Is Observational Congruence on μ-Expressions Axiomatisable in Equational Horn Logic?[☆]

Michael Mendler[*] and Gerald Lüttgen

University of Bamberg, Faculty of Information Systems and Applied Computer Sciences, 96045 Bamberg, Germany
{*michael.mendler,gerald.luettgen*}*@uni-bamberg.de*

Abstract

It is well known that bisimulation on μ-expressions cannot be finitely axiomatised in equational logic. Complete axiomatisations such as those of Milner and Bloom/Ésik necessarily involve implicational rules. However, both systems rely on features beyond pure equational Horn logic: either rules that are impure by involving non-equational side-conditions, or rules that are schematically infinitary like the congruence rule which is not Horn. It is an open question whether these complications cannot be avoided in the proof-theoretically and computationally clean and powerful setting of second-order equational Horn logic.

This article presents a positive and a negative result regarding the axiomatisability of observational congruence in equational Horn logic. Firstly, we show how Milner's impure rule system can be reworked into a pure Horn axiomatisation that is complete for guarded processes. Secondly, we prove that for unguarded processes, both Milner's unique fixed point rule and Bloom/Ésik's GA rule are incomplete without the congruence rule, and neither system has a complete extension in rank 1 equational axioms. It remains open whether there are higher-rank equational axioms or other Horn rules which would render Milner's or Bloom/Ésik's axiomatisations complete.

Key words: process algebra, μ-expressions, observational congruence, finite axiomatisation, Horn logic
2000 MSC: 68Q55, 68Q70, 68Q85, 03B70, 03C98

1. Introduction

The existence and nonexistence of equational axiomatisations of behavioural equivalences in process algebra has received significant interest in the literature. Most work is concerned with *finite processes* and *equational axiomatisations* for a range of operators such as for parallel composition [2, 31, 32] and priority [1], and for behavioural semantics such as simulation equivalence [13]. The focus on finite processes is natural since many behavioural relations cannot be finitely axiomatised in the presence of recursion. This has long been known for regular expressions [15]

[☆]An extended abstract of this article appeared in L. Caires and V. T. Vasconcelos, eds., *Concurrency Theory* (CONCUR 2007), vol. 4703 of Lecture Notes in Computer Science, pp. 197–211, Springer–Verlag, 2007.
[*]Corresponding author. Voice: +49 951 863-2828. Fax: +49 951 863-5528.

and has been shown to apply to μ-expressions as well [12, 36]. Except for special and not very well understood situations for bisimulation in the language of $*$-expressions [5, 16, 21, 22, 23], pure equational theories appear to be inadequate for recursive processes.

Thus, a more powerful setting is needed in order to study the relative proof-theoretic complexities of theories for regular processes. A suitable and quite natural setting is provided by *(second-order) equational Horn logic* [33]. Indeed, all known complete axiomatisations for behavioural equivalences on both $*$-expressions and μ-expressions involve *conditional equations*; e.g., these are:

- [26, 34] for $*$-expressions and trace equivalence, generalised to iteration theories (†-expressions) by [17];
- Milner's axiomatisation of strong bisimulation for finite state μ-processes [28] and Bloom/-Ésik's abstract generalisation [10, 19];
- Axiomatisations of *observational congruence* by [9] and [30];
- Van Glabbeek's axiomatisation of branching bisimulation [38];
- The many axiomatisations of bisimulation-style equivalences in timed process algebras such as [4, 7, 14], just to name a few.

Looking at these in detail, however, reveals that they are not strictly Horn theories because they depend on the congruence rule for recursion (cf. rule C4 below) which is not Horn, and they are not pure because rules have guardedness side conditions (cf. rule R2 below).

$$\mathsf{C4}\quad \frac{E = F}{\mu x.\, E = \mu x.\, F} \qquad\qquad \mathsf{R2}\quad \frac{F = E\{F/x\}}{\mu x.\, E = F}\ x \text{ guarded in } E$$

To see that rule C4 is not Horn, consider the soundness of C4 which logically corresponds to the formula $(\forall x.\, E = F) \supset \mu x.\, E = \mu x.\, F$, where '$\mu$' is the recursion operator. This formula is not Horn since the precondition of the implication is universally quantified; in [19], this is called an *implication between equations*. The Horn interpretation of C4 would be $\forall x.\ (E = F \supset \mu x.\, E = \mu x.\, F)$ which is unsound. Take for example $E \equiv a.x$ and $F \equiv a.\mathsf{0}$ where '$a.$' is the standard action prefix operator and $\mathsf{0}$ the (inactive) nil process. Then, the equation $E\{\mathsf{0}/x\} = F\{\mathsf{0}/x\}$ is sound, but the recursive process $\mu x.\, a.x$ is not bisimilar to $\mu x.\, a.\mathsf{0}$. In the Horn theory of closed terms, rule C4 is only *admissible* in the sense that, if all closed instantiations of $E = F$ are derivable, then all closed instantiations of $\mu x.\, E = \mu x.\, F$ are derivable, too. But this rule is infinitary and not expressible in Horn form.

Interestingly, pure Horn axiomatisations have been found for some semantics such as trace equivalence or simulation equivalence, which admit cpo-style denotational models so that recursion can be characterised as a least pre-fixed point. Specifically, [26] introduced a pure Horn system for trace inclusion on $*$-expressions which was later generalised to μ-expressions by [17]. Also, [18] presented a pure Horn axiomatisation of simulation-preorder for μ-expressions. However, bisimulation-style equivalences are not cpo-based. This leads us to the following – in our opinion – key open problem:

> *Can bisimulation for finite–state μ-expressions be axiomatised in pure equational Horn logic?*

The answer to this question relates to the issue of guardedness. On the face of it, C4 appears to be necessary to prove equalities between recursive processes. Consider processes $p =_{\mathrm{df}} \mu x.\, (\alpha.x + \beta.x)$ and $q =_{\mathrm{df}} \mu x.\, (\beta.x + \alpha.x)$, where '+' denotes non-deterministic choice. The processes p and q are bisimilar, and the equation $p = q$ can be derived by first applying the commutativity law on open terms $\alpha.x+\beta.x = \beta.x+\alpha.x$ and then closing them under recursion using C4. Interestingly, for guarded processes, i.e., if both α and β are observable actions, the same is achieved without C4. Using recursive unfolding and commutativity, one derives $p = \alpha.p + \beta.p$ and $q = \alpha.q + \beta.q$, i.e., both p and q provably satisfy the same guarded equation system. From there, by way of rule R2, symmetry and transitivity of equality, one finally gets $p = \mu x.(\alpha.x + \beta.x) = q$. Due to this issue of unguardedness, the above question is particularly challenging for *observational congruence* [29].

The question's importance lies in the fact that the Horn rule format is crucial for standard automated reasoning based on Prolog-style SLD resolution. Moreover, the question is an interesting one since, as we will show,

- Sewell's axiomatisation, which is derived from work of Bloom and Ésik and is commonly considered pure Horn, is in fact impure;
- Milner's axiomatisation, which is commonly considered impure, is in fact pure Horn on *extensional*[1] processes, i.e., guardedness is equational;
- Milner and Sewell's systems cannot be extended to pure Horn on all processes when using only rank-1 equation schemes.

These insights constitute first steps in conquering new ground in Horn axiomatisations for bisimulation-style behavioural equivalences.

As our first technical result, we provide an axiomatisation of observational congruence for finite state processes which is in pure equational Horn form. This axiomatisation is an adaptation of Milner's proof system and interprets the underlying equality as *partial equivalence* via which we may encode the side condition of rule R2. Our axiom system is sound for all processes and complete for guarded processes. Hence, the question remains whether this axiom system can be extended to handle unguardedness.

As our second technical result, we show that no finite rank-1 *equational* extension of Milner's axiom system yields completeness for unguarded processes, not even when including the impure rule R2 or the pure *GA-implication* rule of Bloom/Ésik [11] (as suggested in [35]). There are only few negative results on process-algebraic axiomatisations in equational Horn logic reported in the literature. It is known that, e.g., non-axiomatisability of parallel composition [3] or of priorities [1] can be extended to conditional equations where the conditions are external predicates over time or actions. Here we take this further and permit proper equations as conditions. Our negative result can be generalised to rank 2 and provides a number of technical insights into the proof-theoretic expressiveness of Horn logic for observational congruence. Specifically, we conjecture that unguardedness on μ-expressions cannot be axiomatised in second-order equational Horn logic of any rank.

Organisation. The article's remainder is structured as follows. The next section formally introduces our process language and the key terminology that is used throughout. Sec. 3 then recalls

[1]The notion of extensionality is defined in Sec. 4.

Milner's original axiomatisation of observational congruence which motivates our work. Our first, positive result is presented in Sec. 4, while our negative result on eliminating unguardedness in Horn theories is established in Sec. 5. The concluding section, Sec. 6, discusses these results in the light of related work and points out open problems in the field. Some of the proofs of our results cannot be presented in this special issue due to space constraints, but are included in a technical report [27].

2. The Process Language μBCCSP[2]

This section introduces our process language and makes precise what we understand by a second-order Horn axiomatisation. In particular, the language must be general enough to capture not only the object-level syntax of processes but also the meta-level syntax of schemes and rules needed to formalise logic deduction.[2] Our language μBCCSP[2] is an extension of BCCSP [37] by recursion and schematic variables. It corresponds to the second-order fragment T^2 of [36].

2.1. Second-order Syntax

The second-order language of (schematic, context) *μ-expressions*, or *expressions* for short, is defined by

$$F \quad ::= \quad x(F_1, F_2, \ldots, F_n) \;\mid\; \$\mathsf{k} \;\mid\; \mathsf{0} \;\mid\; \alpha.F \;\mid\; F_1 + F_2 \;\mid\; \mu x.\, F\,.$$

It includes *variables* x and the usual process-algebraic operators of *prefixing* $\alpha.F$, *summation* $F_1 + F_2$ and *recursion* $\mu x.\, F$. The *prefixes* α range over a denumerable set of *observable actions* $a_0, a_1, a_2, \ldots$ and the distinguished *silent action* τ. The constant $\mathsf{0}$ represents the *inactive* process. The expressions $\$\mathsf{k}$, for $k \geq 1$, are *call-back constants* which will be used to form contexts. We let $\equiv$ stand for the syntactic identity on expressions and denote the sub-expression relation by $\trianglelefteq$, i.e., $E \trianglelefteq F$ if either E is a proper sub-expression of F or if $E \equiv F$.

Example 2.1. *The set of sub-expressions of $F =_{df} \mu x.\, (\$1 + \alpha.x)$ is*

$$\{E \mid E \trianglelefteq F\} = \{\$1, x, \alpha.x, \$1 + \alpha.x, \mu x.(\$1 + \alpha.x)\}.$$

Every variable x has a *rank* which specifies the number of parameters that x must be instantiated with to form a process. This is done in *(context) applications* of the form $x(F_1, F_2, \ldots, F_n)$, where $rank(x) = n$. We assume that there is a countably infinite number of variables at every rank. The variable x in $x(F_1, F_2, \ldots, F_n)$ stands for a context with uniquely identified syntactic slots into which the expressions F_i, for $1 \leq i \leq n$, are inserted. These slots are represented by the call-back constants $\$1, \$2, \ldots, \$n$. Formally speaking, call-back constants are nothing but implicitly bound and canonically named process variables. These would be represented as explicit λ-abstractions in higher-order syntax like [36]. The result of *instantiating* x in $x(F_1, F_2, \ldots, F_n)$ by expression F is written $F[F_1, F_2, \ldots, F_n]$ and obtained if each occurrence of $\$\mathsf{k}$ in F is substituted by F_k. Rank 0 variables are called *process variables* and all other variables *schematic variables*. For process variables we simply write x instead of $x()$, and the instantiation of x by F is written F rather than $F[\,]$.

[2] Variable-binding operators require second-order matching in order to handle syntactic contexts such as the bodies F of recursive processes $\mu x.\, F$.

Recursion is possible over process variables only, i.e., we require $rank(x) = 0$ in any expression $\mu x.\, F$. The recursion operator $\mu x.\, F$ binds all occurrences of x in F. There is no variable binder for schematic variables. The notions of *free* and *bound* occurrences of variables and of *guardedness* of variables are as usual. In particular, a variable x is called guarded in an expression F, if all occurrences of x in F are within the scope of an α-prefix with $\alpha \neq \tau$ [29]. An expression F without free variables (of any rank) is *closed*; otherwise it is *open*.

By the *variable rank* of an expressions F we mean the maximal rank of a free variable in F. Since only variables of rank 0 may be bound, closed expressions do not contain variables of rank ≥ 1. We say that F has *context rank n* if it does not contain call-back constants larger than $\$n$. Expressions of context rank 0 are called *process schemes*, and those of higher rank are called *contexts*. Thus, if F has context rank n and all F_i, for $1 \leq i \leq n$, are process schemes, then $F[F_1, F_2, \ldots, F_n]$ is a process scheme. Process schemes without schematic variables, i.e., both context rank and variable rank 0, are called *process terms*. Process terms without free process variables are called *process constants*, or simply *processes*. We use $E, F, \ldots$ to range over general expressions, $t, u, \ldots$ to range over process terms, and $p, q, \ldots$ for process constants.

Example 2.2. *The expression*

$$F \quad =_{df} \quad \alpha.(x + \$1) + \mu y.\, (\$2 + z(x, y, 0))$$

has context rank 2 *and variable rank* 3*. Instantiating the callback constants $\$1$ and $\$2$ in F by the process term $t =_{df} \alpha.\, x$ and process constant $p =_{df} \beta.0$, respectively, yields the process scheme*

$$F[\, t, p\,] \quad \equiv \quad \alpha.(x + \alpha.\, x) + \mu y.\, (\beta.0 + z(x, y, 0)).$$

This scheme has the free process variable x and free schematic variable z. Because of the presence of rank 3 *variable z, expression $F[\, t, p\,]$ is a proper process scheme as opposed to a process term.*

Besides the meta-level identity $E \equiv F$ on expressions we consider formal equalities $E = F$ between process schemes, called *equation schemes*. By the *rank* of an equation scheme $E = F$ we understand the maximal variable rank of E and F. As noted above, the rank of a variable specifies the rank of the context expression by which it needs to be instantiated to generate a process scheme.

An *instantiation* σ is a finite partial mapping from variables to expressions which is rank-preserving, i.e., such that for any variable x in the *domain* of σ, expression $\sigma(x)$ is of rank $rank(x)$. If E is an expression with free variables in X and σ an instantiation with domain X, the *instantiation of E by σ*, written $\sigma(E)$, is obtained by recursively replacing each sub-expression $x(F_1, F_2, \ldots, F_n) \trianglelefteq E$ for $x \in X$ by $\sigma(x)[\sigma(F_1), \sigma(F_2), \ldots, \sigma(F_n)]$.

Example 2.3. *Let E be the process scheme $E =_{df} a.\, x + \mu y.\, w(x + 0, y, z)$ and σ the instantiation defined by*

$$\begin{aligned} \sigma(x) &=_{df} b.\, 0 \\ \sigma(z) &=_{df} a.\, 0 \\ \sigma(w) &=_{df} \$2 + c.\, (\$1 + \$3). \end{aligned}$$

Then, $\sigma(E) \equiv a.\, b.\, 0 + \mu y.\, (y + c.\, ((b.\, 0 + 0) + a.\, 0))$.

The μBCCSP^2 language is general enough to express standard second-order axiom schemes for μ-expressions. For example, schematic recursion unfolding $\mu x.\, t = t\{\mu x.\, t/x\}$ turns into a single equation $\mu x.\, z(x) = z(\mu x.\, z(x))$.

Instantiation is a second-order operation and to be distinguished from the standard first-order *substitution* $E\{F/x\}$ in which variable x is replaced by F in a single recursive pass through E. Instantiations, in contrast to substitution, preserve well-formedness and rank. In particular, if E is a process scheme, then $\sigma(E)$ is again a process scheme.

Example 2.4. *Let x and y be two variables of rank* 0 *and* 1*, respectively, and $E =_{df} x(y)$. Then, the instantiation σ with $\sigma(x) =_{df} y + \$1$ and $\sigma(y) =_{df} 0$ yields $\sigma(E) \equiv \sigma(x(y)) \equiv \sigma(x)[\sigma(y)] \equiv (y + \$1)[0] \equiv y + 0$, while substitution $E\{y + \$1/x, 0/y\}$ would return $(y + \$1)(0)$ which is not well-formed.*

Preserving well-formedness is not enough for instantiations to be sensible in equational reasoning for recursive processes with variable binding. It must be ensured that, in the instantiation $\sigma(E) = \sigma(F)$ of an equation $E = F$, we do not inadvertently capture free process variables inside E or F. An instantiation σ is called *free* for E, if its application $\sigma(E)$ avoids name capture of free process variables, i.e., every occurrence of a free variable in $\sigma(x)$ remains free after instantiation into $\sigma(E)$. We will use symbol θ to range over free instantiations. In practice, there are two options to keep instantiations free. One is to require that θ is *closed*, i.e., for all x in its domain, $\theta(x)$ is closed. The other is to rename bound variables systematically, e.g., by taking expressions up to α-conversion.

2.2. Operational Semantics and Observational Congruence

The semantics of μBCCSP^2 is the transition system induced by process constants as states and where the action–labelled transition relation is inductively defined by the standard (schematic) operational rules:

$$\frac{—}{\alpha.\, x \xrightarrow{\alpha} x} \qquad \frac{x_1 \xrightarrow{\alpha} y}{x_1 + x_2 \xrightarrow{\alpha} y} \qquad \frac{x_2 \xrightarrow{\alpha} y}{x_1 + x_2 \xrightarrow{\alpha} y} \qquad \frac{z(\mu x.\, z(x)) \xrightarrow{\alpha} y}{\mu x.\, z(x) \xrightarrow{\alpha} y}$$

in which x, y are process variables and z has rank 1.

Based on the transition relation, the definition of *observational equivalence* and *observational congruence* [29] are the usual ones. Specifically, $\stackrel{\epsilon}{\Longrightarrow}$ stands for the reflexive and transitive closure $(\xrightarrow{\tau})^*$ of unobservable transitions, and a weak α transition $\stackrel{\alpha}{\Longrightarrow}$ denotes the composition $\stackrel{\epsilon}{\Longrightarrow} \circ \xrightarrow{\alpha} \circ \stackrel{\epsilon}{\Longrightarrow}$ which permits an arbitrary number of unobservable steps before and after α. Further, let $\hat{\alpha} =_{\mathrm{df}} \alpha$, if $\alpha \neq \tau$, and $\hat{\tau} =_{\mathrm{df}} \epsilon$ be the usual action abstraction. A symmetric binary relation $\mathcal{R}$ on process constants is a *weak bisimulation relation* if

$$\forall \langle p, q\rangle \in \mathcal{R}.\ \forall \alpha, p'.\ (p \xrightarrow{\alpha} p' \text{ implies } \exists q'.\, q \stackrel{\hat{\alpha}}{\Longrightarrow} q' \text{ and } \langle p', q'\rangle \in \mathcal{R}).$$

The largest such relation $\approx$ is an equivalence and referred to as *observational equivalence*. The largest congruence $\cong$ contained in $\approx$, called *observational congruence*, is characterised by symmetry and the condition that $p \cong q$ iff

$$\forall \alpha, p'.\ (p \xrightarrow{\alpha} p' \text{ implies } \exists q'.\, q \stackrel{\alpha}{\Longrightarrow} q' \text{ and } p' \approx q').$$

The relation $\cong$ is lifted to process schemes E, F by universal abstraction: $E \cong F$, if $\theta(E) \cong \theta(F)$ for all closed instantiations θ.

2.3. Second-order Equational Horn Logic

If E and F are two well-formed process schemes, then formal equations $E = F$ are of second order, also known as *hyper-identities* [12], due to the presence of schematic variables. A (pure) second-order equational Horn system is a finite set of *Horn rules*, i.e., rules of the form

$$\frac{E_1 = F_1 \quad \cdots \quad E_n = F_n}{E = F}\ , \tag{1}$$

where the $E_i = F_i$ are referred to as the rule's *premises* and $E = F$ as the rule's *conclusion*. If the rule has no premises, i.e., $n = 0$, then it is an *axiom*. Given a finite set $\mathcal{A}$ of Horn rules, we say that an equation scheme $G = H$ is derivable from $\mathcal{A}$, in symbols $\mathcal{A} \vdash G = H$, if there exists a finite sequence of equation schemes $G_0 = H_0,\ G_1 = H_1,\ \ldots,\ G_n = H_n$ such that (a) $G \equiv G_n$ and $H \equiv H_n$; and (b) every equation $G_i = H_i$ is derived by instantiating some Horn rule

$$\frac{E_1 = F_1 \quad \cdots \quad E_m = F_m}{E = F}$$

from $\mathcal{A}$ by way of a free instantiation θ_i such that (i) $\theta_i(E) \equiv G_i$, $\theta_i(F) \equiv H_i$ and (ii) for all $1 \leq s \leq m$, there exists an index $r < i$ satisfying $\theta_i(E_s) \equiv G_r$ and $\theta_i(F_s) \equiv H_r$. Permitting arbitrary free instantiations yields a rather general notion of deduction for Horn theories. In particular, we can derive equations $\mathcal{A} \vdash t = u$ between open process terms. Sometimes we will simply write $\vdash t = u$ where the rules $\mathcal{A}$ involved in the derivation are clear from the context.

Naturally, a theory $\mathcal{A}$ is *sound* if $\mathcal{A} \vdash G = H$ implies $G \approxeq H$, i.e., $\theta(G) \approxeq \theta(H)$ for all closed instantiations θ. For this to hold true, each Horn rule (1) must be *strongly sound* in the sense that

$$\forall \text{ closed instantiations } \theta, \text{ if } \forall i.\, \theta(E_i) \approxeq \theta(F_i), \text{ then } \theta(E) \approxeq \theta(F).$$

This interpretation of soundness, where the universal quantifier over the interpretation of free variables covers the whole rule, is the definitive characteristic of Horn logic. It is important to note that this is very different from requiring that validity of $E_i \approxeq F_i$, for all i, implies validity of $E \approxeq F$, which would be saying that

$$\forall \text{ closed } \theta \text{ and } i,\ \theta(E_i) \approxeq \theta(F_i) \text{ implies that } \forall \text{ closed } \theta.\ \theta(E) \approxeq \theta(F).$$

This is a strictly weaker soundness criterion. The former and stronger Horn-style soundness is the basis for the standard process of Prolog-style SLD resolution, which is known to be complete for Horn theories and ground goals. On open goals $G = H$, the backward proof search generates closed solution instantiations through unification, essentially treating the free variables in the goal as existential or *flexible*. That this works is due to the strong soundness of Horn rules. The only difference to the usual first-order setting of Prolog is that we permit instantiation of schemes by syntactic context functions, which requires *second-order unification*. We refer the reader to [33] for more details on higher-order unification and the proof theory of higher-order Horn logic.

3. Milner's Axiomatisation

For direct reference and to motivate our investigations, we briefly recall Milner's original axiomatisation [30]. To begin with, any algebraic axiomatisation of equality depends on the logical rules of reflexivity, symmetry, transitivity and congruence of equality. These rules, called Refl, Sym, Tran and Cong, are seen in the top part of Table 1. The algebraic structure of summation + and 0 is captured by the equational axioms of commutativity S1, associativity S2, idempotence

Table 1: The equational theory M_0 of finite processes

Equality

$$\mathsf{Refl}\ \frac{-}{x = x} \qquad \mathsf{Sym}\ \frac{x_1 = x_2}{x_2 = x_1} \qquad \mathsf{Tran}\ \frac{x_1 = x_2 \quad x_2 = x_3}{x_1 = x_3} \qquad \mathsf{Cong}\ \frac{x_1 = x_2}{z(x_1) = z(x_2)}$$

Sums

$$\mathsf{S1}\ \frac{-}{x_1 + x_2 = x_2 + x_1} \qquad \mathsf{S2}\ \frac{-}{(x_1 + x_2) + x_3 = x_1 + (x_2 + x_3)} \qquad \mathsf{S3}\ \frac{-}{x + x = x} \qquad \mathsf{S4}\ \frac{-}{x + 0 = x}$$

τ-Laws

$$\mathsf{T1}\ \frac{-}{\alpha.\,\tau.\,x = \alpha.\,x} \qquad \mathsf{T2}\ \frac{-}{x + \tau.\,x = \tau.\,x} \qquad \mathsf{T3}\ \frac{-}{\alpha.x_2 + \alpha.(x_1 + \tau.x_2) = \alpha.(x_1 + \tau.x_2)}$$

S3 and neutrality S4. The τ-laws T1, T2 and T3 express the special properties of τ-prefixes under observational abstraction. Let M_0 be the finite rank-1 Horn theory of Table 1. Here, all x and x_i are process variables, z is a schematic variable of rank 1, α is an arbitrary action and a stands for an action different from τ. Strictly speaking, for finite axiomatisation, a and α must be read as special forms of action variables.

Theorem 3.1 (Hennessy/Milner 1985). *M_0 is sound and complete for $\cong$ in the fragment of μ-*free *processes.*

M_0 is an *equational axiomatisation* of rank 0 since all *non-logical* rules in M_0 are *axioms* in rank 0. The fact that the logical rules Sym, Tran and Cong are Horn rules is conventionally ignored. In order to classify an algebraic theory as 'equational', all that matters is that the *non-logical* part consists of equational axioms. As an aside, note that rule Cong has rank 1 but can be replaced by a finite number of rank 0 rules if each operator of the algebra is treated separately.

Table 2: The classic μ-laws

μ-Laws $\qquad \mathsf{R1}\ \dfrac{-}{\mu x.\,z(x) = z(\mu x.\,z(x))} \qquad \mathsf{R2}\ \dfrac{y = z(y)}{y = \mu x.\,z(x)}$ z guarded context

According to Thm. 3.1, the theory M_0 is fine for finite processes. But what about regular processes involving recursion? As is well known, there is no purely equational axiomatisation for strong bisimulation on μ-expressions [12, 36]. It has been conjectured [36, Sec.4.4, p.79] that there is none for observational congruence $\cong$ either. Our result below (Thm. 5.8) confirms this for rank 1 equations if we exclude the non-Horn congruence rule C4. All known axiomatisations for regular processes and the recursion operator μ involve Horn rules in addition to C4. The simplest such rules are the classic but *impure* μ-laws of Milner, Bergstra and Klop depicted in Table 2.

Rule R1 is known as *μ-unfolding*. It is related to β-reduction in the λ-calculus in that it captures the operational meaning of μ as recursive substitution. The other rule is the *μ-folding* rule R2 which expresses a form of extensionality similar to η-reduction in the λ-calculus and is instrumental for unique fixed point induction. Many axiomatisations of process algebras on

μ-expressions use these two rules or variations thereof. R1 is also called the *Recursive Definition Principle* and R2 the *Recursive Specification Principle* [8] (see also [7]).

Theorem 3.2 (Milner 1989). *$M_g = M_0 \cup \{R1, R2\}$ is sound and complete for $\cong$ in the fragment of* guarded *processes.*

The completeness Thm. 3.2 (and likewise many similar results in the literature) has two deficiencies. Firstly, it only holds for guarded processes and, secondly, M_g is not a pure equational Horn theory because of the guardedness side-condition in rule R2.

To see that soundness of R2 depends on guardedness, consider the instantiation θ defined by $\theta(y) =_{df} \tau.a.0$ and $\theta(z) =_{df} \tau.\$1$. Then, the premise of R2 is valid since $\theta(y) \equiv \tau.a.0 \cong \tau.\tau.a.0 \equiv \tau.\theta(y) \equiv \theta(z(y))$. On the other hand, the conclusion of R2 is false because $\theta(y) \equiv \tau.a.0 \ncong \mu x.\tau.x \equiv \theta(\mu x.z(x))$. Thus, the instantiation of z with the unguarded context $\theta(z)$ would render R2 unsound.

Table 3: Milner's fair abstraction axioms to eliminate unguardedness

Fair Abstraction

$$\text{R3}\ \frac{—}{\mu x.(x + z(x)) = \mu x.z(x)} \qquad \text{R4}\ \frac{—}{\mu x.(\tau.x + z(x)) = \mu x.\tau.z(x)}$$

$$\text{R5}\ \frac{—}{\mu x.(\tau.(x + z_1(x)) + z_2(x)) = \mu x.(\tau.x + z_1(x) + z_2(x))}$$

The restriction of Thm. 3.2 to guarded processes is not necessarily a serious problem in practice since this fragment is expressive complete: every regular (finite-state) behaviour up to $\cong$ can be represented by a guarded process and is thus covered by M_g. Indeed, a second contribution of Milner [30] was to show that adding the three *fair abstraction* axioms (cf. Table 3) yields it possible to transform every unguarded process into a guarded one.

However, closer inspection of Milner's proof reveals that elimination of unguardedness makes use of the congruence rule for recursion

$$\text{C4}\ \frac{z_1(x) = z_2(x)}{\mu x.z_1(x) = \mu x.z_2(x)}$$

which is not a Horn rule as pointed out in Sec. 1. To see that C4 is not strongly sound, consider the instantiation

$$\theta(z_1) =_{df} a.(\$1 + \tau.0) \qquad \theta(z_2) =_{df} a.\tau.0 + \$1 \qquad \theta(x) =_{df} 0.$$

Then, the premise of C4 is valid under θ since

$$\theta(z_1(x)) = a.(0 + \tau.0) \cong a.\tau.0 + 0 = \theta(z_2(x)).$$

Yet, the conclusion of C4 is false because

$$\theta(\mu x.z_1(x)) \equiv \mu x.a.(x + \tau.0) \ncong \mu x.(a.\tau.0 + x) \equiv \theta(\mu x.z_2(x)).$$

Rule C4 is only *weakly sound* and thus transcends the realm of Horn logic. Using rules like C4 means that we can no longer identify process variables and logical variables, which effectively blocks Prolog-style SLD resolution. This prevents the standard assumption that substitutions can be pushed to the leaves of the proof tree. Moreover, reasoning about closed terms involves open terms, which is proof-theoretically not conservative. Many axiomatisations for μ-expressions in the literature tacitly employ the congruence rule C4 as part of 'standard' equational reasoning. However, when it comes to drawing the line between Horn theories and general theories, it is important to make this dependency explicit. For instance, it is interesting to note that C4 is not actually required for completeness in the guarded fragment. Indeed, the proof of Thm. 3.2 as given by Milner only needs congruence in the form of rule Cong (cf. Table 1) which is Horn.

Regarding the deficiencies of the classical axiomatisations such as Milner's, our question of Sec. 1 may now be refined as follows:

- Is there a complete and pure Horn theory for $\cong$ in the fragment of guarded processes, i.e., can the side condition of R1 be handled equationally?
- Is the non-Horn congruence rule C4 necessary to eliminate unguardedness?

In the remainder of this article, we answer the first question to the positive and present results suggesting that the answer to the second question is negative. In the next section we show that the impure side condition of R2 can indeed be eliminated without losing completeness in the sense of Thm. 3.2, by a judicious reformulation of Milner's axioms. Thereafter, in Sec. 5, we prove that it is not possible to extend Milner's axiomatisation without C4 by finite equational axioms such as R3, R4 or R5, so as to become complete for unguarded processes. We also show that the same is true for Bloom/Ésik's axiom system which is reported by Sewell to be a pure Horn theory for strong bisimulation and observational congruence.

4. A Pure Horn Axiomatisation for Extensional Processes

The side condition "z guarded context" in Milner's rule R2 (cf. Table 2) can be eliminated in pure equational Horn logic, provided we are prepared to re-interpret equations so that they only relate *extensional* processes. These processes are defined using the notion of visibility.

The syntactic relation $\trianglerighteq$ of *weak visibility* is the least relation that satisfies the rules $t \trianglerighteq t$ and, if $t \trianglerighteq r$, then $t + u \trianglerighteq r$, $u + t \trianglerighteq r$, $\tau.t \trianglerighteq r$ and $\mu y.\, t \trianglerighteq r\{\mu y.\, t/y\}$. Intuitively, $t \trianglerighteq r$ states that r occurs weakly unguarded in t. Note that $\trianglerighteq$ abstracts from τ-actions unlike the strong form of $\triangleright$ in [30, 36]. A process term t is called *extensional* if there is no term $\mu y.\, u$ such that $t \trianglerighteq \mu y.\, u$ and $u \trianglerighteq y$. Hence, an extensional process term is a process term that cannot engage in an initial divergence, i.e., an initial, infinite sequence of τ-transitions. Moreover, every guarded process is extensional, but not vice versa. However, whenever $\mu x.t$ is extensional, x is guarded in t.

Example 4.1. *The process $\mu x.\, (a.x + b.x)$ is extensional and guarded, whereas $\mu x.\, (a.x + \tau.x)$, $\tau.\mu x.\, (a.x + \tau.x)$ and $\mu x.\, x$ are not extensional and unguarded. The process $a.\mu x.\, x$ is extensional and also unguarded.*

Extensionality is an appropriate weakening of guardedness which admits a finite equational Horn axiomatisation and allows us to reconstruct Milner's original axiomatisation of observational congruence as a *pure* equational Horn theory. Specifically, we will modify all rules of M_g so that they are sound and complete wrt. the *partial equivalence relation* $\cong^e$, where

$$p \cong^e q \quad \text{iff} \quad p \cong q \text{ and } p, q \text{ extensional.}$$

Note that $\cong^e$ is only a partial equivalence relation, i.e., it is transitive and symmetric but not necessarily reflexive. The reflexivity $p \cong^e p$ only holds if p is extensional. The resulting modified system M_g^* is pure Horn and such that p is extensional iff $\mathsf{M}_g^* \vdash p = p$. In particular, $\mu x.\, t$ is extensional iff $\mathsf{M}_g^* \vdash \mu x.\, t = \mu x.\, t$. Since extensionality of $\mu x.\, t$ implies that x is guarded in t, we can replace the side condition of R2 by the equation $\mu x.\, t = \mu x.\, t$.

Table 4: Purified equality rules for extensional processes

Equality

$$\mathsf{Refl1}\frac{-}{0 = 0} \qquad \mathsf{Refl2}\frac{-}{a.x = a.x} \qquad \mathsf{Refl3}\frac{z(\mu x.\, z(x)) = z(\mu x.\, z(x))}{\mu x.\, z(x) = \mu x.\, z(x)} \qquad \mathsf{Sym}\ \frac{x = y}{y = x}$$

$$\mathsf{Tran}\ \frac{x = y \quad y = z}{x = z} \qquad \mathsf{C1}\ \frac{x = y}{\alpha.x = \alpha.y} \qquad \mathsf{C2}\ \frac{x_1 = y_1 \quad x_2 = y_2}{x_1 + x_2 = y_1 + y_2}$$

The new equality rules are given in Table 4. Symmetry Sym and transitivity Tran from M_0 are sound for $\cong^e$ and thus can be taken over directly. The reflexivity rule Refl of M_g, however, is too generous. It derives, e.g., $\vdash \mu x.\, x = \mu x.\, x$, yet $\mu x.\, x$ is not extensional. Clearly, reflexivity must be refined so that it captures extensionality. This is done using the three rules Refl1, Refl2 and Refl3. Similarly, the congruence rule Cong of M_g is unsound for $\cong^e$. For example, if we instantiate $\theta(x_1) = \theta(x_2) =_{\mathrm{df}} 0$ and $\theta(z) =_{\mathrm{df}} \mu x.(x + \$1)$ in Cong, we could derive $\vdash \mu x.(x + 0) = \mu x.(x + 0)$ from $\vdash 0 = 0$ (obtained by Refl1), although $\mu x.(x{+}0)$ is not extensional. Obviously, for soundness we must make sure that congruence can be used only for extensional contexts. This is achieved by restricting congruence to prefix and sum contexts with rules C1 and C2. Both are easily seen to be sound for $\cong^e$.

Example 4.2. *Let $p =_{df} \mu x.(a.x{+}\mu y.(b.y{+}0))$ which is extensional. We show how to derive $\vdash p\downarrow$, where $p\downarrow$ abbreviates reflexivity $p = p$. To this end let $q =_{df} \mu y.(b.y + 0)$, i.e., $p \equiv \mu x.(a.x + q)$. First, use Refl1 to get $\vdash 0\downarrow$ and Refl2 to get $\vdash b.q\downarrow$. From this, C2 obtains $\vdash b.q + 0\downarrow$ to which we can apply Refl3 yielding $\vdash q\downarrow$. Finally, another application of Refl2 gives us $\vdash a.p\downarrow$ so that C2 derives $\vdash a.p + q$. This yields $\vdash p\downarrow$ by Refl3.*

Table 5: Purified sum and τ-theory for extensional processes

Sums

$$\mathsf{S1}^*\ \frac{x_2 + x_1 = y}{x_1 + x_2 = y} \quad \mathsf{S2}^*\ \frac{x_1 + (x_2 + x_3) = y}{(x_1 + x_2) + x_3 = y} \quad \mathsf{S3}^*\ \frac{x = y}{x + x = y} \quad \mathsf{S4}^*\ \frac{x = y}{x + 0 = y}$$

τ-Laws

$$\mathsf{T1}^*\ \frac{\alpha.x = y}{\alpha.\tau.x = y} \quad \mathsf{T2}^*\ \frac{\tau.x = y}{x + \tau.x = y} \quad \mathsf{T3}^*\ \frac{\alpha.(x_1 + \tau.x_2) = y}{\alpha.x_2 + \alpha.(x_1 + \tau.x_2) = y}$$

The standard axioms of commutativity, associativity, idempotence and neutrality, as well as Milner's τ-laws may be rephrased as seen in Table 5, so that they become sound for extensional processes. The key observation here is that all axioms S1–S4 and T1–T3 are equalities $E = F$ which preserve extensionality weakly, in the sense that if one of E or F is extensional then the other is extensional, too. The simplest way to make these axioms sound for $\approxeq^e$ is to rewrite each of these equational axioms R into a *rule* R^* as follows:

$$\mathsf{R}\ \frac{-}{E = F} \quad \mapsto \quad \mathsf{R}^*\ \frac{F = y}{E = y}.$$

Example 4.3. *For instance, the original commutativity axiom S1 would generate the equality $p+q = q+p$ for the non-extensional process $p =_{df} \mu x.x$ and $q =_{df} 0$. This is prevented by* rule *$S1^*$ which requires us to prove the reflexivity $\vdash q + p = q + p$ before we can derive the commutation $\vdash p + q = q + p$. But the former reflexivity means that $q + p$ is extensional and thus $p + q$ is extensional, too.*

The new rules of Tables 4 and 5 constitute a weak extension of the standard equational theory M_0 for finite processes by recursion. It is equivalent to M_0 on μ-free processes but weaker on recursion since it proves $p = p$ for extensional processes only, as shown in Prop. 4.4 below. The crucial advantage of this, however, is that we can turn the guardedness side-condition in the μ-laws of M_g into a proper equation.

Our new μ-laws $\mathsf{R1}^*$ and $\mathsf{R2}^*$ are given in Table 6. Rule $\mathsf{R1}^*$ is derived from R1 as mentioned above. It describes that $\mu x.\, t$ is a solution of the fixed point equation $x = t$. But instead of generating arbitrary recursion unfoldings $\mu x.\, t = t\{\mu x.\, t/x\}$, our formulation uses the conditional form to restrict the unfolding to the cases where $\mu x.\, t$ is extensional. Rule $\mathsf{R2}^*$ mimics R2 and states that *extensional* equations have unique recursive solutions. More precisely, if $p = t\{p/x\}$ and if the fixed point $\mu x.\, t$ is provably identical to some process q, then p and q are identical. Hence, if $\mu x.\, t$ is extensional, then all solutions of the equation $x = t$ are equal to $\mu x.\, t$. The second premise $\mu x.\, z(x) = y$ of $\mathsf{R2}^*$ takes the place of the non-equational side condition "z guarded context" in Milner's R2, since $\mu x.\, z(x) = y$ can only be derived if z is a guarded context.

Table 6: Purified μ-theory for extensional processes

μ-Laws

$$\mathsf{R1}^*\ \frac{\mu x.\, z(x) = y}{z(\mu x.\, z(x)) = y} \qquad \mathsf{R2}^*\ \frac{x = z(x) \qquad \mu x.\, z(x) = y}{x = y}$$

Let M_g^* be the system of rules of Tables 4–6. Observe that M_g^* is a finite and pure equational Horn theory in rank 1.

Proposition 4.4. *A process p is extensional iff $\mathsf{M}_g^* \vdash p = p$.*

Proof. For notational convenience let us abbreviate an equation $t = t$ as $t\downarrow$ in what follows. The proof tacitly uses the fact that extensionality can be decided by the following procedure:

- $\alpha.t$ is extensional iff $\alpha \neq \tau$ or t is extensional.
- $t_1 + t_2$ is extensional iff both t_1 and t_2 are extensional.
- $\mu x.\, t$ is extensional iff t is extensional and x occurs only guarded in t.

Furthermore, $t\{\mu x.\, u/x\}$ is extensional iff t is extensional, and x is guarded in t or $\mu x.\, u$ is extensional.

($\Rightarrow$) We must show that, for every extensional process p, the system M_g^* derives the equation $p = p$. We prove a slightly more general statement: Let t be an extensional process term with at most x as free variable and x guarded in t. We show by induction on the structure of t that $\mathsf{M}_g^* \vdash t\{\mu x.\, u/x\} \downarrow$ for any term u. From the special case $t \equiv p$, where x is trivially guarded, it follows that $\mathsf{M}_g^* \vdash p \downarrow$ whenever process p is extensional. We will leave the reference to M_g^* implicit in the following.

Let t be extensional and x guarded in t. We make a case analysis on the structure of t. Observe that t cannot be variable x since x is assumed to be guarded. Also, by assumption, t cannot be any other variable. Thus, we only need to consider the cases of nil, prefix, sum and recursion:

- If $t \equiv \mathsf{0}$, we prove $t \downarrow$ by way of rule Refl1.
- Suppose $t \equiv a.t'$. Then, we use rule Refl2 to derive $a.t'\{\mu x.\, u/x\} \downarrow$ as desired.
- If $t \equiv \tau.t'$, then x must be guarded in t' which must be extensional. By induction hypothesis, there is a derivation of $t'\{\mu x.\, u/x\} \downarrow$. Using rule C1 yields $\tau.t'\{\mu x.\, u/x\} \downarrow$.
- If $t \equiv t_1 + t_2$, then x is guarded in t_1 and t_2, which are both extensional. By induction hypothesis, there are derivations for $t_1\{\mu x.\, u/x\} \downarrow$ and $t_2\{\mu x.\, u/x\} \downarrow$. From these we obtain $t_1\{\mu x.\, u/x\} + t_2\{\mu x.\, u/x\} \downarrow$ by rule C2.
- Finally, consider the case $t \equiv \mu y.\, t'$. We must show that $(\mu y.\, t')\{\mu x.\, u/x\} \downarrow$ is derivable, where we may assume that y is not free in u and $x \not\equiv y$. Since t is extensional, y is guarded in t' and t' is extensional. This implies that the recursive unfolding $t'\{\mu y.\, t'/y\}$ is extensional. Also, the assumption that x is guarded in t means that it is guarded in t' and thus also in $t'\{\mu y.\, t'/y\}$. The induction hypothesis now yields a derivation of $t'\{\mu y.\, t'/y\}\{\mu x.\, u/x\} \downarrow$. Since
$$t'\{\mu y.\, t'/y\}\{\mu x.\, u/x\} \equiv t'\{\mu x.\, u/x\}\{\mu y.\, t'\{\mu x.\, u/x\}/y\},$$
a suitable instantiation of rule Refl3 obtains a derivation of $\mu y.\, t'\{\mu x.\, u/x\} \downarrow$, as desired.

($\Leftarrow$) Again, we prove a slightly more general statement. For all terms t, u and processes q, if $\vdash t\{\mu x.\, u/x\} = q$ or $\vdash q = t\{\mu x.\, u/x\}$, then $t\{\mu x.\, u/x\}$ is extensional. This is the same as proving that t is extensional, and x guarded in t or $\mu x.\, u$ extensional. Also note that this induction invariant implies, in particular, that if $\vdash p = q$ then both p and q are extensional.

Suppose that one of (i) $\vdash t\{\mu x.\, u/x\} = q$ or (ii) $\vdash q = t\{\mu x.\, u/x\}$ is true. We show by induction on the structure of these derivations that t is extensional and that, in addition, x is guarded in t or $\mu x.\, u$ is extensional. We consider the last rule that was used to derive (i) or (ii), respectively. Obviously, the induction hypothesis yields the result trivially for (i) and for all rules in which the left-hand term in the conclusion equation also appears as the left-hand term or right-hand term in one of the premises. This is the case in rules Sym, Tran and R2*. Regarding (ii), the conclusion is trivially obtained from the induction hypothesis in case of rules Sym, Tran, S1*–S4*, T1*–T3*, R1* and R2*. We verify all remaining cases by individual arguments as follows:

- If any of the cases (i) and (ii) is obtained by rule Refl1, then $t\{\mu x.\, u/x\} \equiv \mathsf{0}$ and thus $q \equiv \mathsf{0}$ and $t \equiv \mathsf{0}$. Hence, t is extensional and x guarded in t.

- If (i) or (ii) are because of rule Refl2, then $t\{\mu x.\, u/x\} \equiv a.p'$ where $t \equiv a.t'$ and $t'\{\mu x.\, u/x\} \equiv p'$. Obviously, t is extensional and x guarded in t.

- Suppose that derivation (i) has C1 as its last rule. Then, $t\{\mu x.\, u/x\} \equiv \alpha.p'$, and $\vdash \alpha.p' = q$ is obtained from a strictly smaller derivation $\vdash p' = q'$ such that $q \equiv \alpha.q'$. In this case $t \equiv \alpha.\, t'$ and $p' \equiv t'\{\mu x.\, u/x\}$. By induction hypothesis, t' is extensional and, if x is unguarded in t', then $\mu x.\, u$ is extensional. But then t is extensional, too, and if x is unguarded in t, it must be unguarded in t', so that $\mu x.\, u$ is extensional, as desired. By symmetry, the same argument applies in case (ii) when $\vdash q = t\{\mu x.\, u/x\}$ is derived using C1.

- Consider that derivation (i) ends in rule C2. Then, $t\{\mu x.\, u/x\} \equiv p_1 + p_2$ and $\vdash p_1 + p_2 = q$ is obtained by two strictly smaller derivations $\vdash p_1 = q_1$ and $\vdash p_2 = q_2$ such that $q \equiv q_1 + q_2$. Substitution distributes with the summation operator, i.e., $t \equiv t_1 + t_2$ such that $p_1 \equiv t_1\{\mu x.\, u/x\}$ and $p_2 \equiv t_2\{\mu x.\, u/x\}$. The induction hypothesis implies that t_1 and t_2 and thus t are extensional. Further, if x is unguarded in t, it must be unguarded in one of t_1 or t_2, whence $\mu x.\, u$ is extensional by induction hypothesis. This was to be shown. Again, case (ii) when $\vdash q = t\{\mu x.\, u/x\}$ is obtained by C2, since the last rule is treated symmetrically.

- Suppose either derivation (i) or (ii) arises by instantiating rule Refl3. Then, $t\{\mu x.\, u/x\} \equiv \mu y.\, v$ for some process variable y and process term v, and the derivation (i) or (ii), which is $\vdash \mu y.\, v \downarrow$, arises from a smaller derivation $\vdash v\{\mu y.\, v/y\} \downarrow$. The induction hypothesis implies that v is extensional and further that $\mu y.\, v$ is extensional if y is unguarded in v. Since $\mu y.\, v$ cannot be extensional without y being guarded, we must have that y is guarded in v. Consequently, $\mu y.\, v$ and thus $t\{\mu x.\, u/x\}$ are extensional, as desired.

- Consider that (i) $\vdash t\{\mu x.\, u/x\} = q$ arises by rule S1* which means $t\{\mu x.\, u/x\} \equiv t_1\{\mu x.\, u/x\} + t_2\{\mu x.\, u/x\}$. Then, there must be a smaller derivation $\vdash (t_2 + t_1)\{\mu x.\, u/x\} = q$, which implies that $(t_2 + t_1)\{\mu x.\, u/x\}$ is extensional by induction hypothesis. It is easy to see that this also implies that $t\{\mu x.\, u/x\}$ is extensional. As pointed out above, case (ii) derived by rule S1*, and all the remaining rules, does not need to be considered. From now on we only deal with case (i).

- Next, suppose $\vdash t\{\mu x.\, u/x\} = q$ arises from rule S2*, i.e., $t \equiv (t_1 + t_2) + t_3$. Applying the induction hypothesis yields that $(t_1 + (t_2 + t_3))\{\mu x.\, u/x\}$ is extensional. It follows that $t\{\mu x.\, u/x\}$ must be extensional.

- If $\vdash t\{\mu x.\, u/x\} = q$ by rule S3*, then $t \equiv t' + t'$ and this must be from a derivation of the equation $t'\{\mu x.\, u/x\} = q$. Now we apply the induction hypothesis to conclude that $t'\{\mu x.\, u/x\}$ is extensional, which means that $t\{\mu x.\, u/x\} \equiv t'\{\mu x.\, u/x\} + t'\{\mu x.\, u/x\}$ is extensional, too.

- If $\vdash t\{\mu x.\, u/x\} = q$ by rule S4*, then $t \equiv t_1 + 0$ and the premise is a smaller derivation $\vdash t_1\{\mu x.\, u/x\} = q$. From here, the induction hypothesis obtains that $t_1\{\mu x.\, u/x\}$ is extensional, from which we conclude without difficulty that $t\{\mu x.\, u/x\}$ must be extensional.

- The next rule to look at is T1*. Here, for case (i), we would have $t \equiv \alpha.\tau.t'$. The induction hypothesis is applied to a derivation of $(\alpha.t')\{\mu x.\, u/x\} = q$, implying that $\alpha.t'$ is extensional by induction hypothesis. This, however, means that t is extensional, too. Moreover, suppose that x is unguarded in t. Then, $\alpha \equiv \tau$, whence x is unguarded in $\alpha.t'$. By induction hypothesis, then, $\mu x.\, u$ is extensional.

- Rules $\mathsf{T2}^*$ and $\mathsf{T3}^*$ are handled by observing that whenever $(\tau.t')\{\mu x.\, u/x\}$ or $(\alpha.(t_1 + \tau.t_2))\{\mu x.\, u/x\}$ are extensional, then so are $(t'+\tau.t')\{\mu x.\, u/x\}$ and $(\alpha.t_2+\alpha.(t_1+\tau.t_2))\{\mu x.\, u/x\}$, respectively.

- Suppose that $t\{\mu x.\, u/x\} = q$ has been derived by instantiating rule $\mathsf{R1}^*$. Hence, there exists a term t' such that $t\{\mu x.\, u/x\} \equiv t'\{\mu x.\, t'/x\}$ and also $\vdash \mu x.\, t' = q$. By induction hypothesis, $\mu x.\, t'$ is extensional. Thus, t' is extensional and x is guarded in t', which finally implies that $t'\{\mu x.\, t'/x\}$ is extensional, as desired. □

It is important to note that the statement of Prop. 4.4 is sensitive to the choice of axioms. Adding axioms to M_g^* may yield provable reflexivities for non-extensional processes, while removing axioms may mean that some extensional processes are not verifiably reflexive any longer.

Example 4.5. *Adding Milner's fair abstraction axioms (cf. Table 3) renders M_g^* unsound. For example, we could use R3 to get $\vdash \mu x.(x + 0) = \mu x.0$ by instantiating $\theta(z) =_{df} 0$. Since we can also derive $\vdash a.0 = a.0 + 0$, rule R2* would yield $\vdash a.0 = \mu x.0$ for instantiation $\theta'(z) =_{df} \$1 + 0$ and $\theta'(x) =_{df} a.0$. But $\vdash a.0 = \mu x.0$ is unsound. For completeness, let us see how $\vdash a.0 = a.0 + 0$ is proven: first use Refl2 for $\vdash a.0 = a.0$; from this S4* obtains $\vdash a.0 + 0 = a.0$ and finally Sym establishes the desired result.*

Theorem 4.6. *M_g^* is sound with respect to $\cong$ for all processes, and it is complete for guarded processes.*

Proof. The proof is a replay of Milner's proof in [30], exploiting the invariant that $\mathsf{M}_g^* \vdash p = q$ iff $\mathsf{M}_g \vdash p = q$ and both p,q are extensional.

Soundness has been argued for above. Specifically, one observes that the side condition of Milner's rule R2 is captured by the equation $\mu x.\, z(x) = y$ in $\mathsf{R2}^*$. For if $\mathsf{M}_g^* \vdash \mu x.\, t = p$, then by symmetry and transitivity, $\mathsf{M}_g^* \vdash \mu x.\, t = \mu x.\, t$, which means that $\mu x.\, t$ is extensional by Prop. 4.4 and thus x is guarded in t.

For completeness one observes that, by Prop. 4.4 and since guardedness implies extensionality, $\mathsf{M}_g^* \vdash p = p$ is derivable for every guarded process, and also that all rules of Milner can be simulated by the associated rule in M_g^*.

Thm. 4.6 is an improvement over Thm. 3.2, not because it covers a larger fragment of processes but because it removes the impurity in the proof system. It implies that the deductive mechanism of equational (second-order, rank-1) Horn logic is sufficient to axiomatise recursion in the fragment of guarded μBCCSP^2 processes. Our result shows that the guardedness side condition can be captured equationally in the form of extensionality. Thus, Milner's rank-1 axiomatisation – counter to common belief – is essentially pure Horn. Note that neither M_g nor M_g^* are complete for unguarded processes. As pointed out before, the restriction of completeness to *guarded* processes does not affect expressiveness. Many process algebras (see, e.g., [7]) and tools are based on guarded recursive specifications, and it is well known that every unguarded process is provably equivalent to a guarded one. As shown in [30], unguardedness can be eliminated. However, as we shall see next, this latter property seems to depend crucially on the presence of non-Horn rule C4 which is implicit in [30].

5. Can Horn Eliminate Unguardedness?

We now show that for unguarded processes, neither Milner's axiomatisation [30] nor Bloom and Ésik's axiomatisation (see Sewell [35]) are complete when leaving out the only non-Horn rule C4. Moreover, both cannot be made complete by adding a finite number of rank-1 equational axioms. We first establish our incompleteness result by considering equational axioms, and then lift it to include the standard recursion rules employed by Milner and Bloom/Ésik.

Our plan is to show that certain sound equations cannot be derived from any finite and sound equational axiomatisation of $\cong$. More specifically, we apply the standard proof-theoretic diagonalisation strategy for proving non-axiomatisability:

1. Find an infinite family of *intensional* equivalences $\cong_n$ which are strictly finer than $\cong$. Typically, $\cong_n$ would compare some purely syntactic aspects parameterised in n.
2. Show that every Horn rule that is sound for $\cong$ must necessarily also be sound for $\cong_n$, for *large enough* n. For equational axioms $E = F$, this specifically means that there exists a bound m with $\forall n > m.\ \forall\theta.\ \theta(E) \cong \theta(F) \Rightarrow \forall\theta.\ \theta(E) \cong_n \theta(F)$.
3. Find an infinite family of processes p_n, q_n so that, for all n, $p_n \cong q_n$ but $p_n \not\cong_n q_n$.
4. Let $\mathcal{A}$ be a finite theory sound for $\cong$. Then, there is a single bound m such that all rules of $\mathcal{A}$ are sound for $\cong_{m+1}$ according to (2). No matter how large m is, there are processes p_{m+1}, q_{m+1} by (3) such that $p_{m+1} \cong q_{m+1}$ but $p_{m+1} \not\cong_{m+1} q_{m+1}$. But then we cannot have $\mathcal{A} \vdash p_{m+1} = q_{m+1}$ for otherwise, by intensional soundness (2), we would have $p_{m+1} \cong_{m+1} q_{m+1}$, contradicting the assumption.

Because of the previous section, it is obvious that the processes p_n and q_n will have to involve non-extensional processes. In fact, we will define these gadgets so that p_n contains a τ-loop of size n of a certain shape, called an *n-noose*, and q_n is the equivalent process *without* this τ-loop. The intensional equivalence $p \cong_n q$ then is the statement that $p \cong q$ and either both p and q have an n-noose or none of them has. This obtains (3) in our case, which is the easy bit. The hard bit is to find a suitable notion of n-noose that is sufficiently robust and syntactically bulky to force intensional soundness (2). Our *n-nooses* described below (cf. Sec. 5.1) will do the trick for arbitrary rank-1 (and somewhat more) equational axioms (cf. Sec. 5.2) and two specific Horn rules, Milner's R2 (cf. Sec. 5.3) and Bloom/Ésik's GA-Implication (cf. Sec. 5.4).

5.1. Pearls, Shells and Nooses

Our gadgets involve choices between pairwise distinct actions a_i, for $0 \le i \le n-1$, where $n \in \mathbb{N}$. Such a choice can be written in a straightforward way, say as the process $q_n =_{\text{df}} \tau.\ \sum_{i=0}^{n-1} a_i.0$, or expressed in a more complex manner through a recursive maze of τ-transitions, each of which postpones the choice without preempting any of the actions a_i. An example for $n = 3$ is seen in Fig. 1. The process $q_3 =_{\text{df}} \tau.(a_0.0 + a_1.0 + a_2.0)$ offers an initial choice of observable actions a_0, a_1, a_2 that is delayed by a single unobservable τ, while the process $p_3 =_{\text{df}} \mu x_0.\ (a_0.0 + \tau.\ \mu x_1.(a_1.0 + \tau.(a_2.\ 0 + \tau.x_1 + \tau.x_0)))$ does the same by wrapping up the observable actions inside two nested τ-loops. Obviously, $p_3 \cong q_3$.

Suppose we are trying to generate p_3 through matching with a process scheme G, i.e., to find an instantiation θ such that $p_3 \equiv \theta(G)$. Each such choice of F and θ decomposes p_3 into fragments that are instantiated through θ and other parts that come from scheme G. In particular, consider the two inner nodes in the syntax graph of p_3 (cf. Fig. 1) from which the transition arrows a_1 and a_2, respectively, emanate. These nodes correspond to the subexpressions $t_3^1 =_{\text{df}} \mu x_1.(a_1.0 +$

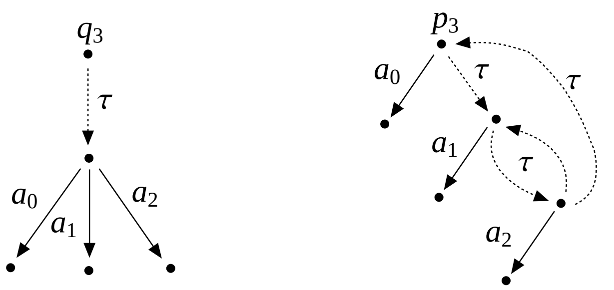

Figure 1: Two equivalent ways of presenting an initial choice of actions a_0, a_1 and a_2.

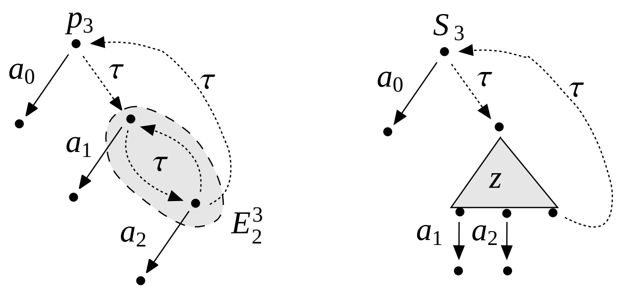

Figure 2: The two inner nodes are inseparable under rank-1 unification.

$\tau.(a_2.\,\mathsf{0} + \tau.x_1 + \tau.x_0))$ and $t^2_3 =_{\mathrm{df}} a_2.\,\mathsf{0} + \tau.x_1 + \tau.x_0$. Note that t^2_3 has two free variables, x_0 and x_1, corresponding to the fact that t^2_3 is part of two τ-loops. This means that if we wanted to separate t^2_3 from t^1_3 in a syntactic decomposition $p_3 = \theta(G)$ so that t^2_3 is introduced by θ and t^1_3 is part of G, then scheme G must contain at least one schematic variable of rank 2 or higher in order to break away the introduction of the two recursion variables $\mu x_0 \cdots \mu x_1 \cdots$ which are to remain in G, from the use of these variables in node t^2_3 which is to be instantiated by θ. To put it the other way round: if we restrict scheme G to have at most rank 1, then we can assume that nodes t^1_3 and t^2_3, and thus the inner τ-loop, are inseparable: either they are both in G or arise both from θ. This is indicated in Fig. 2 on the left. The inner loop, which is abbreviated as $E^3_2 =_{\mathrm{df}} \mu x_1.(\$1 + \tau.(\$2 + \tau.x_1 + \$3))$, behaves *atomically* under rank-1 matching. However, if this context E^3_2 is never broken apart in the (rank-1) handling of process $p_3 \equiv \mu x_0.\,(a_0.\mathsf{0} + \tau.\,E^3_2[a_1.\mathsf{0}, a_2.\mathsf{0}, \tau.x_0])$, then we might just as well replace the context E^3_2 by a rank-3 variable z as shown by the process scheme S_3 in the right of Fig. 2. Assuming that E^3_2 is preserved wholesale in all manipulations of p_3 is the same as saying that all syntactic manipulations $p_3 \mapsto p'_3 \mapsto p''_3 \mapsto \cdots$ can be factorised into transformations $S_3 \mapsto S'_3 \mapsto S''_3 \mapsto \cdots$ and a *fixed* instantiation $\xi_3(z) = E^3_2$ such that $p_3 \equiv \xi_3(S_3)$, $p'_3 \equiv \xi_3(S'_3)$ and $p''_3 \equiv \xi_3(S'''_3)$, etc. The presence of structures like S_3 will be our syntactic invariant to achieve intensional soundness. We will call p_3 a 3-*noose* and S_3 a 3-*shell*. From the generalisation to arbitrary p_n and S_n, we will obtain our intensional equivalence $\cong_n$.

Our terms p_n are constructed from the following family E^n_k of context expressions, indexed by $k \geq 0$ and $n \geq \max(k, 1)$:

$$\begin{array}{rcl} E_0^1 & =_{\mathrm{df}} & \$1 \\ E_0^{i+2} & =_{\mathrm{df}} & \$1 + E_0^{i+1}[\$2, \ldots, \$(i+2)] \\ E_{j+1}^{i+1} & =_{\mathrm{df}} & \mu x_{i-j}.(\$1 + \tau.\, E_j^{i+1}[\$2, \$3, \ldots, \$(i+1), x_{i-j}]), \end{array}$$

where $x_0, x_1, \ldots, x_n$ are pairwise distinct process variables. Each E_k^n is closed and of rank n with k bound variables $x_{n-k}, x_{n-k+1}, \ldots, x_{n-1}$. For example,

$$\begin{array}{rcl} E_3^3 & \equiv & \mu x_0.\,(\$1 + \tau.\, E_2^3[\$2, \$3, x_0]) \\ & \equiv & \mu x_0.\,(\$1 + \tau.\,\mu x_1.\,(\$2 + \tau.\, E_1^3[\$3, x_0, x_1])) \\ & \equiv & \mu x_0.\,(\$1 + \tau.\,\mu x_1.\,(\$2 + \tau.\,\mu x_2.\,(\$3 + \tau.\, E_0^3[x_0, x_1, x_2]))) \\ & \equiv & \mu x_0.\,(\$1 + \tau.\,\mu x_1.\,(\$2 + \tau.\,\mu x_2.\,(\$3 + \tau.\,(x_0 + (x_1 + x_2))))). \end{array}$$

Note that this definition of E_k^n deviates slightly from the structure of what we have called E_2^3 above and represented in Figs. 1 and 2. The generalisation here is technically more convenient but cannot be depicted as easily.

Consider the sequence $\tilde{a} =_{\mathrm{df}} a_0.0, a_1.0, \ldots, a_{n-1}.0$ for which $E_n^n[\tilde{a}] \simeq q_n$. However, as we will see, no finite (rank-1) axiomatisation can derive $E_n^n[\tilde{a}] = q_n$ for every n. The reason is that the syntactic structure of the E_k^n is judiciously chosen in such a way that they behave atomically under second-order syntactic matching. More specifically, in every solution of an equation $w(\tilde{y}) = E_k^n[\tilde{z}]$ for a rank-m variable w and process variables $\tilde{y} = y_1, y_2, \ldots, y_m$ and $\tilde{z} = z_0, z_1, \ldots, z_{n-1}$, the context E_k^n must either be contained wholesale in w or in some y_i, rather than be split across w and $\tilde{y}$. This is made precise by the following proposition (for its proof see [27]):

Proposition 5.1. *Let θ be a free instantiation such that $\theta(w)[\theta(\tilde{y})] \equiv E_k^n[\tilde{U}]$ for rank-m variable w, process variables $\tilde{y} = y_1, y_2, \ldots, y_m$ and schemes $\tilde{U} = U_0, U_1, \ldots, U_{n-1}$. Then, either there exists i with $\theta(y_i) \equiv E_k^n[\tilde{U}]$, or rank m contexts $\tilde{V} = V_0, V_1, \ldots, V_{n-1}$ such that $\theta(w) \equiv E_k^n[\tilde{V}]$ and $V_i[\theta(\tilde{y})] \equiv U_i$, for $0 \leq i \leq n-1$.*

Example 5.2. *Let us consider how the expression $E_2^3[a_1.0, a_2.0, x_0]$ may be matched against the pattern $w(\tilde{y})$ with some instantiation θ, i.e.,*

$$\theta(w)[\theta(\tilde{y})] \;\equiv\; \mu x_1.\,(a_1.0 + \tau.\,\mu x_2.(a_2.0 + \tau.\,(x_0 + (x_1 + x_2))))\,.$$

Recall $E_2^3[a_1.0, a_2.0, x_0] \trianglelefteq E_3^3[\tilde{a}]$, and further assume that θ is free for $E_3^3[\tilde{a}]$, i.e., $\theta(w)$ must not have variable x_0 free. This means that $E_2^3[a_1.0, a_2.0, x_0]$ cannot be generated from a rank-0 pattern w. In rank 1, there is exactly one non-trivial solution to match the pattern $w(y)$, namely $\theta(w) =_{df} E_2^3[a_1.0, a_2.0, \$1]$ and $\theta(y) =_{df} x_0$. Here, 'non-trivial' means that $\theta(w)$ uses at least one call-back but is not identical to it. There are more possibilities for the rank-2 pattern $w(y_1, y_2)$. One of these is $\theta(w) =_{df} E_2^3[a_1.0, \$1, \$2]$, $\theta(y_1) =_{df} a_2.0$ and $\theta(y_2) =_{df} x_0$. Another one is $\theta(w) =_{df} E_2^3[a_1.\$2, a_2.\,\$2, \$1]$, $\theta(y_1) =_{df} x_0$ and $\theta(y_2) =_{df} 0$. The picture is similar for rank-3 pattern $w(y_1, y_2, y_3)$, in the sense that the context E_2^3 is never broken and the call-back arguments $\theta(y_j)$ generate sub-expressions of $a_i.\,0$, with the only constraint that one of $\theta(y_i)$ must be identical to x_0. Now take a look at $E_1^3[a_2.0, x_0, x_1] \trianglelefteq E_3^3[\tilde{a}]$. This time, if θ is to be free for $E_3^3[\tilde{a}]$ again, there is no way in which $E_1^3[a_2.0, x_0, x_1]$ can match the rank-1 pattern $w(y)$. Both variables x_0 and x_1 would have to be introduced by the call-back $\theta(y)$, which is not possible. Moreover, in rank 2 against pattern $w(y_1, y_2)$, we can find a match by setting $\theta(w) =_{df} \mu x_2.\,(a_2.0 + \tau.\,(\$1 + (\$2 + x_2)))$, $\theta(y_1) =_{df} x_0$ and $\theta(y_2) = x_1$.

The importance of Prop. 5.1 is that, if we restrict a scheme G to have variables of at most rank m, then in any matching $\theta(G) \equiv E^n_n[\tilde{U}]$ all contexts E^n_k for $n-k > m$ must either be fully contained in G or fully instantiated via θ from variables in G. In other words, under rank restriction, the E^n_k behave atomically with respect to second-order matching. In this article we shall explore this feature of the indecomposable expressions E^n_{n-1} to prove non-axiomatisability when using only rank-1 schemes. We believe that the families of expressions E^n_{n-m} can be adapted for obtaining non-axiomatisability with respect to maximal rank m, but leave this question open.

Example 5.3. *If we match $E^3_3[\tilde{a}]$ against a scheme G to find an instantiation θ (free for G) so that $\theta(G) \equiv E^3_3[\tilde{a}]$, then, depending on G, either the context E^3_3 is fully contained in G, or some context E^3_k, for $0 \leq k \leq 3$, is fully introduced by θ. An example of the first kind would be $G =_{df} E^3_3[y_0, a_1.y_1, a_2.y_1]$, $\theta(y_0) =_{df} a_0.0$ and $\theta(y_1) =_{df} 0$. Because of what has been discussed above, if E^3_k, for $k = 1, 2, 3$, is to be introduced by θ, we need a variable of rank at least $3 - k$. For instance, $G =_{df} y$ and $\theta(y) =_{df} E^3_3[\tilde{a}]$ would introduce E^3_3 wholesale using a rank-0 variable. Further, $G =_{df} \mu x_0.(a_0.0 + \tau.w(x_0))$ and $\theta(w) =_{df} E^3_2[a_1.0, a_2.0, \$1]$, where w has rank 1, or $G =_{df} \mu x_0.(a_0.0 + \tau.\mu x_1.w(x_1, x_0))$ with $\theta(w) =_{df} a_1.0 + \tau.E^3_1[a_2.0, \$2, \$1]$ for rank 2, are solutions introducing E^3_2 and E^3_1, respectively, through θ.*

To obtain our negative results we must generalise the processes $E^n_n[\tilde{a}] \cong q_n$ so that they become robust against attempts to transform them under equational reasoning for $\cong$. This means that we need to express their essential structural property in slightly more abstract terms. To this end, let Z be a set of variables of rank n and ξ^Z_n the instantiation with domain Z satisfying $\xi^Z_n(z) = E^n_{n-1}$, where $n = rank(z)$. An expression P is called *Z-pure* if P is of rank 0, i.e., a process scheme, and if it does not contain any variables other than those in Z. An action a_i is said to be *i-guarded* in P if each occurrence of a_i appears in the i-th argument S_i of some sub-expression $z(S_1, S_2, \ldots, S_n) \trianglelefteq P$.

Definition 5.4. *An expression P is called an n-*pearl *in* shell variables *Z if*

(**P1**) *P is Z-pure (and all $z \in Z$ have rank n).*

(**P2**) *In every sub-expression $z(S_1, S_2, \ldots, S_{n-1}, U) \trianglelefteq P$, for $z \in Z$, U has a free process variable, and all S_i are process constants such that $S_i \approx a_i.0$.*

(**P3**) *There is at least one occurrence of some $z \in Z$ in P, and each action prefix a_i in P, for $i \geq 1$, is i-guarded.*

*An expression S is an n-*shell *if $P \trianglelefteq S$, for some n-pearl P, and $\xi^Z_n(S) \cong q_n$. A process p is an n-*noose *if there exists an n-shell S such that $p \equiv \xi^z_n(S)$.*

Fig. 3 depicts an example of a 3-pearl S'_3 in shell variables $Z = \{z\}$ which is a special case of a 3-shell. Since the size information n can be derived from the shell variables Z we will simply talk about *pearls* and *shells* in Z. Note that $\xi^Z_n(S) \cong q_n$ implies that shells S can only contain observable actions a_i for $i < n$, and can at most have free variables in Z.

5.2. Nooses Are Equationally Hard

By definition, every n-noose p satisfies $p \cong q_n$. The converse does not hold. Since nooses mix semantic and syntactic properties, they are in general not preserved by observational congruence. For instance, $E^2_2[a_0.0, a_1.0]$ is a 2-noose with shell (and pearl) $S \equiv \mu x_0.(a_0.0 + \tau.z_1(a_1.0, x_0))$

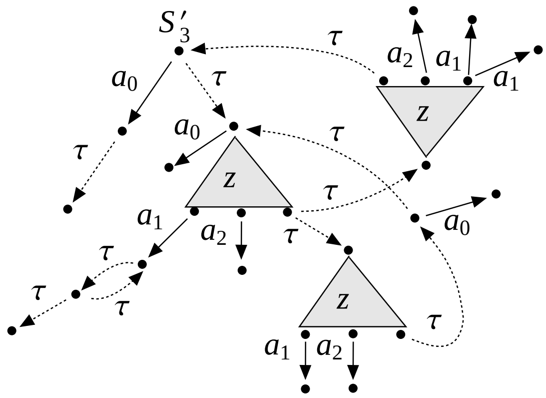

Figure 3: Example of a 3-shell/3-pearl in shell variable z.

and shell variables $Z =_{\rm df} \{z_1\}$, while $q_2 \equiv \tau.(a_0.0 + a_1.0)$ which is observationally congruent to $E^2_2[a_0.0, a_1.0]$ is not a noose. In general, every $E^n_n[\tilde{a}] \cong q_n$ is an n-noose but q_n is not. Our incompleteness result is based on the observation that, although $E^n_n[\tilde{a}]$ may be transformed under equational reasoning, the property of being an n-noose is hard to break up. Once infected by n-nooses for large n, equational transformations in rank-restricted Horn theories cannot get rid of them. The reason for this is that such proofs always factorise through shells that must be preserved by observational congruence. This is the content of the following two key propositions:

Proposition 5.5. *Let E and F be two schemes such that $\theta(E) \cong \theta(F)$ for all instantiations θ. Then, E is an n-shell iff F is an n-shell.*

Proposition 5.6. *Let E be a scheme of maximal recursion depth rd in which all free variables have maximal rank rk. Suppose $rd < 2$ and $rk < n - 2$, or $rd < n - 1$ and $rk < 2$. Then, every instantiation θ such that $\theta(E)$ is an n-noose can be factorised as $\theta = \xi^Z_n \circ \theta'$ for some instantiation θ' and rank n variables Z in such a way that $\theta'(E)$ is a shell in shell variables Z.*

The proofs of Props. 5.5 and 5.6, which can be found in [27], are non–trivial and involve various auxiliary results about the properties of pearls under semantic equivalence transformations and decomposition by second-order unification. However, once Props. 5.5 and 5.6 are established, non-axiomatisability is easily argued along the lines given on page 16 at the beginning of this section. First, we define the *intensional* equivalence $p \cong_n q$ by the condition that $p \cong q$ and that either both p and q have an n-noose or none of them has. Second, one observes that any rank-restricted equational axiomatisation that is sound for $\cong$ must also be sound in the intensional sense for large enough n:

Theorem 5.7. *Let $\mathcal{A}$ be a finite second-order equational axiomatisation of maximal variable rank 1 which is sound for $\cong$. Then, there exists a natural number m such that for all $n \geq m$, $\mathcal{A} \vdash p = q$ implies $p \cong_n q$.*

Proof. Choose m to be larger than the maximal nesting depth of recursions (or the maximal number of free variables) occurring in sub-expressions of any equation of $\mathcal{A}$. Since $\cong_n$ is an equivalence, the rules of reflexivity, transitivity and symmetry are sound for $\approxeq_n$. Hence, the statement of Thm. 5.7 follows directly by induction on the length of derivations $\mathcal{A} \vdash p = q$ if we can show that all equational axioms are sound for $\approxeq_n$. To this end, suppose $E = F$ is an axiom of $\mathcal{A}$ and $p \equiv \theta(E)$ and $q \equiv \theta(F)$ for some instantiation θ. If p is an n-noose, then, by Prop. 5.6 and $n \geq m$, there exists an instantiation θ' such that $\theta'(E)$ is an n-shell and $\theta = \xi_n \circ \theta'$. Since $E = F$ is sound for $\approxeq$, the expression $\theta'(F)$ is observationally congruent to the n-shell $\theta'(E)$. This implies $\theta'(F)$ is an n-shell by Prop. 5.5, whence $\theta(F) \equiv \xi_n(\theta'(F)) \equiv q$ is an n-noose. This proves $p \approxeq_n q$. □

Finally, take the family of processes $p_n =_{\mathrm{df}} E_n^n[\tilde{a}]$. Then, for any natural number n, we have $p_n \approxeq q_n$. Since p_n is an n-noose but q_n is not, by Thm. 5.7, the sound equation $p_n = q_n$, for large enough n, is not derivable in any finite rank-1 equational axiom system. In other words, any finite system of second-order equational axioms of maximal variable rank 1 cannot be both sound and complete for $\approxeq$. This corollary to Thm. 5.7 is in itself not surprising since it is already known, e.g., from the work of Sewell [36], that pure equational logic is insufficient to axiomatise (strong) bisimulation on μ-expressions finitely. However, it is an open problem whether $\approxeq$ can be finitely axiomatised in the more powerful setting of equational Horn logic. As we have seen in Sec. 4, this is indeed possible for the guarded fragment.

In the following we use Thm. 5.7 to derive two negative results, showing that the two well-known Horn-rules considered in [30] and [10] are incomplete. This is because these rules maintain the intensional equivalence $\approxeq_n$.

5.3. Milner's Rule R2

Consider Milner's only rule, the folding rule R2 of rank 1, whose soundness depends on a guardedness side condition:

$$\mathsf{R2} \quad \frac{y = z(y)}{y = \mu x.\, z(x)} \; z \text{ guarded}$$

Theorem 5.8. *There is no finite rank 1 equational extension of Milner's R2 rule, which is sound and complete for $\approxeq$ on unguarded processes.*

The upshot of Thm. 5.8 is that the congruence rule C4 of Milner's original proof system, which is not a Horn rule, cannot be replaced by any number of rank-1 equational axioms without losing completeness on unguarded processes. However, as we have seen in Sec. 4, there is a pure and complete Horn theory for the guarded fragment.

Proof. We may assume that any application of R2 instantiates schematic variable z with a non-trivial guarded context, i.e., in which its argument is called behind observable actions. For any instantiation of R2 in which $\theta(z)$ does not invoke its argument (i.e., does not use call-back \$1) we have $\theta(z)[p] \equiv \theta(z)$, for any p. One can thus derive $\mu x.\, \theta(z)[x] = \theta(z)$ via the unfolding rule R1, and from there obtain the conclusion $\theta(y) = \mu x.\, \theta(z)[x]$ via standard equational reasoning from the premise $\theta(y) = \theta(z)[\theta(y)]$. In other words, R2 becomes redundant for trivial instantiations. Assuming $\theta(z)$ contains \$1 we show that R2 can never produce in its conclusion a term that is an n-noose, for any $n \geq 1$.

Suppose R2 is used with instantiation θ such that $\theta(y)$ is an n-noose. By soundness, we would have $\mu x.\, \theta(z)[x] \approxeq \theta(y) \approxeq \theta(z)[\theta(y)]$. However, this cannot be true. The argument $\theta(y)$ of $\theta(z)$ is guarded by an observable action, say b, so that the right-hand side $\theta(z)[\theta(y)]$ has all noose

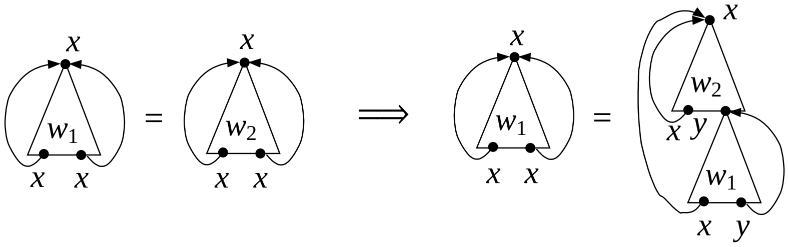

Figure 4: GA-Implication.

actions a_i of the argument $\theta(y)$ accessible behind b. However, the process $\theta(y)$ on the left-hand side, being an n-noose and thus observationally congruent to q_n, does not perform two actions in sequence. Thus, R2 is never applicable when y is instantiated with a process that is an n-noose.

Now suppose that R2 is instantiated so that $\theta(\mu x.\, z(x)) \equiv \mu x.\, \theta(z)[x]$ is an n-noose. Since $\theta(z)$ must have a guarded call-back, this recursion would be able to perform an infinite sequence of actions, which is not possible for nooses. Hence, R2 is sound for $\cong_n$, from which Thm. 5.7 yields Thm. 5.8. □

5.4. Bloom/Ésik's "GA-Implication" Rule

The equational theory of iteration has been studied intensively by Bloom and Ésik, both universally and for special cases such as processes modulo trace equivalence, simulation preorder and bisimulation equivalence. In [11], a range of algebraic axioms and rules are presented for so-called *iteration theories*. These are phrased in the syntax of †*-expressions* which can be understood as categorical (variable-free) process combinators. These †-expressions generalise traditional Kleene ∗-expressions to represent arbitrary (finite) branching structures. Therefore, †-expressions are adequate for bisimulation-style process semantics and equally expressive as μ-expressions. Specifically, [11] presents a pure Horn theory for †-expressions based on a finite number of equational axioms and a single rule scheme called *GA-Implication*. In his thesis [35], Sewell translates this system into the language of μ-expressions and reports results from Bloom and Ésik which show that the system is sound and complete for strong bisimulation. The translation of GA-Implication yields the pure Horn rule

$$\mathsf{GA}\quad \frac{\mu x.\, w_1(x, x) = \mu x.\, w_2(x, x)}{\mu x.\, w_1(x, x) = \mu x.\, w_2(x, \mu y.\, w_1(x, y))}$$

in two rank-2 variables w_1 and w_2. The rule is visualised in Fig. 4. Further, [35] conjectures that Bloom and Ésik's theory, together with the usual τ-laws [30], is also complete for $\cong$. Note that GA like Milner's R2 rule introduces one extra recursion binder into the right-hand side of an equation.

Thus, on the face of it, GA seems to be a way of removing the impurity from Milner's R2. However, Sewell's system still includes the infinitary congruence rule C4 for μ-expressions. Hence, Sewell's rule system is pure but it is not a Horn theory. Indeed, the following result suggests that GA cannot be used to eliminate unguardedness on μ-expressions. Again, the reason is that it preserves large nooses.

Proposition 5.9. *If Bloom/Ésik's rule GA is sound for $\cong$, then it is also sound for $\cong_n$, for all $n \geq 5$.*

By Thm. 5.7, every sound, finite, rank-1 equational axiom system is sound for $\cong_n$, for some large enough n. Since, by Prop. 5.9, rule GA still preserves $\cong_n$, no finite sound equational extension of GA in rank 1 can derive the equality $E_n^n[\tilde{a}] = q_n$ which is not sound under $\cong_n$. Thus, the following theorem holds:

Theorem 5.10. *There is no finite rank* 1*-equational extension of Bloom/Ésik's rule* GA*, which is sound and complete for* $\cong$ *on unguarded processes.*

6. Concluding Discussion and Open Problems

This article studied the logical basis for equational reasoning about observational congruence on μ-expressions. Pure equational logic is too inexpressive for bisimulation semantics on μ-expressions, while second-order equational Horn logic with its generic rule schemes seems powerful enough to admit finite axiomatisations. Indeed, it is well known that finite axiomatisations for this purpose must employ (second-order) rule schemes [36]. However, this does not mean that those axiomatisations are necessarily pure Horn systems. Specifically, as pointed out here, the congruence rule C4 for the μ-binder is beyond Horn logic. C4 is mostly implicit and taken for granted; however, it breaks the purity of Horn logic and the straightforward applicability of Prolog-style resolution techniques. In particular, it makes the convenient identification of object-level process variables and meta-level schematic variables ("shallow embedding") impossible. From the traditional point of view, perhaps, the formal complications due to C4 may be considered minimal. Still, the question must be asked whether bisimulation-style equivalences on μ-expressions can in fact be axiomatised in pure Horn logic and thus enjoy the pleasant model-theoretic and proof-theoretic properties of this rather natural logic setting.

We undertook some important steps towards answering this question. On the positive side, we showed that observational congruence $\cong$ can in fact be axiomatised in pure Horn logic for the fragment of *guarded* processes. On the negative side, we proved that Milner's rule R2 and Bloom/Ésik's rule GA, which are known to be complete in the presence of congruence rule C4, cannot be finitely extended by rank-1 equational axioms for *unguarded* processes to yield a complete system for $\cong$ without C4. This does not preclude the possibility that there may be higher-rank equations or other rules which do the job, though we conjecture that this is not the case. The proofs for the present results turned out to be highly technical and involved subtle issues in managing second-order unification. However, the effort is well spent since negative results in the more powerful setting of equational Horn logic are potentially more interesting than negative results for pure equational logic.

Our results suggest that pure equational Horn systems are intrinsically limited when dealing with unguarded processes under both strong bisimulation and observational congruence. For the latter, however, this is more serious since unguardedness across unobservable actions is nontrivial when these are generated dynamically from communication (as in CCS [29]) or hiding (as in CSP [24]). In fact, our work was triggered by failed attempts to obtain a complete axiomatisation of observational congruence for regular processes in the timed process algebras PMC [6] and CSA [14]. In those languages, unguarded processes carry nontrivial semantic behaviour and thus cannot be ignored in the axiomatisation. If it turned out that unguardedness cannot be Horn axiomatised, this would exhibit the intrinsically more difficult proof-theoretic nature of deterministically timed process algebras under observational abstraction.

We believe that the notions of pearls and nooses introduced in this article can be extended to obtain a negative result for arbitrary rank-k equational schemes. As currently defined, our E_k^n contexts would not survive the so-called “diagonal” or “double iteration” identity $\mu x.\,\mu y.\,w(x,y) = \mu x.\,w(x,x)$ (see, e.g., [19]), which has rank 2. Using this scheme together with rank 1 axioms $\mu x.(y + z(x)) = y + \mu x.\,z(x + y)$ and $\mu x.\,\tau.\,z(x) = \tau.\,\mu x.z(\tau.\,x)$, as well as a finite list of other rank-1 equations for reasoning about processes with a single recursion, would be strong enough to prove $E^n_n[\tilde{a}] = q_n$. However, we conjecture that by re-defining the E^n_k to rank $n{-}k$ so that $E^{i+1}_{j+1} =_{\mathrm{df}} \mu x_{i-j}.(a_{i-j}.\,x_{i-j} + \tau.\,E^{i+1}_j[\$1, \$2, \ldots, \$(\text{i-j}), x_{i-j}])$, for $j \geq 0$, and E^{i+2}_0 as before in Sec. 5, $E^n_n[\tilde{a}] = q_n$ cannot be proved using rank-2 equations. We leave open the general rank-k result and the more difficult case of equational Horn rules other than R2 and GA. We conjecture that the results in this article do not depend on the cardinality of the action set: provided there exists at least one action a, we can replace the actions a_i by pairwise non-congruent processes over the single action a.

Our constructions exploit the interplay between open and closed expressions and the limitations in breaking up recursion cycles imposed by second-order matching using free instantiations. Therefore, our results do not directly apply to †-expressions where congruence C4 is subsumed by the standard rule Cong, i.e., $p = q \Rightarrow p^\dagger = q^\dagger$, which is perfectly Horn. Axiomatisability of observational congruence $\cong$ in the categorical language of †-expressions [11] is a different game that needs separate treatment. The Horn axiomatisability question for †-expression therefore remains open. To this end, it is interesting to note that iteration theories on †-expressions require a system of interface types in order to handle arbitrarily-sized *vectors* of terms coherently. This is another form of meta-level *impurity* which involves computations on natural numbers. Let us indicate this briefly. In the theory of [11], each †-expression f has a syntactic type $f : n \to m$ expressing that f represents an n-vector of processes depending on m input parameters. When translated into μ-expressions, this could be written as $\langle F_1[x_1, x_2, \ldots, x_m], F_2[x_1, x_2, \ldots, x_m], \ldots, F_n[x_1, x_2, \ldots, x_m]\rangle$. The dagger $f^\dagger$ of f is the n-vector of closed expressions obtained by simultaneous recursion over all variables $\mu\vec{x}.\vec{F}[\vec{x}]$, which can be defined by induction on n (see, e.g., [20, p.8], [35, p.23]). An important element in the theory of †-expression is the so-called *weak functorial implication*,

$$h \cdot g = f \cdot (h \oplus \mathbf{1}_p) \quad \Rightarrow \quad h \cdot g^\dagger = f^\dagger.$$

In this rule, the †-expressions $f : n_f \to m_f$, $g : n_g \to m_g$ and $h : n_h \to m_h$, however, cannot be instantiated freely but must satisfy certain type constraints. Specifically, we must have $n_f \leq m_f$, $n_g \leq m_g$, $m_f - n_f = m_g - n_g$, $n_h = n_f$ and $m_h = n_g$. Moreover, h is restricted to so-called (surjective) *base morphisms*, which means they are essentially expressions built from variables only. It is not clear if such side conditions can be put into pure Horn form without extending the syntax. The translation of weak functorial implication into μ-terms reveals that it is really a generalisation of the congruence rule C4; e.g., for $f : n \to n$, $g : 1 \to 1$, $h : n \to 1$ we get the following implication between equations:

$$\begin{aligned}&\forall y.\,G[y] = F_1[y, \ldots, y] \wedge \cdots \wedge \forall y.\,G[y] = F_2[y, \ldots, y]\\ &\qquad \Rightarrow \quad \langle \mu y.\,G[y], \ldots, \mu y.\,G[y]\rangle = \mu\vec{x}.\,\langle F_1[\vec{x}], \ldots, F_n[\vec{x}]\rangle\,,\end{aligned}$$

where $\vec{x} = \langle x_1, x_2, \ldots, x_n\rangle$, which reduces precisely to C4 for $n = 1$. In its general form, the above implication is *impure*, firstly because of the reference to (surjective) base morphisms h and secondly because of the infinite parametricity in the vector width n which requires *unbounded rank*. Also, due to the quantifications $\forall y.\,G[y] = F_i[y, \ldots, y]$ in the premise, the rule is *not Horn*.

One of the reasons for why bisimulation equivalence is difficult to characterise equationally is that it is not a finitary cpo-style semantics. In other process theories, the ordering $p \leq q$ defined by $p + q = q$ is a cpo in which the operators of prefix $\alpha.$ and sum + are continuous. These admit a variation of Milner's rule R2 without side-condition, the weak and strong *Park Induction* [Park 70]:

$$\mathsf{SPI}\ \frac{z(y) \leq y}{\mu x.\, z(x) \leq y} \qquad \mathsf{WPI}\ \frac{z(y) = y}{\mu x.\, z(x) \leq y}$$

These rules express that $\mu x.\, z(x)$ is the least (pre-)fixed point of z. For instance, WPI (SPI) is sound for trace inclusion and simulation preorder, yielding complete finite Horn theories for Kleene algebra [26], (inequational) iteration theories [17] and the semi-lattice algebra of μ-expressions [18]. In fact, one can show that, if we have WPI together with monotonicity of prefix and sum, then unguardedness can be eliminated.

Unfortunately, WPI (SPI) is inconsistent for observational congruence with $p \leq q$ defined as $p + q = q$. To see this, take $\theta(z) =_{\mathrm{df}} \tau.\, \$1$ and $\theta(y) =_{\mathrm{df}} \tau.\, a.\, 0$. Then, by T1, $\theta(z(y)) \equiv \tau.\, \tau.\, a.\, 0 \cong \tau.\, a.\, 0 \equiv \theta(y)$. From here, WPI yields

$$\tau.\, 0 \cong \mu x.\, \tau.\, x \equiv \theta(\mu x.\, z(x)) \;\leq\; \theta(y) \equiv \tau.\, a.\, 0.$$

But $\tau.\, 0 \leq \tau.\, a.\, 0$ means $\tau.\, 0 + \tau.\, a.\, 0 = \tau.\, a.\, 0$, and thus $0 \cong a.\, 0$ which is plainly false. On the other hand, WPI and unguardedness elimination R3 force any partial ordering $p \leq q$ for which + is monotonic to be definable as $p + q = q$. To see why, we first show that WPI and R3 imply the injection $p \leq p + q$:

$$\begin{array}{rcll} (p + x)\{p + q/x\} & \equiv & p + (p + q) & \\ & = & p + q & (\mathsf{S2}, \mathsf{S3}, \mathsf{Cong}, \mathsf{Tran}). \end{array}$$

Therefore, applying WPI for unguarded context $\theta(z) =_{\mathrm{df}} p + \$1$, we derive

$$\begin{array}{rcll} p & = & \mu x.\, (p + x) & (\mathsf{R3}) \\ & \leq & p + q & (\mathsf{WPI}, \mathsf{Tran}), \end{array}$$

as desired. In the same fashion (using also S1), it follows that $q \leq p + q$. Now assume $p \leq q$. Then, $p + q \leq q + q = q$ by monotonicity and S3. Since both $q \leq p + q$ and $p + q \leq q$, we obtain $p + q = q$ because $\leq$ is a partial ordering. Vice versa, if $p + q = q$ then $p \leq p + q = q$ by injection, which was to be shown. All in all, it seems that WPI and R3 are incompatible with observational congruence when the operators are monotonic. This raises the question: is there a preorder $\leq$ for observational congruence so that + is non-monotonic while WPI is sound? If so, can $\leq$ then be axiomatised in Horn logic? This seems unlikely in the face of results like those of [25] on fully-abstract cpo-style denotational semantics for $\cong$, which show that such denotational models involve ideal completions on finite observations which are not finitary in the original syntax.

7. Acknowlegdements

The authors would like to thank the reviewers for their insightful comments and their suggestions for improving the article. The first author was supported by the EU Types network IST 510996.

References

[1] Aceto, L., Chen, T., Fokkink, W., Ingólfsdóttir, A., 2006. On the axiomatizability of priority. In: ICALP'06. Vol. 4052 of LNCS. Springer, pp. 480–491.

[2] Aceto, L., Fokkink, W., Ingólfsdóttir, A., Luttik, B., 2005. CCS with Hennessy's merge has no finite equational axiomatisation. Theoretical Computer Science 330 (3), 377–405.

[3] Aceto, L., Ingólfsdóttir, A., Mousavi, M. R., 2007. Impossibility results for the equational theory of timed CCS. In: CALCO 2007. Vol. 4624 of LNCS. Springer, pp. 80–95.

[4] Aceto, L., Jeffrey, A., 1995. A complete axiomatization of timed bisimulation for a class of timed regular behaviours. Theoretical Computer Science 152 (2), 251–268.

[5] Aceto, L., van Glabbeek, R., Fokkink, W., Ingólfsdóttir, A., 1996. Axiomatizing prefix iteration with silent steps. Information and Computation 127 (1), 26–40.

[6] Andersen, H., Mendler, M., 1994. An asynchronous process algebra with multiple clocks. In: ESOP'94. Vol. 788 of LNCS. Springer, pp. 58–73.

[7] Baeten, J., Middelburg, C., 2002. Process Algebra with Timing. Springer.

[8] Bergstra, J., Klop, J., 1986. Verification of an alternating bit protocol by means of process algebra. In: Bibel, W., Jantke, K. P. (Eds.), Mathematical Models of Specification and Synthesis of Software Systems. Vol. 215 of LNCS. Springer, pp. 9–23.

[9] Bergstra, J. A., Klop, J. W., 1988. A complete inference system for regular processes with silent moves. In: Drake, F. R., Truss, J. K. (Eds.), Logic Colloquium'86. North Holland, pp. 21–81.

[10] Bloom, S., Ésik, Z., 1992. Iteration algebras. Foundations of Computer Science 3 (3), 245–302.

[11] Bloom, S., Ésik, Z., 1993. Iteration Theories: The Equational Logic of Iterative Processes. EATCS Monographs in Theoretical Computer Science. Springer.

[12] Bloom, S., Ésik, Z., 2000. Iteration algebras are not finitely axiomatizable. In: LATIN 2000. Vol. 1776 of LNCS. Springer, pp. 367–376.

[13] Chen, T., Fokkink, W., 2006. On finite alphabets and infinite bases III: Simulation. In: CONCUR'06. Vol. 4137 of LNCS. Springer, pp. 421–434.

[14] Cleaveland, R., Lüttgen, G., Mendler, M., 1997. An algebraic theory of multiple clocks. In: CONCUR'97. Vol. 1243 of LNCS. Springer, pp. 166–180.

[15] Conway, J., 1971. Regular Algebra and Finite Machines. Chapman & Hall.

[16] Corradini, F., de Nicola, R., Labella, A., 2002. An equational axiomatisation of bisimulation over regular expressions. Journal of Logic and Computation 12 (2), 301–320.

[17] Ésik, Z., 1997. Completeness of Park induction. Theoretical Computer Science 177, 217–283.

[18] Ésik, Z., 2000. Axiomatizing the least fixed point operation and binary supremum. In: CSL'00). Vol. 1862 of LNCS. Springer, pp. 302–316.

[19] Ésik, Z., 2001. The equational theory of fixed points with applications to generalized language theory. In: DLT'01. Vol. 2295 of LNCS. Springer, pp. 21–36.

[20] Ésik, Z., 2001. The equational theory of fixed points with applications to generalized language theory. BRICS report series RS-01-22, BRICS Department of Computer Science, University of Aarhus, Denmark.

[21] Fokkink, W., 1994. A complete equational axiomatization for prefix iteration. Information Processing Letters 52 (6), 333–337.

[22] Fokkink, W., 1996. On the completeness of the equations for the Kleene star in bisimulation. In: AMAST'06. Vol. 1101 of LNCS. Springer, pp. 180–194.

[23] Fokkink, W., Zantema, H., 1994. Basic process algebra with iteration: Completeness of its equational axioms. The Computer Journal 37 (4), 259–267.

[24] Hoare, C., 1985. Communicating Sequential Processes. Prentice Hall.

[25] Ingólfsdóttir, A., Schalk, A., 2001. A fully abstract denotational model for observational precongruence. Theoretical Computer Science 254 (1–2), 35–61.

[26] Kozen, D., 1994. A completeness theorem for Kleene algebras and the algebra of regular events. Information and Computation 110 (2), 366–390.

[27] Mendler, M., Lüttgen, G., 2007. Is observational congruence on μ-expressions axiomatisable in equational Horn logic? Tech. Rep. Nr. 72/2007 (57 pp.), Fakultät für Wirtschaftsinformatik und Angewandte Informatik, Universität Bamberg, Germany, `http://www-users.cs.york.ac.uk/~luettgen/publications/pdf/Bamberg-REP-07-72.pdf`.

[28] Milner, R., 1984. A complete inference system for a class of regular behaviours. Journal of Computer and System Sciences 28 (3), 439–466.

[29] Milner, R., 1989. Communication and Concurrency. Prentice Hall.

[30] Milner, R., 1989. A complete axiomatisation for observational congruence of finite-state behaviours. Information and Computation 81 (2), 227–247.

[31] Moller, F., 1989. Axioms for concurrency. Ph.D. thesis, LFCS, University of Edinburgh, U.K., also published as ECS-LFCS-89-84.
[32] Moller, F., 1990. The nonexistence of finite axiomatisations for CCS congruences. In: LICS'90. IEEE Press, pp. 142–153.
[33] Nadathur, G., Miller, D., 1990. Higher-order Horn clauses. Journal of the ACM 37 (4), 777–814.
[34] Salomaa, A., 1966. Two complete axiom systems for the algebra of regular events. Journal of the ACM 13 (1), 158–169.
[35] Sewell, P., 1995. The algebra of finite state processes. Ph.D. thesis, LFCS, University of Edinburgh, U.K., also published as ECS-LFCS-95-328.
[36] Sewell, P., 1997. Nonaxiomatisability of equivalences over finite state processes. Annals of Pure and Applied Logic 90, 163–191.
[37] van Glabbeek, R., 1990. The linear time–branching time spectrum. In: CONCUR'90. Vol. 458 of LNCS. Springer, pp. 278–297.
[38] van Glabbeek, R., 1993. A complete axiomatization for branching bisimulation congruence of finite state behaviours. In: MFCS'93. Vol. 711 of LNCS. Springer, pp. 473–484.