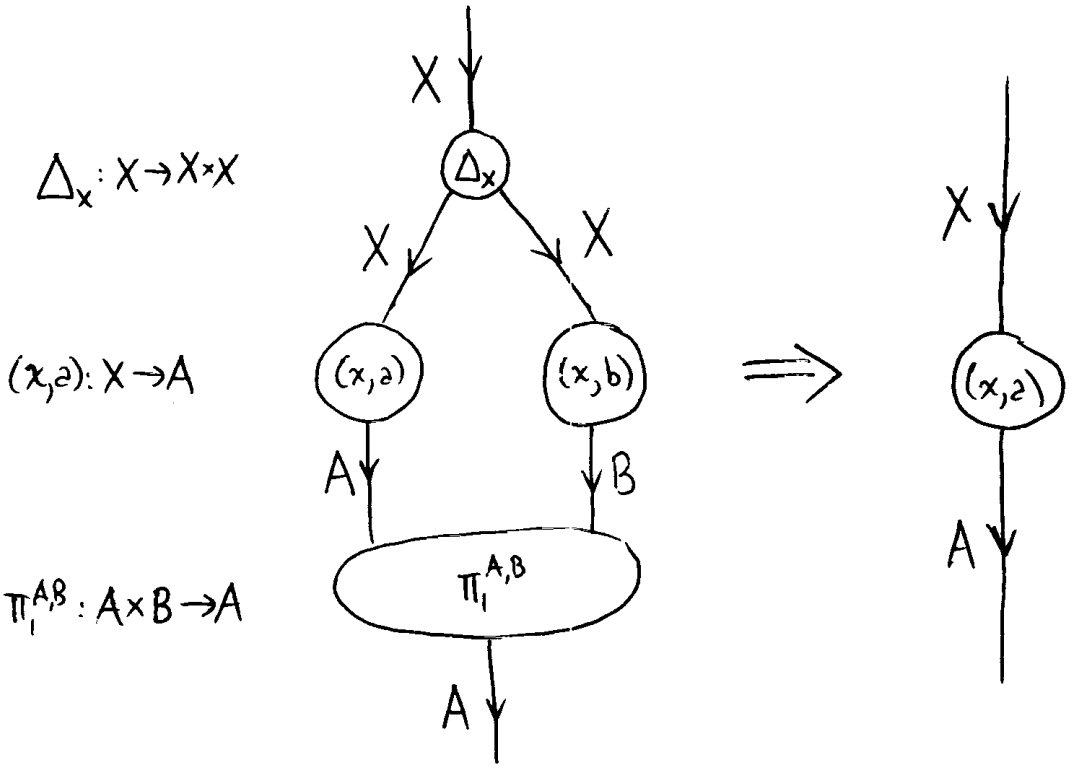
between them. In C_p we have morphisms

$$(x \in X, \pi_1(a,b)) = (x \in X, a)$$

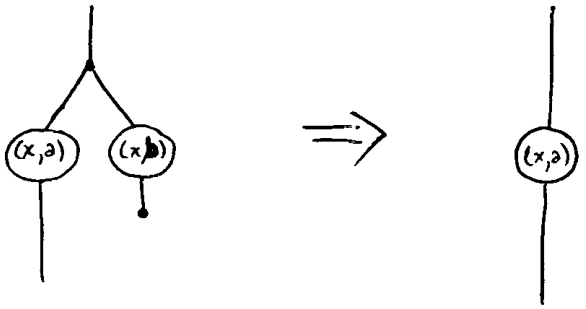
but in $\tilde{C}_p$ there's a 2-morphism

$$(x \in X, \pi_1(a,b)) \Longrightarrow (x \in X, a)$$

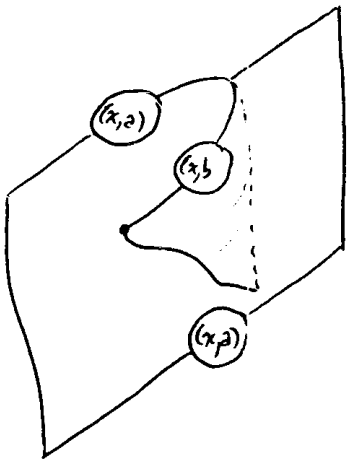
Let's draw this using string diagrams:



or for short:



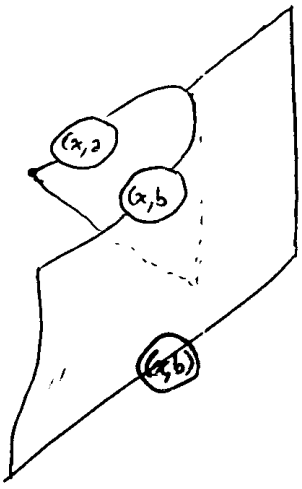
which can be drawn as a surface diagram



Similarly:

2) $\pi_2(a,b) \Longrightarrow b$

gives



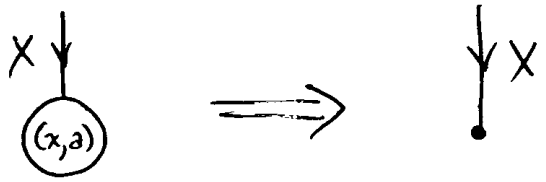
Next:

3) $a \Longrightarrow *$ $\qquad a \in 1$

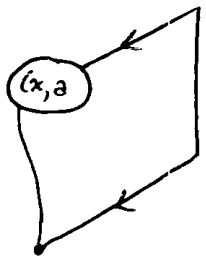
gives a 2-morphism

$$(x \in X, a) \Longrightarrow (x \in X, *)$$

which in string diagrams becomes



and as a surface diagram:

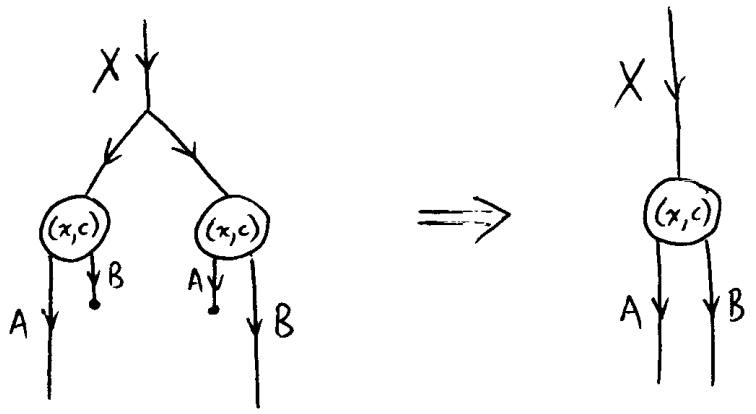


4) $(\pi_1(c), \pi_2(c)) \Longrightarrow c$

This gives a 2-morphism

$$(x \in X, (\pi_1(c), \pi_2(c))) \Longrightarrow (x \in X, c)$$

which in string diagrams becomes:



5) β-reduction:

$$(x \in X \longmapsto \phi(x))(c) \Longrightarrow \phi(c)$$

$\phi(x) \in A$
$c \in X$

6) η-reduction:

$$(x \in X \longmapsto f(x)) \Longrightarrow f$$

$f \in \hom(X, A)$
x not free in f