THE JOURNAL OF SYMBOLIC LOGIC
Volume 64, Number 1, March 1999

CARTESIAN ISOMORPHISMS ARE SYMMETRIC MONOIDAL: A JUSTIFICATION OF LINEAR LOGIC

KOSTA DOŠEN AND ZORAN PETRIĆ

Abstract. It is proved that all the isomorphisms in the cartesian category freely generated by a set of objects (i.e., a graph without arrows) can be written in terms of arrows from the symmetric monoidal category freely generated by the same set of objects. This proof yields an algorithm for deciding whether an arrow in this free cartesian category is an isomorphism.

Introduction. We believe that a logic should be completely determined by its assumptions concerning structural rules. Assumptions concerning logical constants are secondary, because they should be invariable when one passes from one logic to another.

This point of view, which one may also reach on a priori grounds (see [3]), is corroborated by what one finds in the area of substructural logics–namely, logics with restricted structural rules, like intuitionistic, relevant and linear logic (see [6]). To justify a substructural logic one should first of all find a good reason for the choice of structural rules. Assumptions concerning logical constants will not differ from those one could make in classical logic. What may happen only is that classical connectives split into several nonequivalent variants. One finds this already in intuitionistic logic, where $A \rightarrow B$ and $\neg(A \wedge \neg B)$ cease to be equivalent.

All the assumptions for connectives and quantifiers of linear logic may be made in classical logic, too. Only the notorious split between so-called "multiplicative" and "additive" connectives will disappear. The assumptions for the modal operators of linear logic (called "exponentials") are also common: when they are added to a classical base of structural rules, they produce just the modal logic S4.

From this point of view, a justification of linear logic should consist primarily in finding reasons for the rejection of the structural rules of contraction and thinning. The literature on linear logic looks usually for these reasons in applications envisaged in computer science.

Our ambition is more modest. We find that there may be other reasons, of a technical nature and internal to logic, which may also serve to justify the rejection of contraction and thinning. We want to show that if among structural rules one wants to keep only those that replace collections of premises by *isomorphic* collections of premises, one should reject exactly contraction and thinning.

Received October 15, 1996; revised July 1, 1997.

Our result is about the category **Cart**, the cartesian category freely generated by a set of objects (i.e., a graph without arrows), whose arrows correspond to deductions in intuitionistic or classical conjuctive logic, and the category **SyMon**, the symmetric monoidal category freely generated by the same set of objects, whose arrows correspond to deductions in linear product logic. We talk here about ordinary, single-conclusion, deductions. Intuitionistic and classical conjunctive logics have identical deductions of this sort, and so do intuitionistic and classical variants of linear product logic.

We formulate cartesian categories in a nonstandard manner, so that arrows corresponding to structural principles (or to combinators) are primitive. (We have used a similar nonstandard formulation of cartesian categories for other purposes in [5].) The categories **Cart** and **SyMon** are syntactic categories (built up from syntactical material), whose arrows are equivalence classes of arrow-terms. We prove that every isomorphism of **Cart** can be represented by a term of a **SyMon** arrow.

This result is not unexpected, once the problem has been posed. However, the proof we shall present requires some effort. Our direct, proof-theoretical, approach has the advantage of exposing a technique of contraction and thinning elimination. It seems worth knowing that one may sometimes eliminate other structural rules besides cut. Another advantage of our proof is that it makes precise Gentzen's notion of a cluster (in German *Bund*), which is sometimes important in sequent systems. Gentzen used it in [7] for his second consistency proof of formal arithmetic, and Maehara used it in [11] for the first proof-theoretical demonstration of the embeddability of intuitionistic logic into the modal logic S4. However, in the context of these systems it is rather tricky to define clusters quite rigorously, whereas in categories a rigorous definition can be achieved with less trouble: we do it below, and call the resulting concept *progeny*.

So our proof, independently of the importance of the result proved, may perhaps also serve to improve on the arsenal of proof-theoretical techniques. This is the reason why we present it in a rather detailed form.

We said already that our axiomatization of cartesian categories is not standard: besides projections, it does not have a pairing operation on arrows as primitive, but it extends the axiomatization of symmetric monoidal categories with the diagonal natural transformation. We introduced the appropriate coherence conditions, in particular what we call the *octagonal equation*, in [5], but these matters seem to be unknown, and we produce them here again.

Let us try to dispel a possible misunderstanding of the import of our result. It may seem that we are demonstrating something only about the product fragment of linear logic, and this fragment may then be considered too restricted to be of interest. As a matter of fact, we are not demonstrating anything about any connective at all. Rather, as we explained above, we are demonstrating something about the structural basis of linear logic.

Isomorphisms between formulae, which are the objects of our categories, are interesting from a logical point of view. Two formulae are isomorphic when it is possible to find deductions leading from one to the other that when composed reduce to the trivial deduction from A to A (these reductions are represented in categories by identity of arrows). That the formulae A and A' are isomorphic is equivalent to the assertion that the deductions involving A, either as premise or as conclusion, can be extended to deductions where A is replaced by A', the deductions involving A being in one-to-one correspondence with the deductions involving A'. Isomorphism is an equivalence relation stronger than the usual mutual implication. So, for example, $A \wedge B$ is isomorphic to $B \wedge A$ in intuitionistic logic, while $A \wedge A$ only implies and is implied by A, but is not isomorphic to it. We surmise that isomorphic formulae may be taken to stand for the *same proposition*. That means reducing identity of propositions to identity of deductions.

Conjunction in intuitionistic or classical logic and product in linear logic serve to join premises in deductions into a single proposition. Our result may then be interpreted as saying that if among structural rules we keep only those that shall not replace the proposition into which the premises are joined by a different proposition, then we end up with linear logic. This is not so much a result about the connectives of conjunction and product, but rather a result about the underlying structural base of intuitionistic and classical logic on the one hand, and linear logic on the other.

In the first two sections of the paper we give our nonstandard axiomatization of the cartesian category **Cart**, taking care to separate in the second section principles concerning the terminal object. By rejecting these principles we obtain an axiomatization of the category **Cart**$^-$, which is the cartesian category without terminal object freely generated by our set of objects. As a part of the axiomatization of **Cart** we obtain the axiomatization of the symmetric monoidal category **SyMon**. By rejecting the same principles that lead from **Cart**$^-$ to **Cart**, we obtain an axiomatization of the category **SyMon**$^-$, which is the symmetric monoidal category without unit object freely generated by our set of objects.

In the third section we prove the theorem that the isomorphisms of **Cart** are expressible in **SyMon**-terms. The same theorem holds if we replace **Cart** by **Cart**$^-$ and **SyMon** by **SyMon**$^-$. The essential ingredient of the proof is an algorithm that transforms every **Cart** arrow-term expressing an isomorphism into a **SyMon** arrow-term equal to the initial **Cart** arrow-term. In that algorithm we find for every contraction a corresponding thinning, which can then be permuted so that one follows immediately the other. In that position, they can be replaced by a trivial deduction from A to A. The difficulty consists in defining precisely what it means that a contraction corresponds to a thinning, and in checking, as in a cut-elimination procedure, that thinning can be permuted with other structural rules until it reaches a corresponding contraction with which it gets eliminated. As a by-product of our proof we obtain a procedure for deciding whether an arrow of **Cart** is an isomorphism. (This is a different matter from deciding whether two objects of **Cart** are isomorphic, a procedure for which may be inferred from [14] and [1].)

In [2] one can find a number of references to matters related to the topic of this paper. This book is concerned with application in theoretical computer science.

§1. The categories Cart$^-$ and SyMon$^-$.

In this section we give a construction of the free cartesian category without terminal object, generated by a set of propositional letters, i.e., a graph without arrows whose vertices are these letters. This category is the image of this set under the left adjoint of the forgetful functor from the category of cartesian categories without terminal object into the category of graphs.

The *objects* of the category $\mathbf{Cart}^-$ are formulae built up from a set of propositional letters (whose cardinality is not important to us here) with the help of a binary connective $\cdot$, called *product.* (Usually, $\cdot$ is written $\times$ in cartesian categories and $\otimes$ in symmetric monoidal categories; we have chosen a neutral notation.) For propositional letters, which we shall sometimes call simply *letters*, we use the schemata p, $q, r, \dots$, possibly with indices, and for formulae the schemata $A, B, C, \dots$, possibly with indices.

The expression $f\colon A \vdash B$ means that f is an arrow from A to B. (We write $\vdash$ instead of the more usual $\rightarrow$ for reasons given in the introduction of [5].) The category $\mathbf{Cart}^-$ has the following *primitive arrows*, for every formula A, B and C:

$$\begin{aligned}
\mathbf{1}_A &\colon A \vdash A,\\
\overrightarrow{\mathbf{b}}_{A,B,C} &\colon A\cdot(B\cdot C) \vdash (A\cdot B)\cdot C,\\
\overleftarrow{\mathbf{b}}_{A,B,C} &\colon (A\cdot B)\cdot C \vdash A\cdot(B\cdot C),\\
\mathbf{c}_{A,B} &\colon A\cdot B \vdash B\cdot A,\\
\overrightarrow{\mathbf{k}}_{A,B} &\colon A\cdot B \vdash B,\\
\overleftarrow{\mathbf{k}}_{A,B} &\colon A\cdot B \vdash A,\\
\mathbf{w}_A &\colon A \vdash A\cdot A.
\end{aligned}$$

The arrows of $\mathbf{Cart}^-$ are built from the primitive arrows with the help of the binary *operations on arrows* of composition and product:

$$\frac{f\colon A\vdash B \qquad g\colon B\vdash C}{gf\colon A\vdash C}\,, \qquad \frac{f\colon A\vdash B \qquad g\colon C\vdash D}{f\cdot g\colon A\cdot C\vdash B\cdot D}\,.$$

(Note that composition is a partial operation, while product is total.) The expression f in $f\colon A \vdash B$ is called an *arrow-term*, or simply *term*. For terms we use the schematic letters $f, g, h, \dots$, possibly with indices. The terms of $\mathbf{Cart}^-$, defined recursively as the arrows, are called $\mathbf{Cart}^-$*-terms*. Here is a table relating our primitive arrow-terms with structural principles and combinators:

primitive arrow-term	structural principle	combinator
$\mathbf{1}_A$	identity	I
$\overrightarrow{\mathbf{b}}_{A,B,C}$, $\overleftarrow{\mathbf{b}}_{A,B,C}$	association	B
$\mathbf{c}_{A,B}$	permutation	C
$\overrightarrow{\mathbf{k}}_{A,B}$, $\overleftarrow{\mathbf{k}}_{A,B}$	thinning	K
$\mathbf{w}_A$	contraction	W

The category $\mathbf{Cart}^-$ is obtained by postulating the following *equations* between $\mathbf{Cart}^-$-terms:

Categorial equations.

(cat**1**) For $f\colon A\vdash B,\ \mathbf{1}_B f = f,\ f\mathbf{1}_A = f$.

(cat2) For $f\colon A\vdash B, g\colon B\vdash C$ and $h\colon C\vdash D,\ h(gf) = (hg)f$.

Product equations.

($\cdot$) For $f_1\colon A_1\vdash B_1, g_1\colon B_1\vdash C_1, f_2\colon A_2\vdash B_2$ and $g_2\colon B_2\vdash C_2$,

$$(g_1f_1)\cdot(g_2f_2) = (g_1\cdot g_2)(f_1\cdot f_2).$$

($\cdot$1) $\mathbf{1}_A \cdot \mathbf{1}_B = \mathbf{1}_{A \cdot B}$.

b-equations.

(**b**) For $f\colon A \vdash D$, $g\colon B \vdash E$ and $h\colon C \vdash F$,

$$((f \cdot g) \cdot h)\overrightarrow{\mathbf{b}}_{A,B,C} = \overrightarrow{\mathbf{b}}_{D,E,F}(f \cdot (g \cdot h)).$$

(**bb**) $\overrightarrow{\mathbf{b}}_{A,B,C}\overleftarrow{\mathbf{b}}_{A,B,C} = \mathbf{1}_{(A \cdot B) \cdot C}$, $\overleftarrow{\mathbf{b}}_{A,B,C}\overrightarrow{\mathbf{b}}_{A,B,C} = \mathbf{1}_{A \cdot (B \cdot C)}$.

(**b5**) $(\overrightarrow{\mathbf{b}}_{A,B,C} \cdot \mathbf{1}_D)\overrightarrow{\mathbf{b}}_{A,B \cdot C,D}(\mathbf{1}_A \cdot \overrightarrow{\mathbf{b}}_{B,C,D})\overleftarrow{\mathbf{b}}_{A,B,C \cdot D} = \overrightarrow{\mathbf{b}}_{A \cdot B,C,D}$.

c-equations.

(**c**) For $f\colon A \vdash C$ and $g\colon B \vdash D$,

$$(g \cdot f)\mathbf{c}_{A,B} = \mathbf{c}_{C,D}(f \cdot g).$$

(**cc**) $\mathbf{c}_{B,A}\mathbf{c}_{A,B} = \mathbf{1}_{A \cdot B}$.

(**bc6**) $\overleftarrow{\mathbf{b}}_{C,A,B}(\mathbf{c}_{A,C} \cdot \mathbf{1}_B)\overrightarrow{\mathbf{b}}_{A,C,B}(\mathbf{1}_A \cdot \mathbf{c}_{B,C})\overleftarrow{\mathbf{b}}_{A,B,C} = \mathbf{c}_{A \cdot B,C}$.

k-equations.

(**k**) For $f\colon A \vdash C$ and $g\colon B \vdash D$,

$$g\overrightarrow{\mathbf{k}}_{A,B} = \overrightarrow{\mathbf{k}}_{C,D}(f \cdot g), \qquad f\overleftarrow{\mathbf{k}}_{A,B} = \overleftarrow{\mathbf{k}}_{C,D}(f \cdot g).$$

(**bk**) $(\overleftarrow{\mathbf{k}}_{A,B} \cdot \mathbf{1}_C)\overrightarrow{\mathbf{b}}_{A,B,C} = \mathbf{1}_A \cdot \overrightarrow{\mathbf{k}}_{B,C}$.

(**ck**) $\overrightarrow{\mathbf{k}}_{B,A}\mathbf{c}_{A,B} = \overleftarrow{\mathbf{k}}_{A,B}$.

w-equations.

(**w**) For $f\colon A \vdash B$,

$$(f \cdot f)\mathbf{w}_A = \mathbf{w}_B f.$$

(**bw**) $\overrightarrow{\mathbf{b}}_{A,A,A}(\mathbf{1}_A \cdot \mathbf{w}_A)\mathbf{w}_A = (\mathbf{w}_A \cdot \mathbf{1}_A)\mathbf{w}_A$.

(**cw**) $\mathbf{c}_{A,A}\mathbf{w}_A = \mathbf{w}_A$.

(**bcw8**) If $\mathbf{c}^m_{A,B,C,D} =_{\mathrm{df}} \overrightarrow{\mathbf{b}}_{A,C,B \cdot D}(\mathbf{1}_A \cdot (\overleftarrow{\mathbf{b}}_{C,B,D}(\mathbf{c}_{B,C} \cdot \mathbf{1}_D)\overrightarrow{\mathbf{b}}_{B,C,D}))\overleftarrow{\mathbf{b}}_{A,B,C \cdot D}$,

$$\mathbf{c}^m_{A,A,B,B}(\mathbf{w}_A \cdot \mathbf{w}_B) = \mathbf{w}_{A \cdot B}.$$

(**kw**) $\overrightarrow{\mathbf{k}}_{A,A}\mathbf{w}_A = \mathbf{1}_A$, $\overleftarrow{\mathbf{k}}_{A,A}\mathbf{w}_A = \mathbf{1}_A$.

The product equations say that product is a bifunctor. The (**b**), (**bb**), (**c**) and (**cc**) equations say that the $\overrightarrow{\mathbf{b}}$ and $\overleftarrow{\mathbf{b}}$ arrows, which we call simply **b** arrows, and the **c** arrows are natural isomorphisms. The equation (**b5**) is one of Mac Lane's *pentagonal* equations and (**bc6**) is one of Mac Lane's *hexagonal* equations.

The equations (**k**) and (**w**) say that the $\overrightarrow{\mathbf{k}}$ and $\overleftarrow{\mathbf{k}}$ arrows (which we call **k** arrows) and the **w** arrows are natural transformations. The equation (**bk**) is related to Mac Lane's *triangular* equations of [10, VII.1, p. 159]. From it we can derive (without using **c** arrows) the following two analogous triangular equations:

($\mathbf{b}\overrightarrow{\mathbf{k}}$) $(\overrightarrow{\mathbf{k}}_{A,B} \cdot \mathbf{1}_C)\overrightarrow{\mathbf{b}}_{A,B,C} = \overrightarrow{\mathbf{k}}_{A,B \cdot C}$

($\mathbf{b}\overleftarrow{\mathbf{k}}$) $(\mathbf{1}_A \cdot \overleftarrow{\mathbf{k}}_{B,C})\overleftarrow{\mathbf{b}}_{A,B,C} = \overleftarrow{\mathbf{k}}_{A \cdot B,C}$

(cf. [10, VII.1, p. 161, Exercise 1]), which we use below. The equation (**bcw8**) is one of the *octagonal* equations of [5].

We haven't economized in the above axiomatization of **Cart**$^-$ (some primitive arrows are definable in terms of others and some of the equations are superfluous), but this axiomatization has the advantage of extending the standard axiomatization of symmetric monoidal categories.

The category **SyMon**$^-$ is obtained by rejecting in our axiomatization of **Cart**$^-$ the **k** and **w** arrows and the **k** and **w**-equations. Everything else is as for **Cart**$^-$. It is easy to see that all the arrows of **SyMon**$^-$ are components of natural isomorphisms, whereas the arrows of **Cart**$^-$ are components of natural transformations, but not necessarily isomorphisms. (The operations on arrows, composition and product, preserve natural transformations and isomorphisms.)

§2. **The categories Cart and SyMon.** The category **Cart** is obtained from **Cart**$^-$ by assuming that we have in our propositional language a propositional constant (nullary connective) I. In addition to the primitive arrows of **Cart**$^-$ we also have the primitive arrows:

$$\sigma_{\mathrm{I},A}\colon \mathrm{I}\cdot A \vdash A, \qquad \delta_{A,\mathrm{I}}\colon A\cdot \mathrm{I} \vdash A,$$
$$\sigma^i_A\colon A \vdash \mathrm{I}\cdot A, \qquad \delta^i_A\colon A \vdash A\cdot \mathrm{I},$$

which we call $\sigma\delta$ arrows. These arrows say that I is a unit object. (The index I of $\sigma_{\mathrm{I},A}$ and $\delta_{A,\mathrm{I}}$ may seem superfluous, but we need it for technical reasons, as it will become clear in the next section.) The operations on arrows are as in **Cart**$^-$, and **Cart**-*terms* are defined analogously. In addition to the equations we had for **Cart**$^-$-terms, now understood as equations between **Cart**-terms, we also have:

$\sigma\delta$-equations.
(σ) For $g\colon B\vdash D$, $g\sigma_{\mathrm{I},B} = \sigma_{\mathrm{I},D}(\mathbf{1}_\mathrm{I}\cdot g)$.
(δ) For $f\colon A\vdash C$, $f\delta_{A,\mathrm{I}} = \delta_{C,\mathrm{I}}(f\cdot \mathbf{1}_\mathrm{I})$.
($\sigma\sigma^i$) $\sigma_{\mathrm{I},A}\sigma^i_A = \mathbf{1}_A$, $\sigma^i_A\sigma_{\mathrm{I},A} = \mathbf{1}_{\mathrm{I}\cdot A}$.
($\delta\delta^i$) $\delta_{A,\mathrm{I}}\delta^i_A = \mathbf{1}_A$, $\delta^i_A\delta_{A,\mathrm{I}} = \mathbf{1}_{A\cdot\mathrm{I}}$.
($\sigma\delta$) $\sigma_{\mathrm{I},\mathrm{I}} = \delta_{\mathrm{I},\mathrm{I}}$.
($\sigma\delta\mathbf{b}$) $(\delta_{A,\mathrm{I}}\cdot \mathbf{1}_C)\overrightarrow{\mathbf{b}}_{A,\mathrm{I},C} = \mathbf{1}_A\cdot\sigma_{\mathrm{I},C}$.
($\sigma\delta\mathbf{c}$) $\sigma_{\mathrm{I},A}\mathbf{c}_{A,\mathrm{I}} = \delta_{A,\mathrm{I}}$.
($\sigma\delta\mathbf{k}$) $\sigma_{\mathrm{I},A} = \overrightarrow{\mathbf{k}}_{\mathrm{I},A}$, $\delta_{A,\mathrm{I}} = \overleftarrow{\mathbf{k}}_{A,\mathrm{I}}$.
($\sigma\delta\mathbf{w}$) $\mathbf{w}_\mathrm{I} = \sigma^i_\mathrm{I}$.
(The equation ($\sigma\delta\mathbf{w}$) can be $\mathbf{w}_\mathrm{I} = \delta^i_\mathrm{I}$, as well.)

In **Cart** most of the $\sigma\delta$-equations are superfluous. The equations (σ), ($\sigma\sigma^i$), (δ) and ($\delta\delta^i$) say that σ and δ arrows are natural isomorphisms; the others serve as appropriate coherence conditions.

The category **SyMon** is obtained by rejecting in our axiomatization of **Cart** the **k** and **w** arrows, the **k** and **w**-equations, and the $\sigma\delta$-equations ($\sigma\delta\mathbf{k}$) and ($\sigma\delta\mathbf{w}$). This axiomatization of the free symmetric monoidal category **SyMon** follows closely the postulates of Mac Lane [10, VII.1–7]. The notion of **SyMon**-term is analogous to the notion of **Cart**-term, but based on the arrows of **SyMon**. All the arrows of **SyMon** are components of natural isomorphisms, whereas the arrows of **Cart** are components of natural transformations, but not necessarily isomorphisms.

§3. **The isomorphisms of Cart are expressible by SyMon-terms.** To prove our theorem about the isomorphisms of **Cart** we need a number of definitions and some preliminary lemmata.

From now on we call **Cart**-terms simply *terms*, since this is the largest class of arrow-terms we shall envisage. We speak about **Cart**-terms only if there is a danger of confusion. A *product term* is a term defined recursively as follows:

(0) The primitive terms

$$\overrightarrow{\mathbf{b}}_{A,B,C},\quad \overleftarrow{\mathbf{b}}_{A,B,C},\quad \mathbf{c}_{A,B},\quad \boldsymbol{\sigma}_{\mathrm{I},A},\quad \boldsymbol{\sigma}^i_A,\quad \boldsymbol{\delta}_{A,\mathrm{I}},\quad \boldsymbol{\delta}^i_A,\quad \overrightarrow{\mathbf{k}}_{p,A},\quad \overleftarrow{\mathbf{k}}_{A,p},\quad \mathbf{w}_p$$

are product terms, called *determining factors*.

(1) The terms $\mathbf{1}_A$ are product terms.

(2) If f is a product term, then $\mathbf{1}_A\cdot f$ and $f\cdot\mathbf{1}_A$ are product terms.

The determining factor of a product term f, if it exists, is denoted by $d(f)$. A product term f is a **b**-product if and only if $d(f)$ is a **b** term, **c**-product if and only if $d(f)$ is a **c** term, and similarly for $\boldsymbol{\sigma}$, $\boldsymbol{\sigma}^i$, $\boldsymbol{\delta}$, $\boldsymbol{\delta}^i$, **k** and **w**-products; it is a **1**-product if and only if $d(f)$ does not exist.

A term is *developed* if and only if it is of the form $f_n\dots f_1$, $n\geq 1$, where each f_i, $1\leq i\leq n$, is a product term and f_n is a **1**-product.

In a developed term compositions are not nested in products, and arrows corresponding to structural rules that are not identity don't work in a parallel manner in products, but one after another. Moreover, **k** and **w** terms of the form $\overrightarrow{\mathbf{k}}_{B,C}$, $\overleftarrow{\mathbf{k}}_{C,B}$ and $\mathbf{w}_B$ where the formula B is nonatomic or I have to be replaced by $\overrightarrow{\mathbf{k}}_{p,A}$, $\overleftarrow{\mathbf{k}}_{A,p}$ and $\mathbf{w}_p$, or $\boldsymbol{\sigma}_{\mathrm{I},A}$, $\boldsymbol{\delta}_{A,\mathrm{I}}$ and $\boldsymbol{\sigma}^i_{\mathrm{I}}$. We can prove the following:

DEVELOPMENT LEMMA. *Every term is equal to a developed term.*

PROOF. Start with a given term. First we use the equations:

$$\overrightarrow{\mathbf{k}}_{A\cdot B,C}=\overrightarrow{\mathbf{k}}_{B,C}(\overrightarrow{\mathbf{k}}_{A,B}\cdot\mathbf{1}_C),\qquad \overleftarrow{\mathbf{k}}_{C,A\cdot B}=\overleftarrow{\mathbf{k}}_{C,A}(\mathbf{1}_C\cdot\overleftarrow{\mathbf{k}}_{A,B}),$$

which hold by (**k**) and ($\boldsymbol{\sigma\delta}$**k**), until we are left only with $\overrightarrow{\mathbf{k}}_{p,A}$ and $\overleftarrow{\mathbf{k}}_{A,p}$. Next we use (**bcw8**) and ($\boldsymbol{\sigma\delta}$**w**) until we are left only with $\mathbf{w}_p$. Finally, we use ($\cdot$) and ($\cdot$**1**) to get a developed term equal to the original term.

For example, the term

$$\overrightarrow{\mathbf{k}}_{p\cdot\mathrm{I},p}(\mathbf{c}_{\mathrm{I},p}\cdot(\overleftarrow{\mathbf{k}}_{p,\mathrm{I}}\mathbf{c}_{\mathrm{I},p}))$$

is developed into

$$\mathbf{1}_p\boldsymbol{\sigma}_{\mathrm{I},p}(\overrightarrow{\mathbf{k}}_{p,\mathrm{I}}\cdot\mathbf{1}_p)(\mathbf{c}_{\mathrm{I},p}\cdot\mathbf{1}_p)(\mathbf{1}_{\mathrm{I}\cdot p}\cdot\boldsymbol{\delta}_{p,\mathrm{I}})(\mathbf{1}_{\mathrm{I}\cdot p}\cdot\mathbf{c}_{\mathrm{I},p}),$$

and the term $\mathbf{w}_{\mathrm{I}\cdot p}$ into

$$\mathbf{1}_{(\mathrm{I}\cdot p)\cdot(\mathrm{I}\cdot p)}\overrightarrow{\mathbf{b}}_{\mathrm{I},p,\mathrm{I}\cdot p}(\mathbf{1}_{\mathrm{I}}\cdot\overleftarrow{\mathbf{b}}_{p,\mathrm{I},p})(\mathbf{1}_{\mathrm{I}}\cdot(\mathbf{c}_{\mathrm{I},p}\cdot\mathbf{1}_p))$$
$$(\mathbf{1}_{\mathrm{I}}\cdot\overrightarrow{\mathbf{b}}_{\mathrm{I},p,p})\overleftarrow{\mathbf{b}}_{\mathrm{I},\mathrm{I},p\cdot p}(\boldsymbol{\sigma}^i_{\mathrm{I}}\cdot\mathbf{1}_{p\cdot p})(\mathbf{1}_{\mathrm{I}}\cdot\mathbf{w}_p).\qquad\dashv$$

Note that the developed term of an arrow $f\colon A\vdash B$ where no propositional letter occurs in A, but only I, is a **SyMon**-term. We call such arrows I-*arrows*. (Note that for an I-arrow $f\colon A\vdash B$, the formula B is also made only of I, there being no arrow in **Cart** of the type $\mathrm{I}\vdash C$ where propositional letters occur in C.)

We could give the Development Lemma in a sharper form by eliminating with the help of (**bc6**) every $\mathbf{c}_{A,B}$ where A and B are not atomic, ending up only with those where A and B are atomic. (We can even replace $\mathbf{c}_{\mathrm{I},\mathrm{I}}$ by $\mathbf{1}_{\mathrm{I}\cdot\mathrm{I}}$.) This would simplify a little bit some of our definitions, but is not essential for our proof.

Note that if there are no occurrences of **b** and **c** terms in a term we want to bring into a developed form, they may be introduced only by applications of (**bcw8**). So if we deal with a formulation of **Cart** where **b** and **c** arrows are not primitive, but defined in terms of **k** and **w** arrows, and we want to show that a **Cart**-term is equal to a **SyMon**-term, the eventual presence of **b** and **c** terms in our term will be made explicit through the agency of (**bcw8**). If our term is equal to a **SyMon**-term, the remaining **k** and **w** terms in its developed form will be eliminated by a procedure described in the Maximal-Path Lemma 2 below, while **b** and **c** terms introduced by (**bcw8**) may remain.

Now we introduce a series of definitions leading to an analogue of Gentzen's notion of a *cluster* from [7, §3.41] (cf. [11, §2.621] and [4, §5, pp. 312–313]).

In $\overrightarrow{\mathbf{b}}_{A,B,C}$ the occurrences of formulae A, B and C are the *indices* of this term. Note that some of A, B and C might be occurrences of the same formula. We define indices similarly for other primitive arrow-terms. Let $f_n \dots f_1$, $n \geq 1$, be a developed term. Take the indices in f_i in the order in which they occur and delete product signs and parentheses. We call the resulting sequence the *f_i-index word.* For example, if f_i is

$$\mathbf{1}_{p\cdot q} \cdot (\overleftarrow{\mathbf{b}}_{p,(q\cdot r)\cdot \mathrm{I},r} \cdot \mathbf{1}_r)$$

the f_i-index word is $pqpqr\mathrm{I}rr$.

The *length* $l(A)$ of a formula A is the number of symbols in it after deleting product signs and parentheses. For example, $l((p \cdot \mathrm{I}) \cdot q) = 3$. The sum of the lengths of the occurrences of formulae A in the indices of the $\mathbf{1}_A$ terms left of $d(f_i)$ in f_i is called $L(f_i)$. For example,

$$L(\mathbf{1}_p \cdot (\mathbf{1}_\mathrm{I} \cdot ((\mathbf{1}_p \cdot (\mathbf{1}_{p\cdot q} \cdot \mathbf{w}_p)) \cdot \mathbf{1}_p))) = 5.$$

Take the f_i-index word $\alpha_1 \dots \alpha_k$, $i < n$, $k \geq 1$, and the f_{i+1}-index word $\beta_1 \dots \beta_m$ (it is easy to see that m can be only k, or $k+1$, or $k-1$). Then the set of *successors* of α_j, $1 \leq j \leq k$, denoted by $s(\alpha_j)$, is defined as follows:

(**1**,b) If f_i is a **1**-product or **b**-product, then $s(\alpha_j) = \{\beta_j\}$.
(Note that $m = k$.)

(**c**) If $d(f_i)$ is $\mathbf{c}_{A,B}$, then
- if $j < L(f_i) + 1$ or $j > L(f_i) + l(A) + l(B)$, then $s(\alpha_j) = \{\beta_j\}$;
- if $L(f_i) + 1 \leq j \leq L(f_i) + l(A)$, then $s(\alpha_j) = \{\beta_{j+l(B)}\}$;
- if $L(f_i) + l(A) + 1 \leq j \leq L(f_i) + l(A) + l(B)$, then $s(\alpha_j) = \{\beta_{j-l(A)}\}$.

(Note that $m = k$.)

($\overrightarrow{\mathbf{k}}$) If $d(f_i)$ is $\mathbf{k}_{p,A}$, then
- if $j < L(f_i) + 1$, then $s(\alpha_j) = \{\beta_j\}$;
- if $j = L(f_i) + 1$, then $s(\alpha_j) = \emptyset$;
- if $j > L(f_i) + 1$, then $s(\alpha_j) = \{\beta_{j-1}\}$.

(Note that $m = k - 1$.)

($\overleftarrow{\mathbf{k}}$) If $d(f_i)$ is $\mathbf{k}_{A,p}$, then we proceed as for $\overrightarrow{\mathbf{k}}_{p,A}$, replacing $L(f_i) + 1$ by $L(f_i) + l(A) + 1$.

(**w**) If $d(f_i)$ is $\mathbf{w}_p$, then
- if $j < L(f_i) + 1$, then $s(\alpha_j) = \{\beta_j\}$;
- if $j = L(f_i) + 1$, then $s(\alpha_j) = \{\beta_j, \beta_{j+1}\}$;
- if $j > L(f_i) + 1$, then $s(\alpha_j) = \{\beta_{j+1}\}$.

(Note that $m = k + 1$.)

($\boldsymbol{\sigma}$) If f_i is a $\boldsymbol{\sigma}$-product, then we proceed as in case ($\overrightarrow{\mathbf{k}}$).

($\boldsymbol{\sigma}^i$) If $d(f_i)$ is $\boldsymbol{\sigma}^i_A$, then
- if $j \leq L(f_i)$, then $s(\alpha_j) = \{\beta_j\}$;
- if $j > L(f_i)$, then $s(\alpha_j) = \{\beta_{j+1}\}$.

(Note that $m = k + 1$.)

($\boldsymbol{\delta}$) If $d(f_i)$ is $\boldsymbol{\delta}_{A,\mathrm{I}}$, then we proceed as in case ($\overleftarrow{\mathbf{k}}$).

($\boldsymbol{\delta}^i$) If $d(f_i)$ is $\boldsymbol{\delta}^i_A$, then we proceed as in case ($\boldsymbol{\sigma}^i$), replacing $L(f_i)$ by $L(f_i)+l(A)$.

For every β in the f_n-index word, $s(\beta) = \emptyset$.

Now comes our analogue of Gentzen's notion of a cluster. Take f_1 in our developed term $f_n \dots f_1$. Let the f_1-index word be $\alpha_1 \dots \alpha_k$. Take an α_j *that is not an occurrence of* I (if it exists). Then the set $P(\alpha_j)$, called a *progeny*, is defined recursively as follows:

(0) $\alpha_j \in P(\alpha_j)$;

(1) if $\beta \in P(\alpha_j)$, then $s(\beta) \subseteq P(\alpha_j)$.

For example, the developed term

$$\mathbf{1}_{(p\cdot p)\cdot q}(\mathbf{c}_{p,p}\cdot \mathbf{1}_q)(\overleftarrow{\mathbf{k}}_{p\cdot p,p}\cdot \mathbf{1}_q)(\overrightarrow{\mathbf{b}}_{p,p,p}\cdot \mathbf{1}_q)((\mathbf{1}_p\cdot \mathbf{w}_p)\cdot \mathbf{1}_q)$$

with product-terms written one below another, has the progeny $P(\alpha_2)$ made of the underlined occurrences of p:

$$\begin{array}{l}
(\mathbf{1}_p\cdot \mathbf{w}_{\underline{p}})\cdot \mathbf{1}_q\\
\overrightarrow{\mathbf{b}}_{p,\underline{p},\underline{p}}\cdot \mathbf{1}_q\\
\overleftarrow{\mathbf{k}}_{p\cdot \underline{p},\underline{p}}\cdot \mathbf{1}_q\\
\mathbf{c}_{p,\underline{p}}\cdot \mathbf{1}_q\\
\mathbf{1}_{(\underline{p}\cdot p)\cdot q}.
\end{array}$$

A progeny in a developed term $f_n \dots f_1$ is a set of occurrences of the same propositional letter. Every such occurrence is identified with two coordinates, the first being the i of the product term f_i and the second the place in the f_i-index word. Note that all progenies of a developed term are mutually disjoint.

The following definitions are related to progenies. A *path* in $P(\alpha_j)$ is an m-tuple $\langle \beta^1, \dots, \beta^m\rangle$, $m \geq 1$, such that for every i, $1 \leq i \leq m$, $\beta^i \in P(\alpha_j)$, and for every i, $1 \leq i < m$, $\beta^{i+1} \in s(\beta^i)$. (The upper index of β^i does not refer to the place in an index word, but to the place in the path; lower indices of α, β, ... are reserved for places in index words.) The *length* of a path $\langle \beta^1, \dots, \beta^m\rangle$ is m. If $\beta \in P(\alpha_j)$, then β is *terminal* if and only if $s(\beta) = \emptyset$. So, the leftmost underlined ps in the example above all belong to the same path of length 5. The underlined p in $\mathbf{1}_{(\underline{p}\cdot p)\cdot q}$ and the right underlined p in $\overleftarrow{\mathbf{k}}_{p\cdot \underline{p},\underline{p}}\cdot \mathbf{1}_q$ are terminal.

If $d(f_i)$ is $\overrightarrow{\mathbf{k}}_{p,A}$, then $\kappa(f_i)$, called the *characteristic index* of the f_i-index word is $\alpha_{L(f_i)+1}$. If $d(f_i)$ is $\overleftarrow{\mathbf{k}}_{A,p}$, then $\kappa(f_i)$ is $\alpha_{L(f_i)+l(A)+1}$. Note that in $f_n \dots f_1$ the letter $\beta \in P(\alpha_j)$ is terminal if and only if, for some i, β is $\kappa(f_i)$ or β is from the f_n-index word.

If $\mathbf{1}gf$ is developed and g is a $\mathbf{k}$-product, then f and g are *confronted* if and only if $\kappa(g)$ is a successor of an index of $d(f)$. For example, if gf is $(\overleftarrow{\mathbf{k}}_{q\cdot q,p}\cdot \mathbf{1}_p)(\overrightarrow{\mathbf{b}}_{q,q,p}\cdot \mathbf{1}_p)$, then f and g are confronted; if gf is $(\mathbf{1}_{q\cdot q}\cdot\overleftarrow{\mathbf{k}}_{p,p})\overrightarrow{\mathbf{b}}_{q,q,p\cdot p}$, then f and g are not confronted, since the right p from $\overleftarrow{\mathbf{k}}_{p,p}$ is a successor of a proper part of the index $p\cdot p$ of $\overrightarrow{\mathbf{b}}_{q,q,p\cdot p}$. Graphically, in the first case we have

$$\begin{gathered}\overrightarrow{\mathbf{b}}_{q,q,\underline{p}}\cdot\mathbf{1}_p\\ \overleftarrow{\mathbf{k}}_{q\cdot q,\underline{p}}\cdot\mathbf{1}_p,\end{gathered}$$

whereas in the second case we have

$$\begin{gathered}\overrightarrow{\mathbf{b}}_{q,q,p\cdot\underline{p}}\\ \mathbf{1}_{q\cdot q}\cdot\overleftarrow{\mathbf{k}}_{p,\underline{p}}.\end{gathered}$$

We need all these definitions to prove the following lemma, which is about things analogous to the permuting of rules in a cut-elimination procedure. In our lemma, one permutes **k**-products with other terms.

k-PERMUTATION LEMMA. *Suppose that* $\mathbf{1}gf$ *is developed, that* g *is a* **k**-*product, that* gf *is neither of the form* $\overleftarrow{\mathbf{k}}_{\mathrm{I},p}\sigma^i_p$ *nor* $\overrightarrow{\mathbf{k}}_{p,\mathrm{I}}\delta^i_p$, *and that for* f *a* **w**-*product,* f *and* g *are not confronted. Then* $gf=f'g'$ *where* g' *is a* **k**-*product,* $\mathbf{1}f'g'$ *is developed, and for* j *such that* $\kappa(g)\in s(\alpha_j)$ *in* $P(\alpha_j)$ *in* $\mathbf{1}gf$, *we have in* $\mathbf{1}f'g'$ *that* β_j *from the* g'-*index word* $\beta_1\ldots\beta_k$ *is terminal in* $P(\beta_j)$ (*i.e.,* β_j *is* $\kappa(g')$; *note that the* f-*index word coincides with the* g'-*index word and that* α_j *and* β_j *are occurrences of the same propositional letter*). *Moreover, for* α_j *in the* f-*index word that is not an occurrence of* I, *the number of paths of length* 3 *in* $P(\alpha_j)$ *in* $\mathbf{1}gf$ *is equal to the number of paths of length* 3 *in* $P(\beta_j)$ *in* $\mathbf{1}f'g'$.

PROOF. If f and g are not confronted, then we use one of the equations (cat**1**), ($\cdot$), (**b**), (**c**), (**k**), (**w**), (σ), (δ), (σ^i) and (δ^i). For example, if gf is $(\mathbf{1}_{A\cdot B}\cdot\overleftarrow{\mathbf{k}}_{C,p})\overrightarrow{\mathbf{b}}_{A,B,C\cdot p}$, then we have

$$\begin{aligned}gf&=((\mathbf{1}_A\cdot\mathbf{1}_B)\cdot\overrightarrow{\mathbf{k}}_{p,C})\overrightarrow{\mathbf{b}}_{A,B,p\cdot C}, &&\text{by }(\cdot\mathbf{1})\\ &=\overrightarrow{\mathbf{b}}_{A,B,C}(\mathbf{1}_A\cdot(\mathbf{1}_B\cdot\overrightarrow{\mathbf{k}}_{p,C})), &&\text{by }(\mathbf{b}).\end{aligned}$$

It remains to consider cases where f and g are confronted.

If f is a **b**-product, then we use the equations:

$$\begin{aligned}(\overrightarrow{\mathbf{k}}_{p,B}\cdot\mathbf{1}_C)\overrightarrow{\mathbf{b}}_{p,B,C}&=\mathbf{1}_{B\cdot C}\overrightarrow{\mathbf{k}}_{p,B\cdot C}, &&\text{by }(\mathbf{b}\overrightarrow{\mathbf{k}})\text{ and (cat}\mathbf{1}),\\ (\overleftarrow{\mathbf{k}}_{A,p}\cdot\mathbf{1}_C)\overrightarrow{\mathbf{b}}_{A,p,C}&=\mathbf{1}_{A\cdot C}(\mathbf{1}_A\cdot\overrightarrow{\mathbf{k}}_{p,C}), &&\text{by }(\mathbf{bk})\text{ and (cat}\mathbf{1}),\\ \overleftarrow{\mathbf{k}}_{A\cdot B,p}\overrightarrow{\mathbf{b}}_{A,B,p}&=\mathbf{1}_{A\cdot B}(\mathbf{1}_A\cdot\overleftarrow{\mathbf{k}}_{B,p}), &&\text{by }(\mathbf{b}\overleftarrow{\mathbf{k}}),\ (\mathbf{bb})\text{ and (cat}\mathbf{1}),\\ \overrightarrow{\mathbf{k}}_{p,B\cdot C}\overleftarrow{\mathbf{b}}_{p,B,C}&=\mathbf{1}_{B\cdot C}(\overrightarrow{\mathbf{k}}_{p,B}\cdot\mathbf{1}_C), &&\text{by }(\mathbf{b}\overrightarrow{\mathbf{k}}),(\mathbf{bb})\text{ and (cat}\mathbf{1}),\\ (\mathbf{1}_A\cdot\overrightarrow{\mathbf{k}}_{p,C})\overleftarrow{\mathbf{b}}_{A,p,C}&=\mathbf{1}_{A\cdot C}(\overleftarrow{\mathbf{k}}_{A,p}\cdot\mathbf{1}_C), &&\text{by }(\mathbf{bk}),\ (\mathbf{bb})\text{ and (cat}\mathbf{1}),\\ (\mathbf{1}_A\cdot\overleftarrow{\mathbf{k}}_{B,p})\overleftarrow{\mathbf{b}}_{A,B,p}&=\mathbf{1}_{A\cdot B}\overleftarrow{\mathbf{k}}_{A\cdot B,p}, &&\text{by }(\mathbf{b}\overleftarrow{\mathbf{k}})\text{ and (cat}\mathbf{1}).\end{aligned}$$

If f is a **c**-product, then we use the equations:

$$\begin{aligned}\overrightarrow{\mathbf{k}}_{p,A}\mathbf{c}_{A,p}&=\mathbf{1}_A\overleftarrow{\mathbf{k}}_{A,p}, &&\text{by }(\mathbf{ck})\text{ and (cat}\mathbf{1}),\\ \overleftarrow{\mathbf{k}}_{A,p}\mathbf{c}_{p,A}&=\mathbf{1}_A\mathbf{k}_{p,A}, &&\text{by }(\mathbf{ck}),\ (\mathbf{cc})\text{ and (cat}\mathbf{1}).\end{aligned}$$

If f is a $\mathbf{k}$-product, then we use the equations:

$$\overrightarrow{\mathbf{k}}_{p,A}(\overrightarrow{\mathbf{k}}_{q,p}\cdot \mathbf{1}_A) = \overrightarrow{\mathbf{k}}_{q,A}(\overleftarrow{\mathbf{k}}_{q,p}\cdot \mathbf{1}_A), \quad \text{by } (\mathbf{k}),$$
$$\overleftarrow{\mathbf{k}}_{A,p}(\mathbf{1}_A\cdot \overrightarrow{\mathbf{k}}_{q,p}) = \overleftarrow{\mathbf{k}}_{A,q}(\mathbf{1}_A\cdot \overleftarrow{\mathbf{k}}_{q,p}), \quad \text{by } (\mathbf{k}),$$

which are read from left to right and from right to left.

If f is a $\boldsymbol{\sigma}$ or $\boldsymbol{\delta}$-product, then we use the equations:

$$\overrightarrow{\mathbf{k}}_{p,A}(\boldsymbol{\sigma}_{\mathrm{I},p}\cdot \mathbf{1}_A) = \boldsymbol{\sigma}_{\mathrm{I},A}(\overleftarrow{\mathbf{k}}_{\mathrm{I},p}\cdot \mathbf{1}_A),$$
$$\overleftarrow{\mathbf{k}}_{A,p}(\mathbf{1}_A\cdot \boldsymbol{\sigma}_{\mathrm{I},p}) = \boldsymbol{\delta}_{A,\mathrm{I}}(\mathbf{1}_A\cdot \mathbf{k}_{\mathrm{I},p}),$$
$$\overrightarrow{\mathbf{k}}_{q,A}(\boldsymbol{\delta}_{q,\mathrm{I}}\cdot \mathbf{1}_A) = \boldsymbol{\sigma}_{\mathrm{I},A}(\overrightarrow{\mathbf{k}}_{q,\mathrm{I}}\cdot \mathbf{1}_A),$$
$$\overrightarrow{\mathbf{k}}_{A,q}(\mathbf{1}_A\cdot \boldsymbol{\delta}_{q,\mathrm{I}}) = \boldsymbol{\delta}_{A,\mathrm{I}}(\mathbf{1}_A\cdot \mathbf{k}_{q,\mathrm{I}}),$$

which are derived with the help of $(\mathbf{k})$ and $(\boldsymbol{\sigma\delta}\mathbf{k})$, and correspond to the equations displayed above where f is a $\mathbf{k}$-product.

If f is a $\boldsymbol{\sigma}^i$ or $\boldsymbol{\delta}^i$-product, then we use the equations:

$$(\overleftarrow{\mathbf{k}}_{\mathrm{I},p}\cdot \mathbf{1}_A)(\boldsymbol{\sigma}^i_p\cdot \mathbf{1}_A) = \boldsymbol{\sigma}^i_A\overrightarrow{\mathbf{k}}_{p,A},$$
$$(\mathbf{1}_A\cdot \overleftarrow{\mathbf{k}}_{\mathrm{I},p})(\mathbf{1}_A\cdot \boldsymbol{\sigma}^i_p) = \boldsymbol{\delta}^i_A\overleftarrow{\mathbf{k}}_{A,p},$$
$$(\overrightarrow{\mathbf{k}}_{p,\mathrm{I}}\cdot \mathbf{1}_A)(\boldsymbol{\delta}^i_p\cdot \mathbf{1}_A) = \boldsymbol{\sigma}^i_A\overrightarrow{\mathbf{k}}_{p,A},$$
$$(\mathbf{1}_A\cdot \overrightarrow{\mathbf{k}}_{p,\mathrm{I}})(\mathbf{1}_A\cdot \boldsymbol{\delta}^i_p) = \boldsymbol{\delta}^i_A\overleftarrow{\mathbf{k}}_{A,p}.$$

The left-hand side of the first of the last four equations is equal, by $(\mathbf{bb})$ and (cat$\mathbf{1}$), to

$$(\overleftarrow{\mathbf{k}}_{\mathrm{I},p}\cdot \mathbf{1}_A)\overrightarrow{\mathbf{b}}_{\mathrm{I},p,A}\overleftarrow{\mathbf{b}}_{\mathrm{I},p,A}(\boldsymbol{\sigma}^i_p\cdot \mathbf{1}_A),$$

which with the help of triangular equations and $(\boldsymbol{\sigma}^i)$ reduces to the right-hand side. We proceed similarly to derive the remaining three equations. ⊣

For the proof of the next lemma we need to define what it means to interpret **Cart** in a concrete cartesian category. For example, take as our concrete category **Set**, the cartesian category of sets. The objects of **Set** are sets, the arrows are functions between sets, $\cdot$ is cartesian product on objects, I is the singleton $\{\emptyset\}$, the arrow $\mathbf{1}_A$ is the identity map, $\mathbf{k}$ arrows are the projections, $\mathbf{w}$ arrows are diagonal maps, $\boldsymbol{\sigma}^i$ and $\boldsymbol{\delta}^i$ assign to an element a the pairs $\langle\emptyset, a\rangle$ and $\langle a, \emptyset\rangle$ respectively, $\cdot$ is defined on functions via coordinates, and the $\mathbf{b}$, $\mathbf{c}$, $\boldsymbol{\sigma}$ and $\boldsymbol{\delta}$ arrows are defined in terms of $\mathbf{k}$ and $\mathbf{w}$ arrows. A cartesian functor from **Cart** to **Set** is called an *interpretation* of **Cart** in **Set**. Such interpretations exist because of the freedom of **Cart**.

Preservation Lemma. *For every isomorphism $f: A \vdash B$ of* **Cart**, *a propositional letter occurs n times in the formula A if and only if it occurs n times in the formula B.*

Proof. Suppose p occurs n times in A and m times in B so that $m \neq n$. Take the interpretation v of **Cart** in **Set** such that $v(p) = \{\emptyset, \{\emptyset\}\}$ and, for every letter q different from p, $v(q) = \{\emptyset\}$. Then the cardinality of $v(A)$ is 2^n, which is different from the cardinality of $v(B)$, the latter being 2^m. So $v(f)$, and therefore f too, cannot be isomorphisms. (Note that v interprets **Cart** in the cartesian category of finite sets, too.) ⊣

As a corollary we have the following lemma, which makes inoffensive the restriction concerning $\boldsymbol{\sigma}^i$ and $\boldsymbol{\delta}^i$ in the **k**-Permutation Lemma:

$\boldsymbol{\sigma}^i\boldsymbol{\delta}^i$ LEMMA. *If $f_n \dots f_1$ is the developed term of an isomorphism of* **Cart**, *then for no i, $1 \leq i < n$, we can have $f_{i+1}f_i$ of the form $\overleftarrow{\mathbf{k}}_{\mathrm{I}.p}\boldsymbol{\sigma}^i_p$ or $\overrightarrow{\mathbf{k}}_{p.\mathrm{I}}\boldsymbol{\delta}^i_p$.*

PROOF. There is no arrow of **Cart** of the type $\mathrm{I} \vdash A$ where there are propositional letters in the formula A. So, if the lemma were not true, we would obtain a contradiction with the Preservation Lemma. ⊣

Let us introduce one more definition. We shall say that a **w**-product f_i in a developed term $f_n \dots f_1$ is *linked* to a progeny $P(\alpha_j)$ if and only if the index of $d(f_i)$ belongs to $P(\alpha_j)$. Similarly, a **k**-product f_i is linked to $P(\alpha_j)$ if and only if $\kappa(f_i) \in P(\alpha_j)$ (this characteristic index is terminal in $P(\alpha_j)$). It is clear that every **w** and **k**-product is linked to only one progeny. We need this notion of linkage for the proof of the following lemma, which will serve to determine where lie the **w**-product and **k**-product that shall be brought next to each other by applying the **k**-Permutation Lemma, and then eliminated.

MAXIMAL-PATH LEMMA 1. *If $f_n \dots f_1$ is the developed term of an isomorphism of* **Cart**, *then every progeny in $f_n \dots f_1$ has exactly one path of length n.*

PROOF. Note that in every progeny $P(\alpha_j)$ with m **w**-products linked to it, $m \geq 0$, there are $m+1$ terminal members. If $P(\alpha_j)$ has no path of length n, then all terminal members are characteristic indices originating in **k**-products. Consequently, if $P(\alpha_j)$ has no path of length n and there are m **w**-products linked to $P(\alpha_j)$, then there are $m+1$ **k**-products linked to $P(\alpha_j)$.

Suppose now that $P(\alpha_j)$ has no path of length n. We obtain a contradiction by induction on the number m of **w**-products linked to $P(\alpha_j)$.

If $m = 0$, there is exactly one **k**-product linked to $P(\alpha_j)$. By applying the **k**-Permutation Lemma to $f_n \dots f_1$ we push this **k**-product to the right until we obtain a developed term $g_n \dots g_1$, equal to $f_n \dots f_1$, where g_1 is a **k**-product. Note that, by the $\boldsymbol{\sigma}^i\boldsymbol{\delta}^i$ Lemma, there is no way to get $\overleftarrow{\mathbf{k}}_{\mathrm{I}.p}\boldsymbol{\sigma}^i_p$ or $\overrightarrow{\mathbf{k}}_{p.\mathrm{I}}\boldsymbol{\delta}^i_p$ while pushing our **k**-product to the right. Take an interpretation v of **Cart** in **Set** such that for the propositional letter p in $\kappa(g_1)$ we have $v(p) = \{\emptyset, \{\emptyset\}\}$, and for every other propositional letter q, $v(q) = \{\emptyset\}$. If our original isomorphism was an arrow $f_n \dots f_1 \colon A \vdash B$, then we have $g_n \dots g_1 \colon A \vdash B$. The cardinality of $v(A)$ is twice as big as the cardinality of the image of $v(A)$ under the function $v(g_1)$. Since, for no function, the cardinality of the image of a set can be greater than the cardinality of this set, the cardinality of the image of $v(A)$ under the function $v(g_n \dots g_1)$, which is equal to $v(g_n) \dots v(g_1)$, is strictly smaller than the cardinality of $v(A)$. So $f_n \dots f_1$, which is equal to $g_n \dots g_1$, is not an isomorphism. (The permuting of the **k**-product in the basis of this induction is not strictly needed, but makes the exposition simpler.)

If $m > 0$, take a **k**-product f_i in $f_n \dots f_1$, linked to $P(\alpha_j)$, and apply the **k**-Permutation Lemma. Two cases can arise. In the first case our **k**-product will never get confronted with a **w**-product while applying the **k**-Permutation Lemma. Then we push it to the extreme right and reason as in the basis of the induction. The remaining case is that we get a **k**-product g confronted with a **w**-product f in the term gf that occurs in a developed term equal to $f_n \dots f_1$. Then we replace gf by

$\mathbf{1}_C\mathbf{1}_C$ using (**kw**), $(\cdot)$ and $(\cdot\mathbf{1})$, and we apply the induction hypothesis to the resulting term. So there is at least one path of length n in $P(\alpha_j)$.

If there are two paths of length n in $P(\alpha_j)$, then, by the Preservation Lemma, there is a progeny, made of occurrences of the same letter that occurs in $P(\alpha_j)$, in which there is no path of length n, and this we have just proved to be impossible. $\dashv$

Next we prove a kind of converse of the Maximal-Path Lemma 1:

Maximal-Path Lemma 2. *If every progeny in the developed term $f_n \dots f_1$ has exactly one path of length n, then $f_n \dots f_1$ is equal to a* **SyMon**-*term.*

Proof. Suppose for $f_n \dots f_1$ that

$(*)$ in every progeny there is exactly one path of length n.

If $f_n \dots f_1$ stands for an I-arrow, then, according to the remark after the proof of the Development Lemma, it is a **SyMon**-term. If $f_n \dots f_1$ does not stand for an I-arrow, then we show that it is equal to a **SyMon**-term by induction on the number of **w**-products in it. As we have remarked at the beginning of the proof of the Maximal-Path Lemma 1, in a progeny with m **w**-products linked to it, there are $m + 1$ terminal members. From $(*)$ it follows that there are m **k**-products linked to that progeny. Therefore, since every **w** or **k**-product is linked to only one progeny, the number of **w**-products in $f_n \dots f_1$ is equal to the number of **k**-products.

Let the number m of **w**-products in $f_n \dots f_1$ be 0. As we have just shown, there are no **k**-products in $f_n \dots f_1$, and therefore this term is a **SyMon**-term.

If $m > 0$, then we take the leftmost **w**-product in $f_n \dots f_1$; call it f_i. Let $P(\alpha_j)$ be the progeny to whom f_i is linked. Then, by $(*)$, there is a **k**-product f_k, with $1 \leq i < k < n$, linked to $P(\alpha_j)$ (otherwise, we would have in $P(\alpha_j)$ at least two paths of length n). By the **k**-Permutation Lemma, we can push this **k**-product to the right until it is confronted with f_i in

$$f_n \cdots f_{k+1} f'_k \cdots f'_{i+1} f_i \cdots f_1.$$

Note that terms of the form $\overleftarrow{\mathbf{k}}_{\mathrm{I}.p}\boldsymbol{\sigma}^i_p$ or $\overrightarrow{\mathbf{k}}_{p.\mathrm{I}}\boldsymbol{\delta}^i_p$ cannot occur during the permuting of **k**-products since this permuting preserves $(*)$. Then, due to (**kw**), $(\cdot)$ and $(\cdot\mathbf{1})$, we obtain a developed term

$$f_n \cdots f_{k+1} f'_k \cdots f'_{i+2} \mathbf{1}_A \mathbf{1}_A f_{i-1} \cdots f_1$$

equal to $f_n \dots f_1$. By the induction hypothesis, this term is equal to a **SyMon**-term. $\dashv$

We can now state our main result:

Theorem. *An arrow-term denotes an isomorphism of* **Cart** *if and only if it is equal to an arrow-term of* **SyMon**.

Proof. Suppose for the only-if part that f is a term of an isomorphism of **Cart**. By the Development Lemma, f is equal to a developed term $f_n \dots f_1$, and by the Maximal-Path Lemmata 1 and 2, $f_n \dots f_1$ is equal to a **SyMon**-term.

The if part of the theorem follows immediately from the definition of **SyMon** arrows. $\dashv$

The category **SyMon** is a subcategory of **Cart** (see [13]). Having this in mind, we could reformulate the Theorem as follows: an arrow of **Cart** is an isomorphism if and only if it is an arrow of **SyMon**.

Note that all arrows of **SyMon** are isomorphisms not only in **Cart** but also in **SyMon**. In other words, **SyMon** is a groupoid in the sense of Brandt. So we could express our theorem by saying that it is the greatest groupoid subcategory of **Cart**.

Our theorem cannot be extended to arbitrary cartesian categories. For example, if we deal with a preordered cartesian category (i.e., one in which between any pair of objects there is at most one arrow), then $\mathbf{w}_A$ is an isomorphism, its inverse being $\overrightarrow{\mathbf{k}}_{A,A}$, which is equal to $\overleftarrow{\mathbf{k}}_{A,A}$ in such a category. Of course, $\mathbf{w}_A$ is not expressible by a **SyMon** term.

Our theorem can also not be extended to closed cartesian and symmetric monoidal categories. It does not hold that an arrow of the cartesian closed category **CartCl** freely generated by a set of objects is an isomorphism only if it is an arrow of the symmetric monoidal closed category **SyMonCl** freely generated by the same set (in **SyMon** every arrow is an isomorphism, but not so in **SyMonCl**). In **CartCl** we have an isomorphism between $(p \cdot q)^r$ and $p^r \cdot q^r$, but in **SyMonCl** there are arrows neither from $(p \cdot q)^r$ to $p^r \cdot q^r$ nor vice versa.

However, our theorem holds if we replace **Cart** by **Cart**$^-$ and **SyMon** by **SyMon**$^-$. It is enough to go over our proof and keep the relevant parts involving **Cart**$^-$ and **SyMon**$^-$.

The proof of our theorem provides a procedure for deciding whether an arrow in **Cart** is an isomorphism. Namely, by combining the Maximal-Path Lemmata 1 and 2 and the if part of the Theorem we obtain that $f_n \dots f_1$ is the developed term of an isomorphism of **Cart** if and only if in every progeny it has exactly one path of length n. So, to determine whether a **Cart**-term stands for an isomorphism it is enough to put it in a developed form $f_n \dots f_1$ according to the procedure described in the proof of the Development Lemma, and then check whether every progeny has exactly one path of length n.

For example, take the term

$$(\mathbf{c}_{p,p}\overleftarrow{\mathbf{k}}_{p\cdot p,p}\overrightarrow{\mathbf{b}}_{p,p,p}(\mathbf{1}_p \cdot \mathbf{w}_p)) \cdot \mathbf{1}_q.$$

It is developed into

$$\mathbf{1}_{(p\cdot p)\cdot q}(\mathbf{c}_{p,p} \cdot \mathbf{1}_q)(\overleftarrow{\mathbf{k}}_{p\cdot p,p} \cdot \mathbf{1}_q)(\overrightarrow{\mathbf{b}}_{p,p,p} \cdot \mathbf{1}_q)((\mathbf{1}_p \cdot \mathbf{w}_p) \cdot \mathbf{1}_q),$$

which we already had in the example after the definition of progeny. Every progeny in this developed term has exactly one path of length 5, and hence our original term stands for an isomorphism. According to the procedure in the proof of the Maximal-Path Lemma 2, the developed term above is transformed successively into

$$\mathbf{1}_{(p\cdot p)\cdot q}(\mathbf{c}_{p,p} \cdot \mathbf{1}_q)\mathbf{1}_{(p\cdot p)\cdot q}((\mathbf{1}_p \cdot \overleftarrow{\mathbf{k}}_{p,p}) \cdot \mathbf{1}_q)((\mathbf{1}_p \cdot \mathbf{w}_p) \cdot \mathbf{1}_q)$$
$$\mathbf{1}_{(p\cdot p)\cdot q}(\mathbf{c}_{p,p} \cdot \mathbf{1}_q)\mathbf{1}_{(p\cdot p)\cdot q}\mathbf{1}_{(p\cdot p)\cdot q}\mathbf{1}_{(p\cdot p)\cdot q}.$$

By (cat**1**), the last term stands for the isomorphism $\mathbf{c}_{p,p} \cdot \mathbf{1}_q$ of **SyMon**.

It is not essential for our proof of the Theorem that **Cart** and **Cart**$^-$ should be formulated with the official primitives of the first two sections. The alternative, more standard, formulations where **b** and **c** arrows are not primitive would do as well (though we suppose our official formulations clarify matters). However, it is essential for our proof to derive then the octagonal equation (**bcw8**), and use it as in the proof of the Development Lemma to generate arrow-terms with **b** and **c** by atomizing the indices of **w**. If the arrow of **Cart** from which we started is an isomorphism, we shall then be able to eliminate all the **k** and **w** terms according to our procedure, ending up with an arrow of **SyMon**. In all that, the role of the octagonal equation, primitive or derived, is crucial.

A simpler proof of the result of this paper may be obtained by relying on a coherence theorem for cartesian categories. This theorem says that two arrow-terms of **Cart** are equal in **Cart** if and only if they have the same *graph*, where a graph is defined by using an idea of [8, p. 94] (see also [9], [12, Theorem 2.2] and [13]). A graph of an arrow-term denoting an isomorphism of **Cart** is a bijection between two finite ordinals, and an arrow-term of **SyMon** with the same graph can always be found. We realized that there is such a simpler proof after completing this paper. As we said in the introduction, we believe that the direct, more involved, proof given here is still worth presenting because of its proof-theoretical interest.

Acknowledgements. We would like to thank Anne Preller and Djordje Čubrić for some helpful comments on an earlier version of this paper. We are also grateful to Professor Saunders Mac Lane for reminding us about some references mentioned in the last paragraph. Our work was supported by the Mathematical Institute in Belgrade through Grant 0401A.

REFERENCES

[1] K. B. Bruce, R. Di Cosmo, and G. Longo, *Provable isomorphisms of types*, ***Mathematical Structures in Computer Science***, vol. 2 (1992), pp. 231–247.

[2] R. Di Cosmo, ***Isomorphisms of types: From λ-calculus to information retrieval and language design***, Birkhäuser, Boston, 1995.

[3] K. Došen, *Logical constants as punctuation marks*, ***Notre Dame Journal of Formal Logic***, vol. 30 (1989), pp. 362–381, slightly amended version in ***What is a logical system?*** (D. M. Gabbay, editor), Oxford University Press, Oxford, 1994, pp. 273–296.

[4] ———, *Modal translations in substructural logics*, ***Journal of Philosophical Logic***, vol. 21 (1992), pp. 283–336.

[5] K. Došen and Z. Petrić, *Modal functional completeness*, ***Proof theory of modal logic*** (H. Wansing, editor), Kluwer, Dordrecht, 1996, pp. 167–211.

[6] K. Došen and P. Schroeder-Heister (editors), ***Substructural logics***, Oxford University Press, Oxford, 1993.

[7] G. Gentzen, *Neue Fassung des Widerspruchsfreiheitsbeweises für die reine Zahlentheorie*, ***Forschungen zur Logik und zur Grundlegung der exakten Wissenschaften*** (N. S.), vol. 4 (1938), pp. 19–44, English translation in ***The collected papers of Gerhard Gentzen*** (M. E. Szabo, editor), North-Holland, Amsterdam, 1969, pp. 252–286.

[8] G. M. Kelly, *Many-variable functorial calculus I*, ***Coherence in categories*** (S. Mac Lane, editor), Lecture Notes in Mathematics, no. 281, Springer-Verlag, Berlin, 1972, pp. 66–105.

[9] ———, *An abstract approach to coherence*, ***Coherence in categories*** (S. Mac Lane, editor), Lecture Notes in Mathematics, no. 281, Springer-Verlag, Berlin, 1972, pp. 106–147.

[10] S. Mac Lane, ***Categories for the working mathematician***, Springer-Verlag, Berlin, 1971.

[11] S. Maehara, *Eine Darstellung der intuitionistischen Logik in der klassischen*, ***Nagoya Mathematical Journal***, vol. 7 (1954), pp. 45–64.

[12] G. E. Mints, *Category theory and proof theory*, ***Aktual'nye voprosy logiki i metodologii nauki***, Naukova Dumka, Kiev, 1980, in Russian, pp. 252–278. English translation, with permuted title, in G. E. Mints, ***Selected papers in proof theory***, Bibliopolis, Naples, 1992.

[13] Z. Petrić, *Coherence in substructural categories*, to appear, 1997.

[14] S. V. Soloviev, *The category of finite sets and cartesian closed categories*, ***Zapiski Nauchnykh Seminarov LOMI***, vol. 105 (1981), pp. 174–194, in Russian; English translation in ***Journal of Soviet Mathematics***, vol. 22 (1983), pp. 1387–1400.

IRIT, UNIVERSITY OF TOULOUSE III
31062 TOULOUSE CEDEX, FRANCE
and
MATHEMATICAL INSTITUTE
P. O. BOX 367
11001 BELGRADE, YUGOSLAVIA

UNIVERSITY OF BELGRADE
FACULTY OF MINING AND GEOLOGY
DJUŠINA 7
11000 BELGRADE, YUGOSLAVIA