BRICS

Basic Research in Computer Science



# Open Maps, Behavioural Equivalences, and Congruences

Allan Cheng
Mogens Nielsen

BRICS Report Series RS-96-2

ISSN 0909-0878 January 1996

See back inner page for a list of recent publications in the BRICS Report Series. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK - 8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through WWW and anonymous FTP:

`http://www.brics.dk/`
`ftp ftp.brics.dk (cd pub/BRICS)`

# Open Maps, Behavioural Equivalences, and Congruences*

Allan Cheng and Mogens Nielsen

**BRICS****, Department of Computer Science, Building 540, Ny Munkegade
University of Aarhus, Aarhus 8000 C, Denmark
e-mail: {acheng,mn}@daimi.aau.dk

**Abstract.** Spans of open maps have been proposed by Joyal, Nielsen, and Winskel as a way of adjoining an abstract equivalence, $\mathcal{P}$-bisimilarity, to a category of models of computation $\mathcal{M}$, where $\mathcal{P}$ is an arbitrary subcategory of observations. Part of the motivation was to recast and generalise Milner's well-known strong bisimulation in this categorical setting. An issue left open was the *congruence properties* of $\mathcal{P}$-bisimilarity. We address the following fundamental question: given a category of models of computation $\mathcal{M}$ and a category of observations $\mathcal{P}$, are there any conditions under which algebraic constructs viewed as functors preserve $\mathcal{P}$-bisimilarity? We define the notion of functors being $\mathcal{P}$*-factorisable*, show how this ensures that $\mathcal{P}$-bisimilarity is a congruence with respect to such functors. Guided by the definition of $\mathcal{P}$-factorisability we show how it is possible to parametrise proofs of functors being $\mathcal{P}$-factorisable with respect to the category of observations $\mathcal{P}$, i.e., with respect to a behavioural equivalence.

**Keywords:** *Open maps, $\mathcal{P}$-bisimilarity, $\mathcal{P}$-factorisability, congruences, process algebra, category theory.*

## 1 Introduction

Category theory has proven itself very useful in many fields of theoretical computer science. We mention just one example which is directly related to the work presented in the following sections. In [JNW93], Joyal, Nielsen, and Winskel have used category theory to propose an abstract way of capturing the notion of bisimulation, the so-called *spans of open maps*: first, a category of models of computations $\mathcal{M}$ is chosen, then a subcategory of observations $\mathcal{P}$ is chosen relative to which open maps are defined. Two models are $\mathcal{P}$-bisimilar if there exists a span of open maps between them. In [CN95, NC95] the present authors give examples of application of the theory.

* This work has been supported by The Danish Research Foundation, The Danish Research Academy, and **BRICS**. A short version of this paper will appear in the proceedings of CAAP '96.

** **B**asic **R**esearch **i**n **C**omputer **S**cience,
Centre of the Danish National Research Foundation.

Winskel and Nielsen have presented operators of CCS-like process algebras using category-theoretic concepts such as products and co-products [WN95]. A natural question to ask is whether or not it is also possible to capture the following important aspect of process algebraic operators and bisimulation equivalences: when is $\mathcal{P}$-bisimilarity a congruence with respect to some of these operators?

Based on the view that endofunctors on $\mathcal{M}$ may be seen as abstract operators we define a natural and general notion of a functor being *$\mathcal{P}$-factorisable*. We then show that a $\mathcal{P}$-factorisable functor must preserve $\mathcal{P}$-bisimilarity. We observe an apparent similarity with the idea behind Milner's proofs that CCS operators preserve strong bisimulation.

Common to much work on behavioural equivalences being congruences is that one chooses a specific (a) process term language, (b) class of models, and (c) behavioural equivalence. One then shows that specific operators—such as "parallel composition" and "nondeterministic choice"—preserve the proposed behavioural equivalence. Well-known examples are [Hen88, Mil89]. The behaviour of their process algebras is given by a structural operational semantics (SOS) [Plo81], in which the behaviour of a composite process term is given by the behaviour of its components.

In general, the term languages resemble each other, usually CCS-like, and hence the results differ from each other primarily with respect to the proposed equivalences. Based on this observation, one might look for general results.

One approach could be not to look at specific operators, but try to reason about a general set of operators. In [BIM88], Bloom, Istrail, and Meyer study a meta-theory for process algebras which are defined by SOS rule systems. They identify a rule format which ensures that any process language in so-called GSOS format has strong bisimulation as a congruence. It is worth noticing that they fix the notion of behavioural equivalence, strong bisimulation, and obtain general results by allowing the operators in the language to vary.

Based on the notion of $\mathcal{P}$-factorisability, we choose an approach "orthogonal" to that of [BIM88]. The presentation of $\mathcal{P}$-factorisability focusses, especially, on certain closure properties of the category $\mathcal{P}$. Based on this observation, we show how one can parametrise the proofs of functors being $\mathcal{P}$-factorisable with respect to the choice of the observation category $\mathcal{P}$, i.e., the choice of a behavioural equivalence. Intuitively, we fix the operators, but allow the behavioural equivalence to vary. Then we identify conditions on $\mathcal{P}$ which ensure that the varying equivalences are congruences with respect to the operators. Hence, our results can be seen as "orthogonal" to that of Bloom, Istrail, and Meyer, in that we can parametrise with respect to the behavioural equivalences, as opposed to operators, [BIM88].

In the next section we recall Joyal, Nielsen, and Winskel's theory of open maps. In Sec. 3 we present our notion of $\mathcal{P}$-factorisability. Then, in Sec. 4 we apply our theory to a variant of Winskel and Nielsen's labelled transition systems [WN95]. We consider the universal constructions from [WN95] and provide general "congruence" results parametrised by the category of observations $\mathcal{P}$.

We then continue by examining the trickier recursion operator in Sec. 5. Finally we conclude and give suggestions for further research in Sec. 6.

## 2 Open Maps

In this section we briefly recall the basic definitions from [JNW93]. We present a slightly more general definition since it turns out more beneficial, more specifically for Theorem 30 and the discussion in Sect. 4.8.

Let $\mathcal{U}$ denote a category, the *universe*. A morphism $m : X \longrightarrow Y$ in $\mathcal{U}$ should intuitively be thought of as a simulation of $X$ in $Y$. Then, a subcategory of $\mathcal{U}$ which represents a *model of computation* has to be identified. We denote this category $\mathcal{M}$. Also, within $\mathcal{U}$, we choose a subcategory of "observation objects" and "observation extension" morphisms between these objects. We denote this *category of observations* by $\mathcal{P}$. If nothing else is mentioned, we assume that $\mathcal{U} = \mathcal{M}$, corresponding to the definitions in [JNW93].

Given an observation (object) $O$ in $\mathcal{P}$ and a model $X$ in $\mathcal{M}$, then $O$ is said to be an *observable behaviour* of $X$ if there exists a morphism $p : O \longrightarrow X$ in $\mathcal{M}$. We think of $p$ as representing a "run" of $O$ in $X$. We shall use $O, O', \ldots$ to denote observations and $T, T', X, Y, \ldots$ to denote objects from $\mathcal{M}$. A morphism $O \xrightarrow{q} O'$ is implicitly assumed to belong to $\mathcal{P}$.

Next, we identify morphisms $m : X \longrightarrow Y$ in $\mathcal{M}$ which have the property that whenever an observable behaviour of $X$ can be extended via $f$ in $Y$ then that extension can be matched by an extension of the observable behaviour in $X$.

**Definition 1. Open Maps**
A morphism $m : X \longrightarrow Y$ in $\mathcal{M}$ is said to be $\mathcal{P}$-*open* (or just an *open map*) if whenever $f : O_1 \longrightarrow O_2$ in $\mathcal{P}$, $p : O_1 \longrightarrow X$, $q : O_2 \longrightarrow Y$ in $\mathcal{M}$, and the diagram

$$\begin{tikzcd} O_1 \arrow[r, "p"] \arrow[d, "f"'] & X \arrow[d, "m"] \\ O_2 \arrow[r, "q"'] & Y \end{tikzcd} \tag{1}$$

commutes, i.e., $m \circ p = q \circ f$, there exists a morphism $h : O_2 \longrightarrow X$ in $\mathcal{M}$ (a *mediating* morphism) such that the two triangles in the diagram

$$\begin{tikzcd} O_1 \arrow[r, "p"] \arrow[d, "f"'] & X \arrow[d, "m"] \\ O_2 \arrow[r, "q"'] \arrow[ur, "h"] & Y \end{tikzcd} \tag{2}$$

commute, i.e., $p = h \circ f$ and $q = m \circ h$. When no confusion is possible, we refer to $\mathcal{P}$-open morphisms as *open maps*. $\square$

The abstract definition of bisimilarity is as follows.

**Definition 2. $\mathcal{P}$-bisimilarity**
Two models $X$ and $Y$ in $\mathcal{M}$ are said to be $\mathcal{P}$*-bisimilar* (in $\mathcal{M}$), written $X \sim_{\mathcal{P}} Y$, if there exists a *span of open maps* from a common object $Z$:

$$\begin{tikzcd} & Z \arrow[dl, "m"'] \arrow[dr, "m'"] & \\ X & & Y \end{tikzcd} \tag{3}$$

$\square$

*Remark.* Notice that if $\mathcal{M}$ has pullbacks, it can be shown that $\sim_{\mathcal{P}}$ is an equivalence relation. The important observation is that pullbacks of open maps are themselves open maps. For more details, the reader is referred to [JNW93].

As a preliminary example of a category of models of computation $\mathcal{M}$ we present *labelled transition systems*.

**Definition 3.** A *labelled transition system* over *Act* is a tuple

$$(S, i, Act, \longrightarrow) \ , \tag{4}$$

where $S$ is a set of states with *initial state* $i$, *Act* is a set of actions ranged over by $\alpha$, $\beta$, ... , and $\longrightarrow \subseteq S \times Act \times S$ is the transition relation. For the sake of readability we introduce the following notation. Whenever $(s_0, \alpha_1, s_1)$, $(s_1, \alpha_2, s_2)$, ..., $(s_{n-1}, \alpha_n, s_n) \in \longrightarrow$ we denote this as $s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_n} s_n$ or $s_0 \xrightarrow{v} s_n$, where $v = \alpha_1\alpha_2 \cdots \alpha_n \in Act^*$. Also, we assume that all states $s \in S$ are reachable from $i$, i.e., there exists a $v \in Act^*$ such that $i \xrightarrow{v} s$. $\square$

Let us briefly remind the reader of Park and Milner's definition of strong bisimulation [Mil89].

**Definition 4.** Let $T_1 = (S_1, i_1, Act, \longrightarrow_1)$ and $T_2 = (S_2, i_2, Act, \longrightarrow_2)$. A *strong bisimulation between* $T_1$ *and* $T_2$ is a relation $R \subseteq S_1 \times S_2$ such that

$$(i_1, i_2) \in R \ , \tag{5}$$

$$((r, s) \in R \ \wedge \ r \xrightarrow{\alpha}_1 r') \ \Rightarrow \ \text{for some } s', (s \xrightarrow{\alpha}_2 s' \ \wedge \ (r', s') \in R) \ , \tag{6}$$

$$((r, s) \in R \ \wedge \ s \xrightarrow{\alpha}_2 s') \ \Rightarrow \ \text{for some } r', (r \xrightarrow{\alpha}_1 r' \ \wedge \ (r', s') \in R) \ . \tag{7}$$

$T_1$ and $T_2$ are said to be *strongly bisimilar* if there exists a strong bisimulation between them. $\square$

Henceforth, whenever no confusion is possible we drop the indexing subscripts on the transition relations and write $\longrightarrow$, instead.

By defining morphisms between labelled transition systems we can obtain a category of models of computation, $\mathcal{TS}_{Act}$, labelled transition systems.

**Definition 5.** Let $T_1 = (S_1, i_1, Act, \longrightarrow_1)$ and $T_2 = (S_2, i_2, Act, \longrightarrow_2)$. A morphism $m : T_1 \longrightarrow T_2$ is a function $m : S_1 \longrightarrow S_2$ such that

$$m(i_1) = i_2 \ , \tag{8}$$

$$s \xrightarrow{\alpha}_1 s' \ \Rightarrow \ m(s) \xrightarrow{\alpha}_2 m(s') \ . \tag{9}$$

□

The intuition behind this specific choice of morphism is that an $\alpha$-labelled transition in $T_1$ must be simulated by an $\alpha$-labelled transition in $T_2$. Composition of morphisms is defined as the usual composition of functions.

By varying the choice of $\mathcal{P}$ we can obtain different behavioural equivalences, corresponding to $\mathcal{P}$-bisimilarity. E.g., if, as done in [JNW93], we choose $\mathcal{P}_M$ as the full subcategory of $\mathcal{TS}_{Act}$ whose objects are finite synchronisation trees with at most one maximal branch, i.e., labelled transition systems of the form

$$i \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_n} s_n \ , \tag{10}$$

where all states are distinct, we get:

**Theorem 6.** [JNW93] *$\mathcal{P}_M$-bisimilarity coincides with Park and Milner's strong bisimulation.*

By slightly restricting our choice of observation extension so that $\mathcal{P}_H$ is the subcategory of $\mathcal{TS}_{Act}$ whose objects (observations) are of the form (10), and whose morphisms are the identity morphisms and morphisms whose domains are observations having only one state (the empty word), we get:

**Theorem 7.** [NC95] *$\mathcal{P}_H$-bisimilarity coincides with Hoare trace equivalence.*

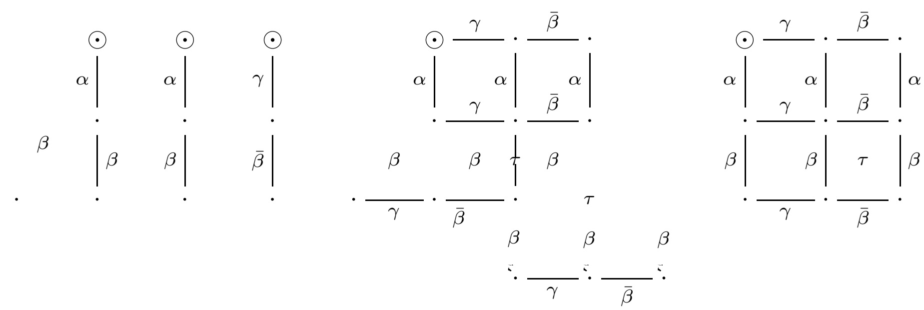
In [NC95] other behavioural equivalences were considered, e.g., weak bisimulation and probabilistic bisimulation.

## 3 $\mathcal{P}$-Factorisability

In this section we propose the notion of *$\mathcal{P}$-factorisability*. We start by a motivating example and continue with some category theoretical preliminaries, which notationally eases the presentation of $\mathcal{P}$-factorisability.

### 3.1 An Example

Consider $\mathcal{M} = \mathcal{TS}_{Act}$ and $\mathcal{P} = \mathcal{P}_M$ from Sec. 2 and the transition systems below, which we denote—left to right—$T_1, \ldots, T_5$. The initial states are depicted as $\odot$.




$T_1$ is strongly bisimilar ($\mathcal{P}$-bisimilar) to $T_2$. In fact, there is an obvious open map $k$ from $T_1$ to $T_2$. Considering $T_3$ to be fixed, we can define a functor $\_\|T_3 : \mathcal{M} \longrightarrow \mathcal{M}$, where $\|$ acts as a CCS-like parallel composition. $T_4 = T_1\|T_3$ and $T_5 = T_2\|T_3$ serve as an informal illustration of $\_\|T_3$, when applied to $T_1$ and $T_2$, respectively. In much the same way as Milner [Mil89] shows that $P \sim P'$ implies $P \,\|\, Q \sim P' \,\|\, Q$, we would like to conclude that if $k : T_1 \longrightarrow T_2$ is open, then so is $T_1\|T_3 \xrightarrow{k\|T_3} T_2\|T_3$. [3]

Recall that $\mathcal{P}$-bisimilarity is based on open maps, which again are based on observations from $\mathcal{P}$. E.g., we can observe $O$, the behaviour $\odot \xrightarrow{\alpha} \cdot \xrightarrow{\gamma} \cdot$, in $T_4$ and—via $k\|T_3 : T_4 \longrightarrow T_5$—in $T_5$. Some of these transitions in $T_4$, here only the $\alpha$ transition, are due to transitions "from" $T_1$. Using $k$, we conclude that the $\alpha$ transition in $O$ must also be observable in $T_2$. In fact, we have a commuting diagram as in (1) with $X = T_4$, $Y = T_5$, $O_1 = O_2 = O$, $m = k\|T_3$, and $f = 1_O$, and by the above we have extracted a second commuting diagram of the form (1) with $X = T_1$, $Y = T_2$, $O_1 = O_2 = O' = \odot \xrightarrow{\alpha} \cdot$, and $m = k$.

The way we have "factored" $O$ into $O'$ is consistent with $\_\|T_3$ in the following sense: there exists a commuting diagram of the form

[3] In fact, just as Milner uses a bisimulation $P \sim P'$ to exhibit a bisimulation $P \,\|\, Q \sim P' \,\|\, Q$, we will "factor" the observation $\odot \xrightarrow{\alpha} \cdot \xrightarrow{\gamma} \cdot$ into transitions from $T_3$ and from $T_1$ and $T_2$, respectively. This will guide us to the mediating morphism required in (2).

$$\begin{tikzcd}
O \arrow[rr, dash] \arrow[dd, dash] & & O'\|T_3 \arrow[dd, dash] \\
& T_1\|T_3 \arrow[dd, dash] & \\
O \arrow[rr, dash] & & O'\|T_3 \\
& T_2\|T_3 &
\end{tikzcd}$$

In the next section, we formalise this by defining the notion of $\mathcal{P}$-factorisability, and, as a consequence, we will be able to conclude that $k\|T_3$ is an open map.

### 3.2 Categorical Preliminaries

Given a category $\mathcal{C}$ with objects $\mathcal{C}_0$ and morphisms (arrows) $\mathcal{C}_1$, let $\widehat{\mathcal{C}}$ be the category whose objects are $\mathcal{C}_1$ and whose morphisms represent commuting diagrams, i.e., there is a morphism $(h_1, h_2)$ from $f$ to $g$ if

$$\begin{tikzcd}
\cdot \arrow[r, dash, "h_1"] \arrow[d, dash, "f"'] & \cdot \arrow[d, dash, "g"] \\
\cdot \arrow[r, dash, "h_2"'] & \cdot
\end{tikzcd} \tag{11}$$

is a commuting diagram in $\mathcal{C}$. Composition of morphisms is defined componentwise. For notational convenience we may "hat" objects and morphisms from $\widehat{\mathcal{C}}$, e.g., $\widehat{X}$ and $\widehat{m}$. When convenient, we will denote objects from $\widehat{\mathcal{C}}$ as morphisms from $\mathcal{C}$, e.g., $\widehat{X}$ might be denoted $f$.

Notice that a functor $F : \mathcal{C} \longrightarrow \mathcal{D}$ induces a functor $\widehat{F} : \widehat{\mathcal{C}} \longrightarrow \widehat{\mathcal{D}}$, which sends an object $\widehat{X}$ to $F(\widehat{X})$ and a morphism $\widehat{m} = (m_1, m_2)$ to $(F(m_1), F(m_2))$.

### 3.3 Factorising Observations

**Definition 8. $\mathcal{P}$-factorisability**
A functor $F : \mathcal{M} \longrightarrow \mathcal{M}$ is said to be $\mathcal{P}$-*factorisable* if whenever we have an object $\widehat{O}$ in $\widehat{\mathcal{P}}$, an object $\widehat{X}$ in $\widehat{\mathcal{M}}$, and a morphism $\widehat{O} \stackrel{\widehat{q}}{\longrightarrow} \widehat{F}(\widehat{X})$ in $\widehat{\mathcal{M}}$, then there exist an object $\widehat{O_1}$ in $\widehat{\mathcal{P}}$ and morphisms $\widehat{O} \stackrel{\widehat{q^\star}}{\longrightarrow} \widehat{F}(\widehat{O_1})$ and $\widehat{O_1} \stackrel{\widehat{q^\#}}{\longrightarrow} \widehat{X}$ in $\widehat{\mathcal{M}}$ such that the diagram

$$\begin{tikzcd}
\widehat{O} \arrow[r, dash, "\widehat{q^\star}"] \arrow[dr, phantom, "\widehat{q}"'] & \widehat{F}(\widehat{O_1}) \arrow[d, dash, "\widehat{F}(\widehat{q^\#})"] \\
& \widehat{F}(\widehat{X})
\end{tikzcd} \tag{12}$$

commutes in $\widehat{\mathcal{M}}$. $\square$

**Definition 9.** A functor $F : \mathcal{M} \longrightarrow \mathcal{M}$ is a $\mathcal{P}$*-operator* if it is $\mathcal{P}$-bisimilarity preserving, i.e., if $A$ is $\mathcal{P}$-bisimilar to $B$, then $F(A)$ is $\mathcal{P}$-bisimilar to $F(B)$. $\square$

**Theorem 10.** *Any $\mathcal{P}$-factorisable functor $F : \mathcal{M} \longrightarrow \mathcal{M}$ is a $\mathcal{P}$-*operator.

*Proof.* It is sufficient to show that $F$ preserves open maps. Assume $m : X \longrightarrow X'$ is an open map and we are given a commuting diagram

$$\begin{tikzcd} O \arrow[r, "q"] \arrow[d, "f"'] & F(X) \arrow[d, "F(m)"] \\ O' \arrow[r, "q'"'] & F(X') \end{tikzcd}$$

with $q$ and $q'$ in $\mathcal{M}$. This diagram is a morphism $\widehat{O} \xrightarrow{\widehat{q}} \widehat{F}(\widehat{X})$ in $\widehat{\mathcal{M}}$. By $\mathcal{P}$-factorisability there exist $\widehat{O_1}$ in $\widehat{\mathcal{P}}$ and morphisms $\widehat{O} \xrightarrow{\widehat{q^\star}} \widehat{F}(\widehat{O_1})$ and $\widehat{O_1} \xrightarrow{\widehat{q^\#}} \widehat{X}$ in $\widehat{\mathcal{M}}$ such that (12) commutes. Denote $\widehat{O}$ as $f : O \longrightarrow O'$, $\widehat{q}$ as $(q, q')$, $\widehat{O_1}$ as $m_1 : O_1 \longrightarrow O_1'$, $\widehat{q^\star}$ as $(q^\star, q'^\star)$, $\widehat{X}$ as $m : X \longrightarrow X'$, and $\widehat{q^\#}$ as $(q^\#, q'^\#)$. Since $\widehat{O_1} \xrightarrow{\widehat{q^\#}} \widehat{X}$ represents a commuting diagram and $m$ was open, there exists a morphism $p : O_1' \longrightarrow X$ such that the diagram

$$\begin{tikzcd} O \arrow[r, "q^\star"] \arrow[d, "f"'] & F(O_1) \arrow[r, "F(q^\#)"] \arrow[d, "F(m_1)"'] & F(X) \arrow[d, "F(m)"] \\ O' \arrow[r, "q'^\star"'] & F(O_1') \arrow[r, "F(q'^\#)"'] \arrow[ur, "F(p)"] & F(X') \end{tikzcd}$$

must commute (by (12)). But then

$$\begin{aligned} q &= F(q^\#) \circ q^\star \ , \text{ by (12)} \\ &= F(p) \circ F(m_1) \circ q_\star \\ &= (F(p) \circ q'^\star) \circ f \ , \end{aligned}$$

and

$$\begin{aligned} q' &= F(q'^\#) \circ q'^\star \ , \text{ by (12)} \\ &= F(m) \circ (F(p) \circ q'^\star) \ . \end{aligned}$$

We conclude that $F(m)$ is open. Hence if $X \xleftarrow{m} Z \xrightarrow{n} Y$ is a span of open maps, $F(X) \xleftarrow{F(m)} F(Z) \xrightarrow{F(n)} F(Y)$ is a span of open maps. $\blacksquare$

## 4 Application, an Example

As an example of the application of the theory we consider the category $\mathcal{TS}$ of labelled transition systems [4] from [WN95]. As it is shown there, process-language constructs can be interpreted as universal constructions in $\mathcal{TS}$. In the following subsections, we show how our theory can be applied to the functors associated to these universal constructions.

### 4.1 The Category of Labelled Transition Systems

In this section we define the category $\mathcal{TS}$ inspired by [WN95].

**Definition 11.** The category $\mathcal{TS}$ has as objects $(S, i, L, \longrightarrow)$, labelled transition systems (lts) with labelling set $L$. We require that all states in $S$ be reachable (from the initial state $i$). □

We shall use the abbreviation $T_j$ for $(S_j, i_j, L_j, \longrightarrow_j)$. If clear from the context we will omit the subscript $j$. Also, all the following constructions do produce ltss in $\mathcal{TS}$, i.e., all states are reachable.

For technical reasons we assume the existence of a special element $*$ which is not member of any labelling set. A partial function $\lambda$ between two labelling sets $L$ and $L'$ can then be represented as a total function from $L \cup \{*\}$ to $L' \cup \{*\}$ such that $*$ is mapped to $*$. If $a \in L$ is mapped to $*$, we interpret this as meaning that $\lambda$ is undefined on $a$. Overloading the symbol $\lambda$, we shall write this as $\lambda : L \hookrightarrow L'$. Given $T = (S, i, L, \longrightarrow)$, we define $\longrightarrow_*$ to be the set $\longrightarrow \cup \{(s, *, s) \mid s \in S\}$. The transitions $(s, *, s)$ are called *idle* transitions.

**Definition 12.** A morphism $m : T_0 \longrightarrow T_1$ is a pair $f = (\sigma_m, \lambda_m)$, where $\sigma_m : S_0 \longrightarrow S_1$ and $\lambda_m : L_0 \hookrightarrow L_1$ are total functions such that

$$\sigma_m(i_0) = i_1 \tag{13}$$

$$s \xrightarrow{a}_0 s' \Rightarrow \sigma_m(s) \xrightarrow{\lambda(a)}_{1*} \sigma_m(s') \tag{14}$$

□

The intuition is that initial states are preserved and transitions in $T_0$ are simulated in $T_1$, except when $\lambda_m(a) = *$, in which case they represent inaction in $T_1$. Composition of morphisms is defined component-wise. This defines the category $\mathcal{TS}$. We suppress the subscript $m$ when no confusion is possible.

Let $\mathbf{Set}_*$ denote the category whose objects are labelling sets $L$ and whose morphisms are partial functions $\lambda : L \hookrightarrow L'$ between labelling sets.

[4] This category is different from the one presented in Sec. 2; we use this category because it has universal constructions such as, e.g., products and co-products which correspond in an almost direct way to the well-known process algebraic constructions.

## 4.2 More Categorical Preliminaries, Fibred Category Theory

Let $p : \mathcal{TS} \longrightarrow \mathbf{Set}_*$ be the function which sends an $\mathsf{lts}$ to its labelling set and a morphism $(\sigma, \lambda) : T_0 \longrightarrow T_1$ to $\lambda : L_0 \longrightarrow L_1$. A *fibre* over $L$, $p^{-1}(L)$, is the subcategory of $\mathcal{TS}$ whose objects have labelling set $L$ and whose morphisms $f$ map to $1_L$, the identity function on $L$, under $p$.

We will use the following notions from fibred category theory.

**Definition 13.** A morphism $f : T \longrightarrow T'$ in $\mathcal{TS}$ is said to be *Cartesian* with respect to $p : \mathcal{TS} \longrightarrow \mathbf{Set}_*$ if for any morphism $g : T'' \longrightarrow T'$ in $\mathcal{TS}$ such that $p(g) = p(f)$ there is a unique morphism $h : T'' \longrightarrow T$ such that $p(h) = 1_{p(T)}$ and $f \circ h = g$.

$$\begin{tikzcd}
 & T'' \arrow[d, "h"'] \arrow[dr, "g"] & \\
\mathcal{TS} \arrow[dd, "p"'] & T \arrow[r, "f"'] & T' \\
 & & \\
\mathbf{Set}_* & p(T) \arrow[r, "p(f)"'] & p(T')
\end{tikzcd}$$

A Cartesian morphism $f : T \longrightarrow T'$ in $\mathcal{TS}$ is said to be a *Cartesian lifting* of the morphism $p(f)$ in $\mathbf{Set}_*$ with respect to $T'$. □

It can be shown now that $p$ is a *fibration*, i.e.,

- any morphism $\lambda : L \longrightarrow L'$ in $\mathbf{Set}_*$ has a Cartesian lifting with respect to any $T'$ in $TS$ such that $p(T') = L'$.
- any composition of Cartesian morphisms is itself Cartesian.

Dually, we define a morphism to be *co-Cartesian*.

**Definition 14.** A morphism $f : T \longrightarrow T'$ in $TS$ is said to be *co-Cartesian* with respect to $p : TS \longrightarrow \mathbf{Set}_*$ if for any morphism $g : T \longrightarrow T''$ in $TS$ such that $p(g) = p(f)$ there is a unique morphism $h : T' \longrightarrow T''$ such that $p(h) = 1_{p(T')}$ and $h \circ f = g$.

$$\begin{tikzcd}
 & & T'' \\
TS \arrow[dd, "p"'] & T \arrow[r, "f"'] \arrow[ur, "g"] & T' \arrow[u, "h"'] \\
 & & \\
\mathbf{Set}_* & p(T) \arrow[r, "p(f)"'] & p(T')
\end{tikzcd}$$

A co-Cartesian morphism $f : T \longrightarrow T'$ in $TS$ is said to be a *co-Cartesian lifting* of the morphism $p(f)$ in $\mathbf{Set}_*$ with respect to $T'$. □

Similarly, it can be shown that $p$ is a *co-fibration*, i.e., $p^{op} : TS^{op} \longrightarrow \mathbf{Set}_*^{op}$ is a fibration.

In the following, let $\mathcal{U}$ be $\mathcal{TS}$, let $\mathcal{F}$ be the union of all fibres over all labelling sets, and let $\mathcal{M}$ be the subcategory of $\mathcal{F}$ induced by all non-restarting $\mathsf{lts}$s, i.e., there are no transitions into the initial state. The reason for staying within fibres is that one commonly insists on having labelled actions simulated by identically labelled actions. Notice that $\mathcal{TS}_{Act}$ from Sect. 2 can be viewed as the fibre $p^{-1}(Act)$. Morphisms in $\mathcal{M}$ will always be of the form $(\sigma, 1_L)$, for some labelling set $L$. In particular, all commuting diagrams of the form (1) in $\mathcal{M}$ will always belong to some fibre $p^{-1}(L)$. It can also be shown that $\mathcal{M}$ has pullbacks, hence $\sim_{\mathcal{P}}$ is an equivalence relation [JNW93]. The reason we consider non-restarting $\mathsf{lts}$s is technical. We will address this issue below.

We shall assume that the category $\mathcal{P}$ of observation is closed under renaming of states and closed under variation of labelling sets, i.e., if $(S, i, L, \longrightarrow)$ is an observation and $L'$ is any labelling set such that $(S, i, L', \longrightarrow)$ is an $\mathsf{lts}$, then it is also an observation.

To emphasise the use of the theory in Sect. 3, we will use the notation $\mathcal{M}$ and $\mathcal{P}$.

### 4.3 Product

In this section, we consider the product construction, which has strong relations to, e.g., CCS's parallel composition operator, see [WN95] and Sect. 4.8. In [WN95], it is shown how CCS's parallel composition operator can be expressed using the product, renaming, and relabelling operators we present below.

**Definition 15.** Let $T_0 \times T_1$ denote $(S, i, L, \longrightarrow)$, where

- $S = S_0 \times S_1$, with $i = (i_0, i_1)$ and projections $\rho_0 : S \longrightarrow S_0$, $\rho_1 : S \longrightarrow S_1$ ,
- $L = L_0 \times_* L_1 = (L_0 \times \{*\}) \cup (\{*\} \times L_1) \cup (L_0 \times L_1)$, with projections $\pi_0 : L_0 \times_* L_1 \hookrightarrow L_0$ and $\pi_1 : L_0 \times_* L_1 \hookrightarrow L_1$, and
- $s \xrightarrow{a}_* s' \;\Leftrightarrow\; \rho_0(s) \xrightarrow{\pi_0(a)}_{0*} \rho_0(s') \;\wedge\; \rho_1(s) \xrightarrow{\pi_1(a)}_{1*} \rho_1(s')$ .

□

Let $\Pi_0 = (\rho_0, \pi_0) : T_0 \times T_1 \longrightarrow T_0$ and $\Pi_1 = (\rho_1, \pi_1) : T_0 \times T_1 \longrightarrow T_1$. It can be shown that this construction is a product of $T_0$ and $T_1$ in the category $\mathcal{TS}$.

The product construction allows the two components $T_0$ and $T_1$ to proceed independently of each as well as synchronising on any of their actions. This behaviour is far too generous compared to CCS's parallel composition. However, by restricting away all action pairs from $T_0 \times T_1$ that are not of the form $(a, *)$, $(*, a)$, or $(a, \bar{a})$, corresponding to a move in the left component, right component, and a synchronisation on complementary actions, and relabelling $(a, *)$, $(*, a)$, and $(a, \bar{a})$ to $a$,$a$, and $\tau$, respectively, we obtain CCS's parallel composition. Both restriction and relabelling can be handled in our setting.

For a fixed $\mathsf{lts}$ $T_0$ the above construction induces an obvious functor $T_0 \times \_ : \mathcal{M} \longrightarrow \mathcal{M}$. We continue by applying our theory to prove a general result for this functor. First we need a definition, which will help formalising the "factoring" of observations in a product object.

**Definition 16.** Let $T = (S, i, L, \longrightarrow)$ and let $\lambda : L \hookrightarrow L'$ represent a partial function between labelling sets. Let $\equiv$ be the least equivalence relation on $S$ such that if $s \xrightarrow{a} s'$ and $\lambda(a) = *$, then $s \equiv s'$. Let $[s]$ denote the equivalence class of $s$ under $\equiv$. Define $[T]_\lambda = (S', i', L', \longrightarrow')$, where

- $S' = \{[s] \mid s \in S\}$ and $i' = [i]$,
- $[s] \xrightarrow{b}{}' [s'] \;\Leftrightarrow\; \exists v \in [s], v' \in [s'], a \in L.\, v \xrightarrow{a} v' \;\wedge\; \lambda(a) = b \neq *$ .

Let $\eta_{(T,\lambda)} : T \longrightarrow [T]_\lambda$ be the pair $(\sigma, \lambda)$, where $\sigma(s) = [s]$. $\square$

A simple argument shows that $\sigma$ is well-defined. If $s \equiv s'$, then there exists a "back and forth" path

$$\begin{array}{cccccccc} & \cdot & & \cdot & & \cdot & & \cdot \\ & & & & & & l_{n-1} & \\ l_1 & & l_2 & & l_3 \quad \cdots & l_{n-2} & & l_n \\ s & & \cdot & & & \cdot & & s' \end{array}$$

where $l_i = *$ or $\lambda(l_i) = *$, for $1 \leq i \leq n$. We conclude that $\sigma(s) = \sigma(s')$.

**Proposition 17.** *The morphism $\eta_{(T,\lambda)} : T \longrightarrow [T]_\lambda$ is co-Cartesian with respect to $p$.*

*Proof.* Assume $f : T \longrightarrow T_1$ and $p(f) = p(\eta_{(T,\lambda)})$. Define $(\sigma', 1_{L'}) : [T]_\lambda \longrightarrow T_1$ by $\sigma'([s]) = \sigma_f(s)$. By an argument similar to the above one can show that $\sigma'$ is well-defined. To see that $(\sigma', 1_{L'})$ is a morphism first notice that $\sigma'([i]) = \sigma_f(i) = i_1$. Next, assume $[s] \xrightarrow{b}{}' [s']$, i.e., $\exists v \in [s], v' \in [s'], a \in L.\, v \xrightarrow{a} v' \;\wedge\; \lambda(a) = b \neq *$. Then $\sigma_f(v) \xrightarrow{\lambda(a)}_{1*} \sigma_f(v')$, i.e., $\sigma'([s]) \xrightarrow{b}_{1*} \sigma'([s'])$. It is easy to see that $(\sigma', 1_{L'})$ is the uniquely determined morphism such that $p((\sigma', 1_{L'})) = 1_{p([T]_\lambda)}$ and $f = (\sigma', 1_{L'}) \circ \eta_{(T,\lambda)}$. $\blacksquare$

**Lemma 18.** *For a partial function $\lambda : L \hookrightarrow L'$ between labelling sets, there is a functor $F_\lambda : p^{-1}(L) \longrightarrow p^{-1}(L')$ which sends $f = (\sigma, 1_L) : T_0 \longrightarrow T_1$ to $F_\lambda(f) = (\gamma, 1_{L'}) : [T_0]_\lambda \longrightarrow [T_1]_\lambda$ defined by $\gamma([s]) = [\sigma(s)]$.*

*Proof.* The proof is routine, hence omitted. $\blacksquare$

We can now show the following theorem.

**Theorem 19.** *Let $T_0$ belong to $\mathcal{M}$ and $L_0 = p(T_0)$. Let $\mathcal{P}$ be any subcategory of $\mathcal{U}$ such that whenever we have $O \xrightarrow{f} O'$ in $\mathcal{P}$, where $p(f) = 1_{L_0 \times_* L}$ for some $L$, then $F_{\pi_1}(O) \xrightarrow{F_{\pi_1}(f)} F_{\pi_1}(O')$ also belongs to $\mathcal{P}$. Then $T_0 \times \_ : \mathcal{M} \longrightarrow \mathcal{M}$ is a $\mathcal{P}$-operator.*

*Proof.* By Theorem 10 it is sufficient to show that $T_0 \times \_$ is $\mathcal{P}$-factorisable. So assume $T \xrightarrow{m} T'$ belongs to $\mathcal{M}$, $p(T) = L$, and we are given $\widehat{O} \xrightarrow{\widehat{q}} \widehat{T_0 \times}(\widehat{T})$, i.e., a commuting diagram in $\mathcal{M}$

$$\begin{tikzcd}
O \arrow[r, dash, "q"] \arrow[d, dash, "f"'] & T_0 \times T \arrow[d, dash, "T_0 \times m"] \\
O' \arrow[r, dash, "q'"'] & T_0 \times T'
\end{tikzcd}$$

Since $\mathcal{M}$ is the union of fibres we have $p(f) = p(q) = p(q') = p(T_0 \times m) = 1_{L_0 \times_* L}$ for some set $L$. Let $\pi_1 : L_0 \times_* L \hookrightarrow L$ be the projection on the second component. By our assumptions $F_{\pi_1}(O) \xrightarrow{F_{\pi_1}(f)} F_{\pi_1}(O')$ is in $\mathcal{P}$. Let $O_1 = F_{\pi_1}(O)$, $O'_1 = F_{\pi_1}(O')$, $q = (\sigma_q, 1_{L_0 \times_* L})$, and $q' = (\sigma_{q'}, 1_{L_0 \times_* L})$. Define

$q^{\#} = (\sigma, 1_L) : O_1 \longrightarrow T$ , where $\sigma([s]) = \rho_1(\sigma_q(s))$, and

$q'^{\#} = (\sigma', 1_L) : O'_1 \longrightarrow T'$ , where $\sigma'([s']) = \rho'_1(\sigma_{q'}(s'))$

$\rho_1$ and $\rho'_1$ are the projections mentioned in Definition 15. Notice, e.g., that for any $s_1, s_2 \in [s]$ in $O_1$ we have $\rho_1(\sigma_q(s_1)) = \rho_1(\sigma_q(s_2))$. Next, define

$q^{\star} = (\gamma, 1_{L_0 \times_* L}) : O \longrightarrow T_0 \times O_1$ , where $\gamma(s) = (\rho_0(\sigma_q(s)), [s])$, and

$q'^{\star} = (\gamma', 1_{L_0 \times_* L}) : O' \longrightarrow T_0 \times O'_1$ , where $\gamma'(s') = (\rho'_0(\sigma_{q'}(s')), [s'])$

It can now be shown that both diagrams

$$\begin{tikzcd}
O \arrow[r, dash, "q^{\star}"] \arrow[d, dash, "f"'] & T_0 \times O_1 \arrow[d, dash, "1_{T_0} \times F_{\pi_1}(f)"] \\
O' \arrow[r, dash, "q'^{\star}"'] & T_0 \times O'_1
\end{tikzcd}
\qquad
\begin{tikzcd}
O_1 \arrow[r, dash, "q^{\#}"] \arrow[d, dash, "F_{\pi_1}(f)"'] & T \arrow[d, dash, "m"] \\
O'_1 \arrow[r, dash, "q'^{\#}"'] & T'
\end{tikzcd}$$

exist in $\mathcal{M}$ and commute, i.e., we have morphisms $\widehat{O} \xrightarrow{\widehat{q^{\star}}} \widehat{T_0 \times}(\widehat{O_1})$ and $\widehat{O_1} \xrightarrow{\widehat{q^{\#}}} \widehat{T}$ in $\widehat{\mathcal{M}}$. It can also be shown that $q = q^{\#} \circ q^{\star}$ and $q' = q'^{\#} \circ q'^{\star}$. Hence we have a commuting diagram of the form (12). Hence $T_0 \times \_$ is $\mathcal{P}$-factorisable. ∎

### 4.4 Co-Product

In this section, we consider the co-product construction, which has strong relations to, e.g., CCS's nondeterministic choice operator, see [WN95] and Sect. 4.8.

**Definition 20.** Let $T_0 + T_1$ denote $(S, i, L, \longrightarrow)$, where

– $S = (S_0 \times \{i_1\}) \cup (\{i_0\} \times S_1)$, with $i = (i_0, i_1)$ and injections $in_0 : S_0 \longrightarrow S$, $in_1 : S_1 \longrightarrow S$ ,
– $L = L_0 \cup_* L_1 = (L_0 \times \{*\}) \cup (\{*\} \times L_1)$, with injections $j_0 : L_0 \longrightarrow L$ and $j_1 : L_1 \longrightarrow L$, and
– $s \xrightarrow{a} s' \;\Leftrightarrow\; \exists v \xrightarrow{b}_0 v'.\, (in_0(v), j_0(b), in_0(v')) = (s, a, s')$ or

$$\exists v \xrightarrow{b}_1 v'.\, (in_1(v), j_1(b), in_1(v')) = (s, a, s')$$

□

Let $I_0 = (in_0, j_0) : T_0 \longrightarrow T_0 + T_1$ and $I_1 = (in_1, j_1) : T_1 \longrightarrow T_0 + T_1$. It can be shown that this construction is a coproduct of $T_0$ and $T_1$ in the category $\mathcal{TS}$.

As opposed to the product construction, the co-product construction resembles more a process algebraic choice, "+", operator. If we consider non-restarting ltss, co-product can be shown to correspond to "+" in a formal sense [WN95].

**Definition 21.** Given $T' = (S', i', L', \longrightarrow')$ and a partial function $\lambda : L \hookrightarrow L'$. Let $T'_{\downarrow\lambda} = (S, i, L, \longrightarrow)$, where

– $S = \{s \in S' \mid \exists a_1, \ldots, a_n \in L, s_1, \ldots, s_n \in S'.$
$\qquad i' \xrightarrow{\lambda(a_1)}{}' s_1 \xrightarrow{\lambda(a_2)}{}' \cdots \xrightarrow{\lambda(a_n)}{}' s_n \wedge s_n = s\}$
– $i = i'$
– $s \xrightarrow{b} s' \;\Leftrightarrow\; s \xrightarrow{\lambda(b)}{}'_{*} s'$

□

Let $\eta_{(T',\lambda)} : T'_{\downarrow\lambda} \longrightarrow T'$ be the pair $(in, \lambda)$, where $in$ is the injection function.

**Proposition 22.** *The morphism $\eta_{(T',\lambda)} : T'_{\downarrow\lambda} \longrightarrow T'$ is Cartesian with respect to $p$.*

**Lemma 23.** *For a partial function $\lambda : L \hookrightarrow L'$ between labelling sets, there is a functor $F_{\downarrow\lambda} : p^{-1}(L') \longrightarrow p^{-1}(L)$ which sends $f = (\sigma, 1_{L'}) : T_0 \longrightarrow T_1$ to $F_{\downarrow\lambda} = (\gamma, 1_L) : T_{0\downarrow\lambda} \longrightarrow T_{1\downarrow\lambda}$ defined by $\gamma(s) = \sigma(s)$.*

**Theorem 24.** *Let $T_0$ belong to $\mathcal{M}$ and $L_0 = p(T_0)$. Assume $\mathcal{P}$ is a subcategory of $\mathcal{M}$ such that whenever we have $O \xrightarrow{f} O'$ in $\mathcal{P}$ with $p(f) = 1_{L_0 \cup_* L}$ for some $L$, $F_{\downarrow\lambda}(O) \xrightarrow{F_{\downarrow\lambda}(f)} F_{\downarrow\lambda}(O')$ also belongs to $\mathcal{P}$, where $\lambda : L \longrightarrow L_0 \cup_* L$ is the injection function. Then $T_0 + \_ : \mathcal{M} \longrightarrow \mathcal{M}$ is a $\mathcal{P}$-operator.*

*Proof.* It is sufficient to show that $T_0 + \_$ is $\mathcal{P}$-factorisable. So assume $T \xrightarrow{m} T'$ belongs to $\mathcal{M}$, $p(T) = L$, and we are given $\widehat{O} \xrightarrow{\widehat{q}} \widehat{T_0 +}(\widehat{T})$, i.e., a commuting diagram in $\mathcal{M}$

$$\begin{tikzcd} O \arrow[r, dash, "q"] \arrow[d, dash, "f"'] & T_0 + T \arrow[d, dash, "1_{T_0}+m"] \\ O' \arrow[r, dash, "q'"'] & T_0 + T' \end{tikzcd}$$

Let $p(f) = 1_{L_0 \cup_* L}$. Let $\lambda : L \longrightarrow L_0 \cup_* L$ be the injection function sending $a \in L$ to $(*, a) \in L_0 \cup_* L$. By our assumptions $F_{\downarrow\lambda}(O) \stackrel{F_{\downarrow\lambda}(f)}{\longrightarrow} F_{\downarrow\lambda}(O')$ is in $\mathcal{P}$. Let $O_1 = F_{\downarrow\lambda}(O)$, $O_1' = F_{\downarrow\lambda}(O')$, $q = (\sigma_q, 1_{L_0 \cup_* L})$, and $q' = (\sigma_{q'}, 1_{L_0 \cup_* L})$. Define

$$q^{\#} = (\sigma, 1_L) : O_1 \longrightarrow T \text{ , where } \sigma(s) = t \text{, where } \sigma_q(s) = (r, t) \text{, and}$$

$$q'^{\#} = (\sigma', 1_L) : O_1' \longrightarrow T' \text{ , where } \sigma'(s') = t' \text{, where } \sigma_{q'}(s') = (r', t')$$

Next, define

$$q^\star = (\gamma, 1_{L_0 \cup_* L}) : O \longrightarrow T_0 + O_1 \text{ , where } \gamma(s) = (r, i_1) \text{ if } \sigma_q(s) = (r, i),$$
$$\gamma(s) = (i_0, t) \text{ if } \sigma_q(s) = (i_0, t) \text{, and}$$

$$q'^\star = (\gamma', 1_{L_0 \cup_* L}) : O' \longrightarrow T_0 + O_1' \text{ , where } \gamma'(s') = (r', i_1') \text{ if } \sigma_{q'}(s') = (r', i'),$$
$$\gamma'(s') = (i_0', s') \text{ if } \sigma_{q'}(t') = (i_0', t')$$

It can now be shown that both diagrams

$$\begin{tikzcd} O \arrow[r, dash, "q^\star"] \arrow[d, dash, "f"'] & T_0 + O_1 \arrow[d, dash, "1_{T_0} + F_{\downarrow\lambda}(f)"] \\ O' \arrow[r, dash, "q'^\star"'] & T_0 + O_1' \end{tikzcd} \qquad \begin{tikzcd} O_1 \arrow[r, dash, "q^\#"] \arrow[d, dash, "F_{\downarrow\lambda}(f)"'] & T \arrow[d, dash, "m"] \\ O_1' \arrow[r, dash, "q'^\#"'] & T' \end{tikzcd}$$

exist in $\mathcal{M}$ and commute, i.e., we have morphisms $\widehat{O} \stackrel{\widehat{q^\star}}{\longrightarrow} \widehat{T_0 + (\widehat{O_1})}$ and $\widehat{O_1} \stackrel{\widehat{q^\#}}{\longrightarrow} \widehat{T}$ in $\widehat{\mathcal{M}}$. It can also be shown that $q = q^{\#} \circ q^\star$ and $q' = q'^{\#} \circ q'^\star$. Hence we have a commuting diagram of the form (12). Hence $T_0 + \_$ is $\mathcal{P}$-factorisable. ∎

### 4.5 Restriction

In this section, we consider relabelling.

**Definition 25.** Given $T' = (S', i', L', \longrightarrow')$ and a labelling set $L$. Let $F\downarrow : \mathcal{M} \longrightarrow \mathcal{M}$ denote the functor which sends $T'$ to $T = (S, i, L, \longrightarrow)$, where

- $S = \{s \in S' \mid \exists a_1, \ldots, a_n \in L \cap L', s_1, \ldots, s_n \in S'.$
  $i' \stackrel{a_1}{\longrightarrow}{}' s_1 \stackrel{a_2}{\longrightarrow}{}' \cdots \stackrel{a_n}{\longrightarrow}{}' s_n \wedge s_n = s\}$
- $i = i'$
- $s \stackrel{a}{\longrightarrow} s' \;\Leftrightarrow\; s \stackrel{a}{\longrightarrow}{}' s',\; a \in L$

and which maps a morphism $m = (\sigma_m', 1_{L'}) : T_1' \longrightarrow T_2'$ to $F\downarrow(m) = (\sigma_m, 1_L) : F\downarrow(T_1') \longrightarrow F\downarrow(T_2')$, where $\sigma_m(s) = \sigma_m'(s)$. □

We have the following perhaps surprising result.

**Theorem 26.** *For any choice of $\mathcal{P}$ the functor $F\downarrow L$ is a $\mathcal{P}$-operator.*

*Proof.* We show that $F\!\downarrow\! L$ is a $\mathcal{P}$-operator. Assume $T \xrightarrow{m} T'$ and we have

$$\begin{tikzcd} O \arrow[r, "q"] \arrow[d, "f"'] & F\!\downarrow\! L(T) \arrow[d, "F\downarrow L(m)"] \\ O' \arrow[r, "q'"'] & F\!\downarrow\! L(T') \end{tikzcd}$$

that commutes in $\mathcal{M}$. Let $p(T) = L'$. By our assumptions we must have a commuting diagram

$$\begin{tikzcd} O_1 \arrow[r, "q^{\#}"] \arrow[d, "m_1"'] & T \arrow[d, "m"] \\ O_1' \arrow[r, "q'^{\#}"'] & T' \end{tikzcd}$$

where $O = (S, i, L, \longrightarrow)$, $O' = (S', i', L, \longrightarrow)$, $f = (\sigma_f, 1_L)$, $O_1 = (S, i, L', \longrightarrow)$, $O_1' = (S', i', L', \longrightarrow')$, $m_1 = (\sigma_f, 1_{L'})$, $q = (\sigma_q, 1_L)$, $q' = (\sigma_{q'}, 1_L)$, $q^{\#} = (\sigma_q, 1_{L'})$, and $q'^{\#} = (\sigma_{q'}, 1_{L'})$. Notice $F \downarrow L(O_1) = O$, $F \downarrow L(O_1') = O'$, and $F\!\downarrow\! L(m_1) = f$. It can easily be shown that we have a diagram in $\widehat{\mathcal{M}}$ as required in (12) and that it commutes. ■

### 4.6 Relabelling

Relabelling, as presented in [WN95], is a bit tricky. We will need some auxiliary definitions and we will have to consider (relabelling) functors between fibres.

**Definition 27.** Let $T = (S, i, L, \longrightarrow)$ be an $\mathsf{lts}$ and $\lambda : L \longrightarrow L'$ be a total function between labelling sets. Define $T\{\lambda\}$ to be the $\mathsf{lts}$ $(S, i, L', \longrightarrow')$, where

$$s \xrightarrow{a}{}' s' \;\Leftrightarrow\; \exists b.\, s \xrightarrow{b} s' \;\wedge\; \lambda(b) = a\ .$$

□

**Proposition 28.** *If $\lambda : L \longrightarrow L'$ is a total function in $\mathbf{Set}_*$, then $T \xrightarrow{f} T\{\lambda\}$, where $f = (1_S, \lambda)$ is co-Cartesian with respect to $p$.*

*Proof.* The proof is routine, hence omitted. ■

Any total function $\lambda : L \longrightarrow L'$ induces a functor $F\{\lambda\} : p^{-1}(L) \longrightarrow p^{-1}(L')$. Notice that $F\{\lambda\}$ is not an endofunctor on $\mathcal{M}$. Instead, given $\lambda : L \longrightarrow L'$ we consider $\lambda' : L \cup L' \longrightarrow L \cup L'$ defined by $\lambda'(a) = \lambda(a)$ if $a \in L$ and $\lambda'(a) = a$ otherwise. Now $p^{-1}(L)$ and $p^{-1}(L')$ embed fully and faithfully in $p^{-1}(L \cup L')$. We will therefore only consider total relabelling functions of the form $\lambda : L \longrightarrow L$.

Let $p_0 : TS \longrightarrow \mathbf{Set}$ be the functor which sends $T$ to $S$ and $(\sigma, \lambda) : T \longrightarrow T'$ to $\sigma$.

**Definition 29.** Let $F^{-1}\{\lambda\}(T)$ denote the subcategory of $p^{-1}(L)$ whose objects are $\mathsf{ltss}$ $T'$ such that $F\{\lambda\}(T') = T$ and whose morphisms $f$ map to $1_{p_0(T)}$ under $p_0$; objects in $F^{-1}\{\lambda\}(T)$ have the same set of states as $T$.

An object $T'$ in $F^{-1}\{\lambda\}(T)$ is *minimal* if the only morphisms in $F^{-1}\{\lambda\}(T)$ with codomain $T'$ is the identity morphism on $T'$. □

*Remark.* Notice that if $T'$ is minimal in $F^{-1}\{\lambda\}(T)$, then for any two transitions $s \xrightarrow{a} s'$ and $s \xrightarrow{b} s'$ in $T'$ we have $a \neq b$ implies $\lambda(a) \neq \lambda(b)$.

**Theorem 30.** *Given a total relabelling function $\lambda : L \longrightarrow L$. Choose $\mathcal{M} = p^{-1}(L)$. Let $\mathcal{P}$ be a subcategory of $\mathcal{U}$. Assume that for all $O \xrightarrow{f} O'$ in $\mathcal{P}$, where $f = (\sigma_f, 1_L)$ and $F^{-1}\{\lambda\}(O)$ and $F^{-1}\{\lambda\}(O')$ are nonempty, $(\sigma_f, 1_L) : O_1 \longrightarrow O'_1$ belongs to $\mathcal{P}$, whenever $O_1$ and $O'_1$ are minimal elements in $F^{-1}\{\lambda\}(O)$ and $F^{-1}\{\lambda\}(O')$, respectively, and $(\sigma_f, 1_L) : O_1 \longrightarrow O'_1$ defines a morphism. Then $F\{\lambda\} : \mathcal{M} \longrightarrow \mathcal{M}$ is a $\mathcal{P}$-operator.*

*Proof.* Choose $\mathcal{M} = p^{-1}(L)$. We show that $F\{\lambda\} : \mathcal{M} \longrightarrow \mathcal{M}$ is a $\mathcal{P}$-operator, where $\lambda : L \longrightarrow L$ is a total relabelling function. Assume $T \xrightarrow{m} T'$ belongs to $\mathcal{M}$ and we have

$$\begin{tikzcd} O \arrow[r, dash, "q"] \arrow[d, dash, "f"'] & F\{\lambda\}(T) \arrow[d, dash, "F\{\lambda\}(m)"] \\ O' \arrow[r, dash, "q'"'] & F\{\lambda\}(T') \end{tikzcd}$$

that commutes in $\mathcal{M}$. Since $O$ is simulated in $F\{\lambda\}(T)$ we know that $F^{-1}\{\lambda\}(O)$ is nonempty. Similarly, $F^{-1}\{\lambda\}(O')$ is nonempty. Since $O$ is simulated in $F\{\lambda\}(T)$ and $p(m) = 1_L$, there must exist a minimal $O_1$ in $F^{-1}\{\lambda\}(O)$ and a minimal $O'_1$ in $F^{-1}\{\lambda\}(O')$ such that $g = (\sigma_f, 1_L) : O_1 \longrightarrow O'_1$ is a well-defined morphism in $\mathcal{P}$ and such that

$q^{\#} = (\sigma_q, 1_L) : O_1 \longrightarrow T$ , where $q = (\sigma_q, 1_L) : O \longrightarrow F\{\lambda\}(T)$, and

$q'^{\#} = (\sigma_{q'}, 1_L) : O'_1 \longrightarrow T'$ , where $q' = (\sigma_{q'}, 1_L) : O' \longrightarrow F\{\lambda\}(T')$

are well-defined morphisms in $\mathcal{M}$.

Next, define

$q^{\star} = (\gamma, 1_L) : O \longrightarrow F\{\lambda\}(O_1)$ , where $\gamma(s) = s$, and

$q'^{\star} = (\gamma', 1_L) : O' \longrightarrow F\{\lambda\}(O'_1)$ , where $\gamma'(s') = s'$

It can now be shown that both diagrams

$$\begin{tikzcd} O \arrow[r, dash, "q^{\star}"] \arrow[d, dash, "f"'] & F\{\lambda\}(O_1) \arrow[d, dash, "F\{\lambda\}(g)"] \\ O' \arrow[r, dash, "q'^{\star}"'] & F\{\lambda\}(O'_1) \end{tikzcd} \qquad \begin{tikzcd} O_1 \arrow[r, dash, "q^{\#}"] \arrow[d, dash, "g"'] & T \arrow[d, dash, "m"] \\ O'_1 \arrow[r, dash, "q'^{\#}"'] & T' \end{tikzcd}$$

exist in $\mathcal{M}$ and commute, i.e., we have morphisms $\widehat{O} \xrightarrow{\widehat{q^\star}} \widehat{F\{\lambda\}}(\widehat{O_1})$ and $\widehat{O_1} \xrightarrow{\widehat{q^\#}} \widehat{T}$ in $\widehat{\mathcal{M}}$. It can also be shown that $q = q^\# \circ q^\star$ and $q' = q'^\# \circ q'^\star$. Hence we have a commuting diagram of the form (12). Hence $F\{\lambda\}$ is $\mathcal{P}$-factorisable. ■

Notice that $\mathcal{M} = p^{-1}(L)$ is no restriction in our case, since $\mathcal{M}$ "consists" of full subcategories of fibres: it is easy to see that a $\mathcal{P}$-open morphism in $p^{-1}(L)$ is also $\mathcal{P}$-open in $\mathcal{M}$.

### 4.7 Prefix

**Definition 31.** Given $T = (S, i, L, \longrightarrow)$ and a label $\alpha$. Let $\alpha.T = (S', i', L \cup \{\alpha\}, \longrightarrow')$, where

- $S' = \{\{s\} \mid s \in S\} \cup \{\emptyset\}$, $i' = \emptyset$, and
- $v \xrightarrow{b}{}' v' \;\Leftrightarrow\; (v = \emptyset \wedge b = \alpha \wedge v' = \{i\})$ or $(v = \{s\} \wedge v' = \{s'\} \wedge s \xrightarrow{b} s')$ .

□

Any label $\alpha$ induces a functor $\alpha.\_ : \mathcal{M} \longrightarrow \mathcal{M}$ which sends $f = (\sigma, 1_L) : T \longrightarrow T'$ to $(\sigma', 1_{L\cup\{\alpha\}}) : \alpha.T \longrightarrow \alpha.T'$, where $\sigma'(\emptyset) = \emptyset$ and $\sigma'(\{s\}) = \{\sigma(s)\}$.

**Definition 32.** Given $T$ and a label $\alpha$. Let $\alpha^{-1}(T) = (S', i', L, \rightarrow')$, where

- $S'' = \{s \in S \mid \exists v \in L^*.\, i \xrightarrow{\alpha}\xrightarrow{v} s\} \backslash \{s \mid i \xrightarrow{\alpha} s\}$ ,
- $S' = \{\{s\} \mid s \in S''\} \cup \{\{s \mid i \xrightarrow{\alpha} s\}\}$ ,
- $i = \{s \mid i \xrightarrow{\alpha} s\}$ , and
- $r \xrightarrow{a}{}' r' \;\Leftrightarrow\; \exists s \in r, s' \in r'.\, s \xrightarrow{a} s'$ .

□

Any label $\alpha$ induces a functor $\alpha^{-1} : \mathcal{U} \longrightarrow \mathcal{U}$ which sends $f = (\sigma, 1_L) : T \longrightarrow T'$ to $\alpha^{-1}(f) = (\sigma', 1_L) : T_1 \longrightarrow T_2$, where $T_1 = \alpha^{-1}(T)$, $T_2 = \alpha^{-1}(T')$, $\sigma'(i_1) = i_2$, and $\sigma'(\{s\})$ is the unique $v \in S_2$ such that $\sigma(s) \in v$. Notice that $\alpha^{-1}(T)$ may not be non-restarting even though $T$ is.

**Theorem 33.** *Let $\mathcal{P}$ be a subcategory of $\mathcal{U}$. Assume that whenever we have $O \xrightarrow{f} O'$ in $\mathcal{P}$, then $\alpha^{-1}(O) \xrightarrow{\alpha^{-1}(f)} \alpha^{-1}(O')$ also belongs to $\mathcal{P}$. Then $\alpha.\_$ is a $\mathcal{P}$-operator.*

*Proof.* We show that $\alpha.\_$ is a $\mathcal{P}$-operator. Assume $T \xrightarrow{m} T'$ and we have

$$\begin{tikzcd} O \arrow[r, dash, "q"] \arrow[d, dash, "f"'] & \alpha.T \arrow[d, dash, "\alpha.m"] \\ O' \arrow[r, dash, "q'"'] & \alpha.T' \end{tikzcd}$$

that commutes in $\mathcal{M}$. Notice that since $T$ and $T'$ are assumed to be non-restarting, $\alpha^{-1}(O)$ and $\alpha^{-1}(O')$ must also be non-restarting. Assume $\alpha \in L = p(T)$. By our assumptions $\alpha^{-1}(O) \stackrel{\alpha^{-1}(f)}{\longrightarrow} \alpha^{-1}(O')$ is in $\mathcal{P}$. Let $O_1 = \alpha^{-1}(O)$ and $O'_1 = \alpha^{-1}(O')$. Define

$$q^{\#} = (\sigma, 1_L) : O_1 \longrightarrow T \text{ , given by } \sigma(i_1) = i \text{ and } \sigma(\{s\}) = r,$$
$$\text{where } \sigma_q(s) = \{r\} \text{ and } q = (\sigma_q, 1_L) : O \longrightarrow \alpha.T, \text{ and}$$

$$q'^{\#} = (\sigma', 1_L) : O'_1 \longrightarrow T' \text{ , given by } \sigma'(i'_1) = i' \text{ and } \sigma'(\{s'\}) = r',$$
$$\text{where } \sigma_{q'}(s') = \{r'\} \text{ and } q' = (\sigma_{q'}, 1_L) : O' \longrightarrow \alpha.T'$$

Next, define

$$q^{\star} = (\gamma, 1_L) : O \longrightarrow \alpha.O_1 \text{ , where } \gamma(i) = \emptyset,$$
$$\gamma(s) = \{i_1\} \text{ for } s \in \{s \,|, i \xrightarrow{\alpha} s\} \text{ in } O,\ \gamma(s) = \{\{s\}\}, \text{ else, and}$$

$$q'^{\star} = (\gamma', 1_L) : O' \longrightarrow \alpha.O'_1 \text{ , where } \gamma'(i') = \emptyset,$$
$$\gamma'(s') = \{i'_1\} \text{ for } s' \in \{s' \,|, i' \xrightarrow{\alpha} s'\} \text{ in } O',\ \gamma'(s') = \{\{s'\}\}, \text{ else.}$$

It can now be shown that both diagrams

$$\begin{tikzcd} O \arrow[r, dash, "q^{\star}"] \arrow[d, dash, "f"'] & \alpha.O_1 \arrow[d, dash, "\alpha.g"] \\ O' \arrow[r, dash, "q'^{\star}"'] & \alpha.O'_1 \end{tikzcd} \qquad \begin{tikzcd} O_1 \arrow[r, dash, "q^{\#}"] \arrow[d, dash, "g"'] & T \arrow[d, dash, "m"] \\ O'_1 \arrow[r, dash, "q'^{\#}"'] & T' \end{tikzcd}$$

exist in $\mathcal{M}$ and commute, i.e., we have morphisms $\widehat{O} \xrightarrow{\widehat{q^{\star}}} \widehat{\alpha}.(\widehat{O_1})$ and $\widehat{O_1} \xrightarrow{\widehat{q^{\#}}} \widehat{T}$ in $\widehat{\mathcal{M}}$. It can also be shown that $q = q^{\#} \circ q^{\star}$ and $q' = q'^{\#} \circ q'^{\star}$. Hence we have a commuting diagram of the form (12).

For the case where $\alpha \notin p(T)$ the same reasoning can be used. First extend $T$ and $T'$'s labelling sets to include $\alpha$. The induced $m_\alpha : T \longrightarrow T'$ in $p^{-1}(L \cup \{\alpha\})$ will be $\mathcal{P}$-open if and only if $m : T \longrightarrow T'$ is due to our assumptions about $\mathcal{P}$. Now notice that $m_\alpha$ and $m$ are identical under $\alpha.\_$. We conclude that $\alpha.\_$ is $\mathcal{P}$-factorisable. ■

### 4.8 Putting it together

Let us consider Milner's CCS-operators except recursion, which is handled in next section. Under the common assumption that only guarded sum is considered, it is shown in [WN95] how these CCS-operators can be expressed by the above constructions (functors). For each operator we have obtained a theorem for the corresponding functor which identifies conditions which guarantee that the functor is a $\mathcal{P}$-operator. Or put differently, for each functor we have meta-theorems providing conditions on $\mathcal{P}$ guaranteeing that $\sim_{\mathcal{P}}$ remains a congruence with respect to the functor (operator).

However, we would like to consider more than one functor at the time. Does there exist choices of $\mathcal{P}$, such that $\mathcal{P}$ satisfies the conditions of all our theorems (including relabelling and prefixing) ?

Choosing $\mathcal{P}$ in $\mathcal{M}$ as the full subcategory induced by words (i.e., fibre-wise as done for $\mathcal{P}_M$ in Sec. 4.2), we can show that $\sim_{\mathcal{P}}$ also corresponds to Milner's strong bisimulation. Moreover, it is easy to see that $\mathcal{P}$ satisfies *all* conditions of our theorems, i.e., $\sim_{\mathcal{P}}$ must be a congruence with respect to all the operators (functors). For example, let us just consider the conditions from Theorem 19. They state that when viewing the objects of $\mathcal{P}$ as finite strings, $\mathcal{P}$ in general has to be closed under the operation of taking a subsequence, and possibly renaming the labels. Furthermore, as an immediate consequence we conclude that $\sim_{\mathcal{P}}$ is a congruence with respect to the aforementioned CCS operators.

What about other choices of $\mathcal{P}$? If—similarly to the choice of $\mathcal{P}_H$ in $\mathcal{P}_M$ in Sect. 2—we choose $\mathcal{P}$ as the subcategory of the previous choice of $\mathcal{P}$ obtained by only keeping identity morphisms and morphisms whose domains are observations having only one state (the empty word), then $\sim_{\mathcal{P}}$ corresponds to Hoare trace equivalence. This choice of $\mathcal{P}$ also trivially satisfies all conditions required by the theorems. Hence, Hoare trace equivalence is a congruence with respect to the presented constructions (and, again, the aforementioned CCS operators).

Choosing $\mathcal{P}$ as, e.g., the subcategory induced by trees will also satisfy all conditions required by the theorems. Hence $\sim_{\mathcal{P}}$, which is a strictly finer equivalence than Milner's strong bisimulation as hinted in [CN95], must also be a congruence with respect to the presented constructions.

## 5 Recursion

For recursion there is no simple way of defining a functor on $\mathcal{M}$ representing Milner's recursion operator. The reason is that one needs some notion of *process variables* which are to be bound by the recursion operator. Some kind of process term language is necessary, as can be seen both in Milner's work [Mil89] and Winskel and Nielsen's [WN95]. However, without introducing a process algebraic term language it is possible to capture a recursion-like operator in a "faithful" way. The restriction is intuitively that free process variable cannot occur under the scope of a parallel composition operator. Such restrictions have been considered by Taubner [Tau89].

First, identify a set of variables *Var* and extend the objects $(S, i, L, \longrightarrow)$ of $\mathcal{M}$ with a partial function $l$ from $S$ to *Var*. Also, we now allow restarting $\mathsf{lts}$s. [5] Furthermore, whenever $l$ is defined on a state $s$, there can be no outgoing transitions from $s$ and morphisms are now required to respect the labelling function $l$.

[5] The only implication of this assumption is, that co-product will have to be handled in a way similar to recursion. We could also have considered a recursion operator which "unfolded" the transition systems, and hence stayed within the non-restarting $\mathsf{lts}$s.

We define $F_X : \mathcal{M} \longrightarrow \mathcal{M}$, which intuitively "binds $X$", on objects as follows. Given $T = (S, i, L, \longrightarrow, l)$, then $F_X(T) = (S', i', \longrightarrow', L, l')$, where

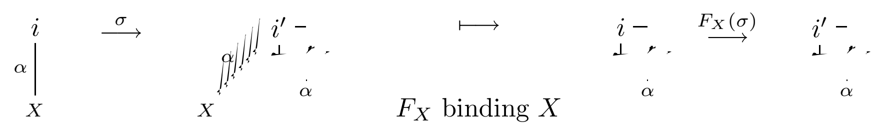
$$S' = \{i\},\ i' = i,\ \longrightarrow' = \emptyset, \text{ and } l' \text{ is totally undefined, when } l(i) = X, \tag{15}$$

$$S' = \{s \in S \mid l(s) \neq X\},\ i' = i,\ l' \text{ equals } l \text{ on } S', \text{ when } l(i) \neq X, \text{ where} \tag{16}$$

$$\begin{aligned} s \xrightarrow{a}{}' s' \ \text{ if } \ & s \xrightarrow{a} s' \ \wedge \ l(s') \neq X \\ & \text{or} \\ & \exists\, s''.\, s \xrightarrow{a} s'' \ \wedge \ l(s'') = X \ \wedge \ s' = i \end{aligned} \tag{17}$$

Given a morphism $f : T_1 \longrightarrow T_2$. $F_X(f) : F_X(T_1) \longrightarrow F_X(T_2)$ is defined to map $s \in S_1'$ to $f(s)$ if $l_2(f(s)) \neq X$, and $i_2'$ otherwise.

Intuitively, $F_X$ simply redirects all transitions going into $X$-labelled states to the initial state. For example:



$F_X$ binding $X$

$F_X$ has the following desirable property:

**Lemma 34.** *For any $X \in Var$, $F_X$ is a functor.*

*Proof.* The proof is routine, hence omitted. ∎

As a special case, let us consider $\mathcal{P}$ as the subcategory of $\mathcal{M}$ corresponding to (10) except that final states may now be labelled with variables from *Var*.

**Theorem 35.** *For any $X \in Var$, $F_X$ is a $\mathcal{P}$-operator.*

*Proof.* The first observation is that (12) is not going to hold. This is due to the fact that an observation of $F_X(T)$ can correspond to many observations of $T$. However, we can apply the theory from Definition 12 on each of these observations individually. So assume $T \xrightarrow{m} T'$ belongs to $\mathcal{M}$ and that

$$\begin{tikzcd} O \arrow[r, dash, "q"] \arrow[d, dash, "f"'] & F_X(T) \arrow[d, dash, "F_X(m)"] \\ O' \arrow[r, dash, "q'"'] & F_X(T') \end{tikzcd}$$

is a commuting diagram in $\mathcal{M}$. Let us denote $f = (\sigma_f, 1_L)$ and use a similar notation for $q$, $q'$, and $m$. Let $O$ be denoted as

$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} s_n$$

and $O'$ as

$$s'_0 \xrightarrow{a_1} s'_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} s'_n \xrightarrow{a_{n+1}} \cdots \xrightarrow{a_{n+m}} s'_{n+m} .$$

Let $1 \leq j_1 < \cdots < j_r \leq n$ be all indexes such that the is no $a_{j_k}$ transition from $\sigma_q(s_{j_k-1})$ to $\sigma_q(s_{j_k})$ in $T$, where $r \geq 0$. This means that for $1 \leq k \leq r$ there exists a transition $\sigma_q(s_{j_k-1}) \xrightarrow{a_{j_k}} r_k$ in $T$ such that $r_k$ is labelled $X$.

Let $j_0 = 0$ and let $U_1, \ldots, U_r$ be observations in $\mathcal{P}$, where for $1 \leq k \leq r$, $U_k$ is given by

$$(j_{k-1}, \sigma_q(s_{j_{k-1}})) \xrightarrow{a_{j_{k-1}}} \cdots \xrightarrow{a_{j_k-1}} (j_k - 1, \sigma_q(s_{j_k-1})) \xrightarrow{a_{j_k}} (j_k, r_k)$$

with final state labelled by $X$ (labelling set $L$, and initial state $(j_{k-1}, \sigma_q(s_{j_{k-1}}))$). We refer to this procedure as *splitting*.

For $1 \leq k \leq r$, let $U'_k$ be the observation

$$(j_{k-1}, \sigma_f(\sigma_q(s_{j_{k-1}}))) \xrightarrow{a_{j_{k-1}}} \cdots \xrightarrow{a_{j_k-1}} (j_k - 1, \sigma_f(\sigma_q(s_{j_k-1})) \xrightarrow{a_{j_k}} (j_k, \sigma_f(r_k))$$

with labelling set $L$. Again, the final state is labelled by $X$. Notice that if $r > 0$, then $\sigma_{q'}(s'_{j_r}) = i'$ in $T'$.

If there exists no $n < k \leq n + m$ such that there is no $a_k$ transition from $\sigma'_q(s'_{k-1})$ to $\sigma'_q(s'_k)$ in $T'$, then choose $r' = 0$ and $U_{r+r'+1}$ as

$$(j_r, \sigma_q(s_{j_r})) \xrightarrow{a_{j_r+1}} \cdots \xrightarrow{a_n} (n, \sigma_q(s_n))$$

where all states are unlabelled, and $U'_{r+r'+1}$ as

$$(j_r, \sigma_{q'}(s'_{j_r})) \xrightarrow{a_{j_r+1}} \cdots \xrightarrow{a_{n+m}} (n + m, \sigma_{q'}(s'_{n+m}))$$

Else, split

$$s'_{j_r} \xrightarrow{a_{j_r+1}} \cdots \xrightarrow{a_{n+m}} s'_{n+m}$$

obtaining indexes $n \leq j_{r+1} < \cdots < j_{r+r'} \leq n+m$, where $r' > 0$, and observations $U'_{j_{r+1}}, \ldots, U'_{j_{r+r'}}$ with final states labelled with $X$. Let $j_{r+r'+1} = n+m$. Let $U_{r+1}$ be the observation

$$(j_r, \sigma_q(s_{j_r})) \xrightarrow{a_{j_r+1}} \cdots \xrightarrow{a_n} (n, \sigma_q(s_n))$$

with all states unlabelled. For $r + 1 < k \leq r + r' + 1$ let $U_k$ be the observation consisting of a single unlabelled state $(j_k, i)$. Let $U'_{r+r'+1}$ be the observation

$$(j_{r+r'}, \sigma_{q'}(s'_{j_{r+r'}})) \xrightarrow{a_{j_{r+r'}+1}} \cdots \xrightarrow{a_{n+m}} (n + m, \sigma_{q'}(s'_{n+m}))$$

with all states unlabelled.

For $1 \leq k \leq r + r' + 1$ let $V_k$ and $V'_k$ denote the unlabelled versions of $U_k$ and $U'_k$, respectively.

Note that for $1 \leq k \leq r + r' + 1$ there exist

– a uniquely determined morphism $f_k : V_k \longrightarrow V'_k$,
– an obvious morphism $q_k : V_k \longrightarrow F_X(T)$, sending a state $(p, s)$ to $s$,
– an obvious morphism $q'_k : V'_k \longrightarrow F_X(T')$,
– a uniquely determined morphism $m_k : U_k \longrightarrow U'_k$,
– an obvious morphism $q_{(k,\#)} : U_k \longrightarrow T$, sending a state $(p, s)$ to $s$,
– an obvious morphism $q'_{(k,\#)} : U'_k \longrightarrow T'$,
– an obvious morphism $q_{(k,\star)} : V_k \longrightarrow F_X(U_k)$, sending a state $(p, s)$ to $s$, and
– an obvious morphism $q'_{(k,\star)} : V'_k \longrightarrow F_X(U'_k)$ .

Now for $1 \leq k \leq r + r' + 1$

$$\begin{tikzcd}
V_k \arrow[r, dash, "q_k"] \arrow[d, dash, "f_k"'] & F_X(T) \arrow[d, dash, "F_X(m)"] \\
V'_k \arrow[r, dash, "q'_k"'] & F_X(T')
\end{tikzcd}$$

commutes. Also, it can be shown that the two diagrams

$$\begin{tikzcd}
V_k \arrow[r, dash, "q_{(k,\star)}"] \arrow[d, dash, "f_k"'] & F_X(U_k) \arrow[d, dash, "F_X(m_k)"] \\
V'_k \arrow[r, dash, "q'_{(k,\star)}"'] & F_X(U'_k)
\end{tikzcd}
\qquad
\begin{tikzcd}
U_k \arrow[r, dash, "q_{(k,\#)}"] \arrow[d, dash, "m_k"'] & T \arrow[d, dash, "m"] \\
U'_k \arrow[r, dash, "q'_{(k,\#)}"'] & T'
\end{tikzcd}$$

commute. Denoting these diagrams as morphisms in $\widehat{\mathcal{M}}$ we can show that the diagram

$$\begin{tikzcd}
\widehat{V_k} \arrow[r, dash, "\widehat{q^\star}"] \arrow[dr, phantom, "\widehat{q_k}"] & \widehat{F_X}(\widehat{U_k}) \arrow[d, dash, "\widehat{F_X}(\widehat{q_{(k,\#)}})"] \\
 & \widehat{F_X}(\widehat{T})
\end{tikzcd}$$

commutes. From the proof of Theorem 10 it follows that there exists morphisms $h_k : V'_k \longrightarrow F_X(T)$, $1 \leq k \leq r+r'+1$, such that $q_k = h_k \circ f_k$ and $q'_k = F_X(m) \circ h_k$. From these morphisms one can then obtain a morphism $h = (\sigma_h, 1_L) : O' \longrightarrow F_X(T)$ such that $q = h \circ f$ and $q' = F_X(m) \circ h$. To see this, let $\sigma_h$ be the function that maps $s'_j$ to $\sigma_{h_k}((j, s'_j))$, when $j_{k-1} < j \leq j_k$, and to $i$, when $j = 0$. It can now be shown that $h$ indeed satisfies the claimed equalities. ■

## 6 Conclusion

We have examined Joyal, Nielsen, and Winskel's notion of behavioural equivalence, $\mathcal{P}$-bisimilarity [JNW93], with respect to congruence properties. Inspired by [WN95], we observed that endofunctors on $\mathcal{M}$ can be viewed as abstract operators. Staying within the categorical setting, we then identified simple [6] and natural conditions, which ensure that such endofunctors preserve open maps, i.e., that $\mathcal{P}$-bisimilarity is a congruence with respect to the functors. We formalised this as $\mathcal{P}$-factorisability. The main varying parameters were $\mathcal{M}$, $\mathcal{P}$, and the functors.

We then continued by giving a concrete application by fixing $\mathcal{M}$. For a set of endofunctors, we obtained meta-theorems stating conditions on $\mathcal{P}$, which guaranteed that $\mathcal{P}$-bisimilarity would be a congruence with respect the functors.

As for future research, there are many possibilities. Returning to the discussion in the introduction, one could try to merge the two "orthogonal" approaches we mentioned, e.g., try to identify a way of presenting functors by SOS-like rule systems such that one could state conditions about both the rule systems and $\mathcal{P}$, which would guarantee congruence of $\mathcal{P}$-bisimilarity with respect to all functors, whose defining rule systems obeyed a special format.

Another possibility is to continue to work as in Sect. 4—other functors may be considered. However, as shown in [NC95], other choices of $\mathcal{M}$ make it possible to capture other interesting behavioural equivalences: weak bisimulation or "true concurrency" equivalences. One could look for similar meta-theorems for such choices of $\mathcal{M}$.

Winskel and Cattani are developing presheaves over categories of observations as models for concurrency [CW96]. For presheaves there are general results on open maps, including the axioms for open maps of Joyal and Moerdijk [JM94], which make light work of showing the bisimulation of presheaves is a congruence for CCS-like languages. Their work exploits universal properties to show preservation of open maps. A condition superficially like $\mathcal{P}$-factorisability is important in transferring such congruence properties from presheaves to other models like transition systems and event structures.

## References


[BIM88] Bard Bloom, Sorin Istrail, and Albert R. Meyer. Bisimulation Can't be Traced. In *Conference Record of the Fifteenth Annual ACM Symposium on Principles of Programming Languages*, pages 229–239, 1988.

[CN95] Allan Cheng and Mogens Nielsen. Open maps (at) work. Research Series RS-95-23, BRICS, Department of Computer Science, University of Aarhus, April 1995. 33 pp.

[CW96] G-L. Cattani and G. Winskel. Presheaf models for concurrency. Technical report, BRICS, 1996. To appear.


[6] We find it a virtue, that the definition of $\mathcal{P}$-factorisability—just as the definition of open maps—doesn't require more than a modest knowledge of category theory.

[Hen88] Matthew Hennessy. *Algebraic Theory of Processes.* MIT Press series in the foundations of computing, 1988.

[JM94] A. Joyal and I. Moerdijk. A completeness theorem for open maps. *Annals of Pure and Applied Logic*, 70:51–86, 1994.

[JNW93] André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation and open maps. In *Proc. LICS'93, Eighth Annual Symposium on Logic in Computer Science*, pages 418–427, 1993.

[Mil89] Robin Milner. *Communication and Concurrency.* Prentice Hall International Series In Computer Science, C. A. R. Hoare series editor, 1989.

[NC95] Mogens Nielsen and Allan Cheng. Observe behaviour categorically. In *Proc. FST&TCS 15, Fifteenth Conference on the Foundations of Software Technology & Theoretical Computer Science*, pages 263–278. Springer-Verlag (*LNCS* 1026), Bangalore, India, December 1995.

[Plo81] Gordon D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN–19, Computer Science Department, Aarhus University, September 1981.

[Tau89] D. Taubner. *Finite Representations of CCS and TCSP Programs by Automata and Petri Nets.* Springer-Verlag (*LNCS* 369), 1989.

[WN95] Glynn Winskel and Mogens Nielsen. *Models for Concurrency*, volume 4, chapter 1, pages 1–148. Oxford University Press, 1995. *eds.* S. Abramsky, D. M. Gabbay, and T. S. E. Gabbay.

This article was processed using the LaTeX macro package with LLNCS style

## Recent Publications in the BRICS Report Series

**RS-96-2** **Allan Cheng and Mogens Nielsen.** ***Open Maps, Behavioural Equivalences, and Congruences*****. January 1996. 25 pp. A short version of this paper is to appear in the proceedings of CAAP '96.**

**RS-96-1** **Gerth Stølting Brodal and Thore Husfeldt.** ***A Communication Complexity Proof that Symmetric Functions have Logarithmic Depth*****. January 1996. 3 pp.**

**RS-95-60** **Jørgen H. Andersen, Carsten H. Kristensen, and Arne Skou.** ***Specification and Automated Verification of Real-Time Behaviour — A Case Study*****. December 1995. 24 pp. Appears in** ***3rd IFAC/IFIP workshop on Algoritms and Architectures for Real-Time Control*****, AARTC '95 Proceedings, 1995, pages 613–628.**

**RS-95-59** **Luca Aceto and Anna Ingólfsdóttir.** ***On the Finitary Bisimulation*****. November 1995. 29 pp.**

**RS-95-58** **Nils Klarlund, Madhavan Mukund, and Milind Sohoni.** ***Determinizing Asynchronous Automata on Infinite Inputs*****. November 1995. 32 pp. Appears in Thiagarajan, editor,** ***Foundations of Software Technology and Theoretical Computer Science: 15th Conference*****, FCT&TCS '95 Proceedings, LNCS 1026, 1995, pages 456–471.**

**RS-95-57** **Jaap van Oosten.** ***Topological Aspects of Traces*****. November 1995. 16 pp. To appear in** ***Application and Theory of Petri Nets: 17th International Conference*****, ICATPN '96 Proceedings, LNCS, 1996.**

**RS-95-56** **Luca Aceto, Wan J. Fokkink, Rob J. van Glabbeek, and Anna Ingólfsdóttir.** ***Axiomatizing Prefix Iteration with Silent Steps*****. November 1995. 25 pp. Appears in Bjerner, Larsson and Nordström, editors,** ***7th Nordic Workshop on Programming Theory*****, NWPT '7 Proceedings, 1995. To appear in** ***Information and Computation*****.**

**RS-95-55** **Mogens Nielsen and Kim Sunesen.** ***Behavioural Equivalence for Infinite Systems - Partially Decidable!*** **November 1995. Full version of paper to appear in Proceedings of the 17th International Conference on Application and Theory of Petri Nets, 1996.**