The main purpose of this paper is to introduce 2-categories to computer scientists as a suitable framework for certain types of semantics. Hence a few words of introduction about 2-categories might be suitable. A 2-category is a category enriched with some extra structure: the hom-sets are themselves categories. (This generalises the familiar context in which one has a category, e.g of domains, in which each hom-set inherits a partial order from the orders on the domains. Since a poset is a category, every such category of domains is in fact a 2-category.) The morphisms between morphisms are called "2-cells". There are various axioms which guarantee that the categorical structures on the hom-sets "mesh" well with the original categorical structure (of objects and morphisms). Of course it is my point that the general flavour of this may be gleaned from considering the (typed) lambda calculus: thinking of types as "objects", it is easy to see how a term a of type A with exactly one free variable x of type B may be considered as a map ("morphism") $B \longrightarrow A$. Given another term b of type A with exactly one free variable x of type B, so $b\colon B \longrightarrow A$ also, then a "2-cell" $p\colon a \Longrightarrow b$ would be a reduction from a to b. (Notice that p does not affect A, B, though A, B are implicit in any description of p ; p only acts on a to produce b.)

There is an identity reduction $a \Longrightarrow a$ for any term a, and one can compose reductions, in fact in two ways: given terms

$$a,b,c : B \longrightarrow A \quad \text{and} \quad d,e : C \longrightarrow B$$

and given reductions

$$p : a \Longrightarrow b,\ q : b \Longrightarrow c, \quad \text{and} \quad r : d \Longrightarrow e$$

there are evident compositions

$$qp : a \Longrightarrow c \ \text{(between terms } B \longrightarrow A\text{), and}$$
$$pr : ad \Longrightarrow be \ \text{(between terms } C \longrightarrow A\text{),}$$

where ad and be are defined by composition in the category of types and terms -- ie. by substitution:

$$ad = a[x:=d] \quad \text{and} \quad be = b[x:=e].$$

The main axiom of 2-categories is the "interchange law", which asserts that these two kinds of composition must commute with each other.

A final remark: initially we shall suppose that eta conversion is decreasing, rather than increasing. This follows the proof theorist's view that eta conversion is an expansion:

$$a \lesssim (\lambda\ x \text{ in } A.\ a(x)) \ \text{(for } x \text{ not free in } a\text{)}$$

rather than a reduction. As indicated above, this will allow us to regard beta and eta conversions as defining a (lax) adjunction in a fairly standard way. Later, we shall consider the effect of reversing the sense of eta conversion; this allows a different formulation of lax adjunction, but the notion of lax functor becomes less satisfactory.

Acknowledgement: As stated above, this paper is primarily intended as propaganda - 2-categories occur naturally as structures in computer science. (The interested reader should pursue this in more mathematically serious works in "2-categorical logic", particularly those from the "Australian school"; some references are given here.) I have not used the heading "Theorem", but rather "Example", since there are in fact no particularly new ideas or theorems here; this paper is based on SEELY [1979], which gives a similar analysis of first order logic. The main difference between that paper and this, is that in [1979], implication is not successfully treated, whereas here, by concentrating only on implication, those difficulties are avoided.

This work was completed with partial support from Fonds F.C.A.R., Québec.

1. 2-categorical preliminaries

We summarise here the basic notions we need from the general body of 2-category theory; a more comprehensive introduction may be found in KELLY-STREET [1974].

1.1 Definition: A 2-category $\underline{A}$ has the following structure:

(i) a collection $Ob(\underline{A})$ of objects, or 0-cells: A,B, etc.

(ii) a collection $Mor(\underline{A})$ of morphisms, or arrows or 1-cells: $a\colon B \longrightarrow A$, etc.

(iii) a collection $Cell(\underline{A})$ of 2-cells: $p\colon a \Longrightarrow b : B \longrightarrow A$, etc. also denoted

$$\begin{tikzcd} B \arrow[r, bend left, "a"] \arrow[r, bend right, "b"'] \arrow[r, phantom, "\Downarrow p"] & A \end{tikzcd}.$$

The objects and morphisms form a category $\underline{A}_o$, the "underlying category of $\underline{A}$". For fixed A,B, the morphisms $B \longrightarrow A$ and the 2-cells between them form a category Hom(B,A), also denoted $\underline{A}$(B,A). Composition in this category is known as "vertical composition":

$$\begin{tikzcd} B \arrow[r, bend left=50, "a"] \arrow[r] \arrow[r, bend right=50, "c"'] \arrow[r, phantom, shift left=2, "\Downarrow p"] \arrow[r, phantom, shift right=2, "\Downarrow q"] & A \end{tikzcd}.$$

This composite is denoted $q{\cdot}p$, or qp if no confusion results. Furthermore, given 2-cells

Proceedings of the Symposium on Logic in Computer Science, 1987
(Computer Society of the IEEE 1987) 65–71