Axiomatization of a Class of Parametrised Bisimilarities

Pranav Singh[1], **S. Arun-Kumar**[2]

[1] *Department of Applied Mathematics and Theoretical Physics*
University of Cambridge, Wilberforce Road
Cambridge, CB3 0WA, United Kingdom
singh.pranav@gmail.com

[2] *Department of Computer Science and Engineering*
Indian Institute of Technology Delhi
Hauz Khas, New Delhi 110 016,India
sak@cse.iitd.ernet.in

Abstract

The question of when two nondeterministic concurrent systems are behaviourally related has occupied a large part of the literature on process algebra and has yielded a variety of equivalences (and congruences) and preorders (and precongruences) all based on the notion of bisimulations.
Recently one of the authors has tried to unify a class of these bisimulation based relations by a parametrised notion of bisimulation and shown that the properties of the bisimilarity relations are often inherited from those of the underlying relationships between the observables. In addition to the usual strong and weak bisimilarity relations, it is possible to capture some other bisimilarity relations – those sensitive to costs, performance, distribution or locations etc – by parametrised bisimulations.
In this paper we present an equational axiomatization of all equivalence relations that fall in the class of parametrised bisimilarities without empty observables. Our axiomatization has been inspired by the axiomatization of *observational congruence* by Bergstra and Klop and attempts to extend it for parametrised bisimilarities. The axiomatization has been proven to be complete for finite process graphs relative to a complete axiomatization for the relations on observables. In the process, we also show that in the absence of empty observables, all preorders and equivalence relations are also precongruences and congruences, respectively.

Keywords: axiomatization, bisimulation, concurrency, process-graphs, proof-system

1 Introduction

It is a well known and accepted fact that the notion of trace equivalence defined in automata theory is too coarse for comparing processes and cannot capture the interactive or communicating behavior that characterize the notion of processes. In particular, it cannot distinguish between the processes $a.(\mathbf{0}+b.\mathbf{0})$ and $a.\mathbf{0}+a.b.\mathbf{0}$. While after taking an a-step the second process can always take a b-step, the first process might become $\mathbf{0}$ and stop. The notion of strong bisimilarity [10], however, distinguishes between these processes. Clearly, this difference between trace equivalence and strong bisimilarity can be seen to stem from the difference in their handling of the meaning of the '+' operator and not from any particular structure of the alphabet of *actions* or *observables* $\{a, b, c \ldots\}$ that they might assume. The notion of observational equivalence [10] follows a definition similar to that of strong bisimilarity, with the additional assumption that the set of actions contains a unique invisible action τ. It differs from trace equivalence in terms of its handling of the semantics of '+' as well as in its special handling of the invisible action τ.

The observation that the structural nature of the definitions of such relations on processes (including their handling of the semantics of operators like '+','.','|' and recursion), and the structure on observables (for example the special handling of τ in observational equivalence) can be viewed separately in this manner, formed the motivation for a generalized notion called (ρ, σ)-bisimilarities [2]. In [2] and [5], many relations such as strong bisimilarity, observational equivalence, performance prebisimulation [7] and pomset prebisimulation [1] have been shown to be special cases of the (ρ, σ)-bisimilarities. While the strength of this generalization is obvious from the wide variety of relations that have been shown to fall under its umbrella, perhaps of greater importance is the manner in which it cleanly separates properties that are arrived at by virtue of the structural nature of its definition (including the semantical interpretation of the choice '+', prefix '.', parallel '|' and recursion) from the properties induced by the nature of relations on observables ρ and σ.

In the large class of (ρ, σ)-bisimilarities, the structure of the definition remains fixed and is similar to that of strong bisimilarity. A large number of properties of these relations can be deduced simply from the nature of this definition, making little or no assumptions about the observables or relations on the observables. Another class of properties are those inherited from the properties of the relations on observables ρ and σ. Of particular interest is the property that a (ρ, σ)-bisimilarity is an equivalence relation if and only if ρ is a preorder and $\sigma = \rho^{-1}$ [5].

The question of when two processes are equal in the context of a particular equivalence relation has occupied a large part of the literature in process algebra. A complete axiomatization of an equivalence relation yields both, a neat algebraic handle on processes and a proof system within which the equality of any two processes expressible in the system can be proven formally. While complete axiomatizations have been provided for some of the most common relations in process algebra, they have all had to be proven sound and complete individually, each involving a distinct method of proving completeness. Also, while the similarity in the axioms of these relations has often been noted, it has hardly been studied in detail. A schema of complete axiomatizations for (ρ, σ)-bisimilarities would automatically provide an axiomatization for any relation that can be proven to fit within the generalized framework of (ρ, σ)-bisimilarities. Apart from saving a great deal of effort in arriving at an axiomatization for each such relation and proving its soundness and completeness separately, such an axiomatization schema highlights the properties that are arrived at as a consequence of the structural nature of processes and relations.

Salomaa's axiomatization of regular expressions [11] has served as a motivation for the axiomatizations of many relations in process algebra. Milner's axiomatization [9] $\mathcal{A}$ for strong bisimilarity over CCS is one. A comparison of these two axiomatizations highlights the difference in the handling of the structural operator '+'. The left distributivity of '.' over '+' that is present in Salomaa's axiomatization cannot be proven in Milner's $\mathcal{A}$ and captures the fundamental difference between strong bisimilarity and trace equivalence. Another axiomatization of strong bisimilarity BPA_{LR} by Bergstra and Klop [6] uses the notion of linear-recursive (LR) expressions instead of the CCS notation and is closer to [11] in terms of its use of recursion equations. BPA_{LR} also closely corresponds to Milner's $\mathcal{A}$ as proven in [6], and was intended as a stepping stone towards formulating $BPA_{\tau LR}$, a complete axiomatization for observational congruence. At around the same time as [6], Milner gave a complete axiomatization [8] $\mathcal{A}_\tau$ for observational congruence on CCS processes.

In this paper we present the axiomatization (schema) $\mathcal{A}_{(\rho,\rho^{-1})}$ where ρ is a preorder. Our axiomatization is built on top of the axiomatization BPA_{LR} for strong bisimilarity [6] and instead of CCS uses LR expressions, which are recursion equations and can also be thought of as term rewrite systems. Our choice is motivated by the close and intuitive correspondence of LR expressions with process graphs, as well as the fact that our proof of completeness relies on a method of saturation similar to the method of Δ-saturation employed for the completeness proof of $BPA_{\tau LR}$ in [6]. However, we see no reason why a simple CCS counterpart of the axiomatization cannot be formulated. We also assume that the set of observables does not contain any empty action.

On lines similar to those of BPA_{LR} and $BPA_{\tau LR}$, we focus on finite or regular process graphs and do not handle the parallel composition construct or the notion of communication. The axiomatization $\mathcal{A}_{(\rho,\rho^{-1})}$ is proved to be complete for all finite process graphs and characterizes all equivalence relations that fall in the class of (ρ,σ)-bisimilarities. The completeness of the axiomatization is proved relative to a complete axiomatization of the relation ρ on observables. The completeness also depends on the completeness of the axiomatization BPA_{LR} for strong bisimilarity, which is presented and proven complete over finite process graphs in [6]. Similarity of the structural definitions of strong bisimilarity and (ρ,σ)-bisimilarities leads to an axiomatization which makes minimal assumptions about the nature of ρ and closely corresponds to BPA_{LR} .

The rest of the paper is organized as follows. In section 2, we establish the notations for labelled transition systems and process graphs [6] used in this paper. In section 3, we summarize the notion of (ρ,σ)-bisimilarities and some results of [2] and [5] that will be of particular relevance to us. Using a very weak assumption, we show that all (ρ,σ)-bisimilarities that are preorders are precongruences while those that are equivalences are congruences. We define the terms of BPA_{LR} and therefore of $\mathcal{A}_{(\rho,\rho^{-1})}$ in section 4, while also quoting some of the important properties of LR expressions from [6]. Having established the required notations, we present the axiomatization $\mathcal{A}_{(\rho,\rho^{-1})}$ in section 5. The proof of soundness of the axiomatization, which largely follows from the soundness of BPA_{LR}, is treated in section 6, while sections 7 and 8 deal solely with the proof of completeness, with the former introducing our notion of graph saturation. In section 8 we formally prove the properties of saturated graphs within the context of the axiomatization, which leads to a proof of completeness. Section 9 is the concluding section.

2 Process Graphs

The notion of process graphs here is based on [6] to which we refer the reader for a more detailed study than is possible to provide here. Process graphs, or graphs for short, are connected, rooted labelled multidigraphs. That is, a process graph g has a root g_r or $root(g)$ and a set of nodes $nodes(g)$ (ranged over by $p, q, \ldots$) with labelled and directed edges between the nodes such that there may be several edges between any two nodes and every node is accessible from the root g_r. The edges in a process graph are labelled by *labels* from the set of observables $\mathcal{O}$ (ranged over by $a, b, \ldots$). A *transition* is a tuple (p, a, q) consisting of a *starting node* p, an *ending node* q and a label a and is also written as $p \xrightarrow{a} q$ for convenience. A transition $p \xrightarrow{a} q$

is a *transition in the graph* g if $p, q \in nodes(g)$ and there is an edge labelled a starting at p and ending at q in the graph g. A node is a *terminal state* if it has no outgoing edges. Note that transitions and edges are considered different. $p \xrightarrow{a} q$ is a transition whereas $\xrightarrow{a}$ is an edge between p and q. $trans(g)$ (ranged over by $\pi, \chi, \ldots$) is the set of all transitions in the graph g.

We sometimes prefer the use of labelled transition system (LTS) notation over process graphs. In this notation a process graph g is written as $\langle nodes(g), \mathcal{O}, trans(g), root(g)\rangle$ with $nodes(g)$ forming the set of states of the LTS, $\mathcal{O}$ being the set of symbols, $trans(g)$ the transition relation and $root(g)$ the initial state.

If $p \in nodes(g)$ then $(g)_p$ is the subgraph of g with p as the root and all the nodes and edges accessible from p in the obvious way. We weaken this boundary by identifying the subgraph $(g)_p$ with the node p. We therefore consider both process graphs and nodes as representation of processes. We weaken the boundary further by identifying them with the processes they represent. Two graphs g and h may have the same node names, in which case we use the subscripts g and h for distinction. p_g is a node in g and p_h is a node in h. Note that these nodes represent different processes $(g)_p$ and $(h)_p$ respectively. Graphs differing only in their naming of the nodes are considered to be identical.

$\mathbb{G}$ (ranged over by $g, h, \ldots$) forms the set of all process graphs while $\mathbb{R}$ is the set of all *finite process graphs* [6] or *regular process graphs.* In this paper we restrict ourselves primarily to the study of finite process graphs.

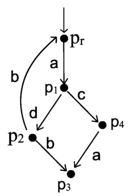

Figure 1: A process graph

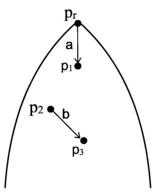

Figure 2: Partial view of a process graph

While drawing process graphs we may either depict all nodes and edges (as in figure 1), or choose to depict only the parts that are of immediate concern to us (as in figure 2).

We indicate nodes in two graphs to be related by drawing a dashed or dotted line or arc between them 3. Other notational conventions we use are the following.

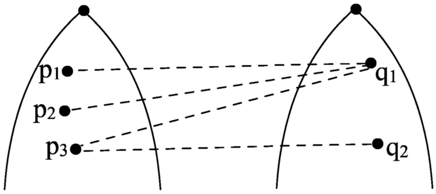

Figure 3: depicting relations

- $\equiv$ for the identity relation on a set. It may be used in the context of observables, processes and also process graphs.
- id_g for the identity relation on the nodes of the process graph g. If two process graphs g and h have the same node names then id_g (or equivalently id_h) could also be used to mean the bijection between these nodes.
- $id_{trans(g)}$ for the identity relation on the transitions of the process graph g.
- $R : pSq$ to indicate $R \subseteq S$ and $(p, q) \in R$ where the relation R may be seen to act as a "proof" of the relation pSq.
- $\circ$ to denote relational composition.

3 (ρ, σ)-Bisimulations

Definition 1. *Let* $\mathbf{P}$ *be the set of processes and let* ρ *and* σ *be binary relations on* $\mathcal{O}$. *A binary relation* $R \subseteq \mathbf{P} \times \mathbf{P}$ *is a* (ρ, σ)**-induced bisimulation** *or simply a* (ρ, σ)**-bisimulation** *if* pRq *implies the following conditions.*

$$\forall a \in \mathcal{O}[p \xrightarrow{a} p' \Rightarrow \exists b, q'[a\rho b \wedge q \xrightarrow{b} q' \wedge p'Rq']] \tag{1}$$

and

$$\forall b \in \mathcal{O}[q \xrightarrow{b} q' \Rightarrow \exists a, p'[a\sigma b \wedge p \xrightarrow{a} p' \wedge p'Rq']] \tag{2}$$

The largest (ρ, σ)*-bisimulation (under set containment) is called* (ρ, σ)**-bisimilarity** *and denoted* $\sqsubseteq_{(\rho,\sigma)}$. *A* $(\equiv, \equiv)$*-induced bisimulation will sometimes be called a* **natural bisimulation***.

In proposition 2 we quote a few important properties of (ρ, σ)-bisimulations and refer the reader to [2] and [5] for their proofs.

*A strong bisimulation on CCS processes with $\mathcal{O} = Act$ is an example of a *natural bisimulation.*

Proposition 2. (Properties).

1. **Monotonicity.** *If* $(\rho,\sigma) \subseteq (\rho',\sigma')$ *pointwise then every* (ρ,σ)*-bisimulation is also a* (ρ',σ')*-bisimulation and hence* $\sqsubseteq_{(\rho,\sigma)} \subseteq \sqsubseteq_{(\rho',\sigma')}$.
2. **Inversion.** *Since* $R : p \sqsubseteq_{(\rho,\sigma)} q$ *implies* $R^{-1} : q \sqsubseteq_{(\sigma^{-1},\rho^{-1})} p$*, we have* $\sqsubseteq_{(\rho,\sigma)} = \sqsubseteq_{(\sigma^{-1},\rho^{-1})}$.
3. **Composition.** $\sqsubseteq_{(\rho_1,\sigma_1)} \circ \sqsubseteq_{(\rho_2,\sigma_2)} \subseteq \sqsubseteq_{(\rho_1\circ\rho_2,\sigma_1\circ\sigma_2)}$ *since* $R_1 : p \sqsubseteq_{(\rho_1,\sigma_1)} q$ *and* $R_2 : q \sqsubseteq_{(\rho_2,\sigma_2)} r$ *implies* $R_1 \circ R_2 : p \sqsubseteq_{(\rho_1\circ\rho_2,\sigma_1\circ\sigma_2)} r$.
4. **Reflexivity.** *If* ρ *and* σ *are both reflexive then the identity relation* $\equiv$ *on* $\mathbf{P}$ *is a* (ρ,σ)*-bisimulation and consequently* $\sqsubseteq_{(\rho,\sigma)}$ *is reflexive.*
5. **Symmetry.** ρ *and* σ *are both symmetric implies the converse of each* (ρ,σ)*-bisimulation is a* (σ,ρ)*-bisimulation. In addition, if* $\rho = \sigma$ *then* $\sqsubseteq_{(\rho,\sigma)}$ *is a symmetric relation.*
6. **Transitivity.** *If* ρ *and* σ *are both transitive then the relational composition of* (ρ,σ)*-bisimulations is a* (ρ,σ)*-bisimulation, and* $\sqsubseteq_{(\rho,\sigma)}$ *is also transitive.*
7. **Preorder characterisation.** $\sqsubseteq_{(\rho,\sigma)}$ *is a preorder iff* ρ *and* σ *are both preorders.*
8. **Equivalence characterisation.** $\sqsubseteq_{(\rho,\sigma)}$ *is an equivalence iff* ρ *and* σ *are both preorders and* $\sigma = \rho^{-1}$.

■

A question that arises in the context of (ρ,σ)-bisimilarities is whether there is a characterization of the smallest congruence or precongruence that contains a particular (ρ,σ)-bisimilarity. We answer this question to a very large extent in lemma 5. Lemma 5 demands an accurate definition of what is considered an empty observable.

Definition 3.

An observable ϵ *is considered* ***empty*** *if* $\forall p,p' \in \mathbf{P}[p \equiv p' \iff p \xrightarrow{\epsilon} p']$ *and corresponds to an extended operational semantics of CCS with the additional rule* Empty: $p \xrightarrow{\epsilon} p$ *along with a restriction of the prefix rule* Act *in figure 4 on* $a \in \mathcal{O} - \{\epsilon\}$.

Act	$a.e \xrightarrow{a} e$
Sum1	$e \xrightarrow{a} e' \Rightarrow e + f \xrightarrow{a} e'$
Sum2	$e \xrightarrow{a} e' \Rightarrow f + e \xrightarrow{a} e'$
Rec	$e\{\mu\ x : e/x\} \xrightarrow{a} e' \Rightarrow \mu\ x : e \xrightarrow{a} e'$

Figure 4: Operational Semantics of CCS expressions

Lemma 4. *Some consequences of having an empty observable.*

1. *A transition may take place without destroying choice. That is,* $\exists \epsilon \in \mathcal{O} : [p + q \xrightarrow{\epsilon} p + q]$.
2. *A transition of a child of a choice expression need not result in a transition of the choice expression. That is,* $\exists \epsilon \in \mathcal{O} : [p \xrightarrow{\epsilon} p' \ \wedge \ p + q \not\xrightarrow{\epsilon} p']$.
3. *There is a special observable that cannot occur as a label of transitions between arbitrary nodes. That is,* $\exists \epsilon \in \mathcal{O} : \nexists\, p, p' \in \mathbf{P} : [p \not\equiv p' \ \wedge \ p \xrightarrow{\epsilon} p']$.

■

Lemma 5.

If $\mathcal{O}$ *does not contain an empty observable, then over the CCS operators '+', '.' and 'μ'*

1. ***Precongruence.*** *Every* $\sqsubseteq_{(\rho,\sigma)}$ *that is a preorder is also a precongruence.*
2. ***Congruence.*** *Every* $\sqsubseteq_{(\rho,\sigma)}$ *that is an equivalence is also a congruence.*

■

The proof of lemma 5 is along the lines of a similar congruence proof in [4]. In the presence of empty observables, the proof does not work out because of part 2 of lemma 4. It should be noted that the characterization of observational equivalence provided in [5] uses $\mathcal{O} = Act^*$ where the empty string can be easily seen to be an empty observable. This is the reason for observational equivalence not being a congruence. In this paper we will concern ourselves only with the cases where $\mathcal{O}$ has no empty observables. While the preorders and equivalences have been proven to be congruences over operators of CCS, it is not hard to see that the same holds for these over LR expressions. Our preference for CCS notation over LR expressions here, is for the sake of brevity.

4 Syntactical Representations

While process graphs are convenient graphical representations of processes and provide the semantics of a process, we also need a linear representation of processes that is syntactically convenient. Process algebras have traditionally evolved with a stronger focus on the syntactical representations because of the convenience of representation and preciseness of argument they allow. The notation of CCS introduced by Milner in [10] and LR expressions introduced by Bergstra and Klop in [6] are more convenient syntactically because of their compactness of representation that allows a more precise reasoning than graphically rich representations such as process graphs do. The semantics of processes are given here in terms of process graphs.

The axiomatizations of relations, even if such relations are defined over process graphs, are therefore given in one of the syntactical representations. We introduce Linear Recursive (LR) expressions, which have been used in [6] and will be the preferred syntactical representation of processes in this paper.

4.1 LR Expressions

LR expressions, where LR stands for Linear-Recursive, are the recursive terms of the system BPA_{LR} introduced in [6]. Since the axiomatization $\mathcal{A}_{(\rho,\rho^{-1})}$ is built on top of the system BPA_{LR}, LR expressions along with the non-recursive terms of BPA_{LR} will be the choice of syntactical representation of processes in this paper.

Let VAR be a denumerably infinite set of variables $\{X, Y, Z \ldots\}$. $\mathcal{O}$ is the set of observables. The set of *non-recursive terms* T over $\mathcal{O}$ is defined by the following BNF:

$$T ::= \ a \mid X \mid T + T \mid T.T$$

where $a \in \mathcal{O}$ and $X \in VAR$. We may omit the concatenation operator '.' when it is obvious. We do not have a $\mathbf{0}$ in BPA_{LR}, and instead of a process $a.\mathbf{0}$ we simply have the process a. While in CCS we may only prefix a process expression by an observable a, here we have a full sequential composition instead, allowing us to sequentially compose any two terms in BPA_{LR}. The reader is referred to [6] for a deeper insight into the notion of full sequential composition.

The free variables $fv(T)$ of a term T are defined in the usual way. $fv(a) = \emptyset, fv(X) = \{X\}, fv(T_1.T_2) = fv(T_1+T_2) = fv(T_1) \cup fv(T_2)$. A term is *closed* if it has no free variables. We also write $T(\overline{X})$, where $\overline{X} = X_1X_2 \ldots X_n$ is the vector of free variables of T, to indicate that T may depend at most on $X_1, X_2, \ldots, X_n$. The substitution of free variables $T\{p_1/x_1\}\{p_2/x_2\} \ldots \{p_n/x_n\}$, where $\overline{p} = p_1p_2 \ldots p_n$ is the vector of non-recursive substitution terms, is also written as $T(\overline{p})$.

The free variables of the non-recursive term $T = abX + cY(X + Z)Y$ are X, Y, Z and we write $T(X, Y, Z)$ instead of T to indicate this fact. $T(a, b + Z, YY) = aba + c(b + Z)(a + YY)(b + Z)$ and $T(a, b, c) = aba + cb(a + c)b$ are obtained by the substitutions of the free variables X, Y, Z by $a, b+Z, YY$ and a, b, c respectively. $T(a, b, c)$ is a closed term.

Definition 6. *A non-recursive term T is* ***guarded*** *if every occurrence of a variable in T is preceded by some $a \in \mathcal{O}$. Formally:*

1. *$a \in \mathcal{O}$ is guarded*

2. *if T is guarded and T' is an arbitrary non-recursive term, then $T.T'$ is guarded*
3. *if T, T' are both guarded then so is $T + T'$*

Definition 7. *A non-recursive term T is said to be* ***linear*** *if all occurrences of variables are "at the end". Formally:*

1. *$X \in$ VAR is linear*
2. *Closed non-recursive terms are linear*
3. *if T, T' are both linear then so is $T + T'$*
4. *if T is a closed non-recursive term and T' is linear, then $T.T'$ is linear*

Definition 8. *A non-recursive term is* **strictly linear** *if it is of the form $\sum a_i + \sum b_j.X_j$.*

The terms $abX + cY(X+Z)Y, a, b+c(X+e)$ are guarded while $X + cY(X+Z)Y, a+Y, b+c(X+e)+X(Y+a)$ are not. The terms $X, a, a+Y, b+c(X+e), abX$ are linear while $cY(X+Z)Y, abYY, b+c(X+e)+X(Y+a)$ are not. The terms $a, a+bY$ and $a+b+cY+dZ$ are strictly linear while the linear terms $X, a+Y$ and $c(X+e)$ are not strictly linear.

Definition 9. *LR-expressions are syntactical constructs of the form $\langle X_1|E\rangle$ where $X_i \in$ VAR, $E = \{X_i = T_i(\overline{X}) \mid i = 1, \ldots, n\}$ is a set of recursion equations, $\overline{X}$ is the vector of variables $X_1, X_2 \ldots X_n$ and for every i,*

1. *$T_i(\overline{X})$ is guarded*
2. *$T_i(\overline{X})$ is linear*
3. *$T_i(\overline{X})$ may contain variables only from $\overline{X} = X_1, \ldots, X_n$.*

Definition 10. *An equation $X_i = T_i(\overline{X})$ is* ***superfluous*** *if X_i is not accessible from X_1 in the obvious sense. Superfluous equations in E may be omitted.*

Definition 11. ***Canonical LR-expressions*** *are LR-expressions where every $T_i(\overline{X})$ is strictly linear and E does not contain any superfluous equations.*

It is understood that LR expressions that differ only by a renaming of variables are identical. The semantics of the terms of BPA_{LR} are provided by a mapping $[\ ] : Ter(BPA_{LR}) \longrightarrow \mathbb{R}$ where $Ter(BPA_{LR})$ stands for the set of all terms of BPA_{LR}. This mapping is termed *intermediate semantics* in [6] and has been quoted in definition 12. In lemma 15, we provide a procedural mapping $[\![\ ]\!]$ restricted to canonical LR expressions that highlights their intuitive correspondence with process graphs. This is termed the *direct intermediate semantics* in [6]. The direct intermediate semantics will prove useful for the purpose of providing graphical intuition. However, the true semantics of a term in BPA_{LR} is given by the intermediate semantics $[\,]$. We proceed to establish the intuitive correspondence of canonical LR expressions with process graphs and refer the reader to the construction in lemma 15 for the precise correspondence.

Figure 5: The same set of equations with different root nodes

The LR expression $\langle X_1|E\rangle$ represents a process graph with X_1 as the root. The expression $\langle X_i|E\rangle$ represents the same graph but rooted at X_i instead. Note that in this case the rest of the graph might not be reachable anymore and some equations might become superfluous. In figure 5, we consider two canonical LR expressions $\langle X|X = a.Y+b, Y = c.X\rangle$ and $\langle Y|X = a.Y+b, Y = c.X\rangle$ which differ only in their root nodes. The nodes have been labelled X and Y for the purpose of naming and this convention should not be confused with the process chart notation [9], [6] where nodes decorated with variables have special semantics. $Y = c.X$ intuitively implies that there is an edge labelled c that starts at the node named Y and ends at the node named X. In a similar way $X = a.Y+b$ implies that there is an edge labelled a starting at X and ending at Y. In addition, the lone b in $X = a.Y + b$ implies that there is an outgoing edge labelled b that goes to a node that has not been named, or alternatively to a terminal node. In the LR expression $\langle X|X = a + bY, Y = cY + dX, Z = eZ\rangle$, $Z = eZ$ is superfluous because Z is not accessible from X. In the LR expression $\langle Z|X = a+bY, Y = cY+dX, Z = eZ\rangle$, $X = a + bY$ and $Y = cY + dX$ are superfluous.

We use the notation E_{-k} for the set of equations in E except the k^{th} equation, $E_{-k} = \{X_i = T_i(\overline{X}) \mid 1 \leq i \neq k \leq n\}$. This is useful for focusing attention on a particular equation. For example, if we want to focus on the k^{th} equation in $\langle X_1|E\rangle$, we may instead write this LR expression as $\langle X_1|X_k = T_k(\overline{X}), E_{-k}\rangle$. All the variables $X_1, X_2, \ldots X_n$ in $\langle X|E\rangle$, where $E = \{X_i = T_i(\overline{X}) \mid i = 1, \ldots, n\}$, are bound in $\langle X|E\rangle$. If a variable is bound in a term, it is not free and since $T_i(\overline{X})$ may contain variables only from $\overline{X} = X_1, \ldots, X_n$, every variable in a LR expression $\langle X|E\rangle$ is bound. Hence every LR expression $\langle X|E\rangle$ is a closed term.

Definition 12. *The intermediate semantics [6] is an inductively defined mapping* $[\,] : Ter(BPA_{LR}) \longrightarrow \mathbb{R}$ *of the terms of* BPA_{LR} *into the domain of finite process graphs*

1. $[a] = \;\rightarrow\circ\xrightarrow{a}\circ$
2. $[S + T] = [S] \oplus [T]$
3. $[S.T] = [S] \odot [T]$
4. $[\langle X|E\rangle] = \omega(c(t(p(\langle X|E\rangle))))$

where $\oplus$ *is the addition of process graphs,* $\odot$ *is their full sequential multiplication,* $\omega : \mathbb{G} \longrightarrow \mathbb{G}$ *is the* ***root-unwinding*** *operator* †*,* $c : \mathbb{G} \longrightarrow \mathbb{G}$ *is the* ***collaps*** *operator,* t *is the* ***tree*** *operator which defines the mapping of LR expressions into the domain of process trees* $\mathbb{T}$ *(note that* $\mathbb{T} \subseteq \mathbb{G}$*) and* p *is the* ***prefix*** *operator which makes a LR expression prefix free. The notation* $\rightarrow\circ_p\xrightarrow{a}\circ_q$ *is used to denote the process graph that has a root* p *which has a single edge labelled 'a' leading to a terminal node* q*.*

We will not concern ourselves with a detailed study of the mapping $[\,]$ and refer the reader to [6] for the definitions of root-unwinding, collaps, tree and prefix operators as well as the properties quoted in proposition 13. For the purpose of this paper, only the intermediate semantics of a and $a+b$ as given by $[a] = \;\rightarrow\circ\xrightarrow{a}\circ$ and $[a+b] = [a]\oplus[b]$ are important as they are required in the soundness proof. Without defining $\oplus$ in detail, we note that $\rightarrow\circ_{p_1}\xrightarrow{a}\circ_{q_1} \oplus \;\rightarrow\circ_{p_2}\xrightarrow{b}\circ_{q_2}$ is the process graph obtained by combining the roots p_1 and p_2 into a single node p which is the new root. In this process graph p will have two outgoing edges, one labelled a going to q_1 and another labelled b going to q_2.

Proposition 13. (Properties).

1. *LR expressions are always closed.*
2. $[\langle X|E_1\rangle.\langle Y|E_2\rangle] \sqsubseteq_{(\equiv,\equiv)} [\langle Z|E_3\rangle]$ *for some* Z *and* E_3*.*
3. $[\langle X|E_1\rangle + \langle Y|E_2\rangle] \sqsubseteq_{(\equiv,\equiv)} [\langle Z|E_3\rangle]$ *for some* Z *and* E_3*.*
4. $[T] \sqsubseteq_{(\equiv,\equiv)} [\langle Z|E\rangle]$ *for some* Z *and* E *if* T *is a closed non-recursive term.*
5. *For every LR expression there is a corresponding canonical LR expression, such that the intermediate semantical mappings of the two are strongly bisimilar.*

†In [6] the letter ρ is used in place of ω, however this conflicts with our convention of using ρ for relations on observables.

■

Proposition 13.4 states that a closed non-recursive term can also be represented by an appropriate LR expression. The proof of this is trivial, with $E = \{Z = T\}$. From proposition 13.2 and 13.3, we also know that expressions formed over LR expressions with the '+' and '.' operators also have an equivalent LR expression. As a consequence any process expressible in BPA_{LR} can be represented by an LR expression. Every LR expression, in turn, can be converted to an equivalent canonical form upto strong bisimilarity, as stated in proposition 13.5.

In order to highlight the correspondence between canonical LR expressions and process graphs we provide here two correspondences, $[\![\ ]\!]$ from canonical LR expressions to process graphs and $\|\ \|$ from process graphs to canonical LR expressions. These can be seen to have the interesting properties that each process graph g is strongly bisimilar to $[\![\|g\|]\!]$. Also, as pointed out in [6] $[\![\langle X|E\rangle]\!]$ is strongly bisimilar to $[\langle X|E\rangle]$.

Lemma 14. (LTS to canonical LR) *There is a mapping* $\|\ \|$ *from the domain of finite rooted LTS into the domain of canonical LR Expressions such that, for each finite rooted LTS* $\mathcal{L} = \langle Q, \Sigma, \longrightarrow, p_k\rangle$ *there is a canonical LR Expression* $\|\mathcal{L}\| \equiv \langle X_k|E\rangle$ *with* $[\langle X_k|E\rangle]$ *strongly bisimilar to* $\mathcal{L}$*.*

Proof. If the non-terminal states of $\mathcal{L}$ are named $p_1, p_2, \ldots, p_n$ for some n, we create a corresponding LR expression in the following manner.

1. Create and identify a variable X_i with each non-terminal state p_i.
2. Start with each T_i being empty.
3. If $p_i \xrightarrow{a} q$ where q is some terminal state, convert $X_i = T_i$ to $X_i = T_i + a$.
4. If $p_i \xrightarrow{a} p_j$ where p_j is some non-terminal state, convert $X_i = T_i$ to $X_i = T_i + a.X_j$.

The required canonical LR expression is $\langle X_k|E\rangle$ where $E = \{X_i = T_i(\overline{X}) \mid i = 1, \ldots, n\}$. It is not hard to see that $[\langle X_k|E\rangle]$ is strongly bisimilar to $\mathcal{L}$. The mapping $\|\ \|$ is defined through this construction, with $\|\mathcal{L}\| \equiv \langle X_k|E\rangle$. ■

Lemma 15. (Canonical LR to LTS)

There is a mapping $[\![\ ]\!]$ *from the domain of canonical LR expressions into the domain of finite rooted LTSs such that, for each canonical LR Expression* $\langle X_1|E\rangle$ *there is a finite rooted LTS* $\mathcal{L} \equiv [\![\langle X_1|E\rangle]\!]$ *with* $[\langle X_1|E\rangle]$ *strongly bisimilar to* $\mathcal{L}$*.*

Proof. Given a canonical LR expression $\langle X_1|E\rangle$ where $E =$

$\{X_i = T_i(\overline{X}) \mid i = 1, \ldots, n\}$ and $T_i = \sum_j a_{ij} + \sum_k b_{ik}.X_{ik}$, construct a rooted LTS $\mathcal{L}$ such that

1. The set of states of $\mathcal{L}$ is $\{X_i | i = 1, \ldots n\} \cup \{X_T\}$ where we introduce a variable X_T which stands for a terminal state.
2. If $T_i = \sum_j a_{ij} + \sum_k b_{ik}.X_{ik}$, then add the transitions $X_i \xrightarrow{a_{ij}} X_T$ and $X_i \xrightarrow{b_{ik}} X_{ik}$ for every j and k.
3. The set of observables in $\mathcal{L}$ can be taken to be the set of all observables appearing in $\langle X_1 | E \rangle$.
4. X_1 is the root of $\mathcal{L}$.

Clearly, $\mathcal{L}$ obtained here is finite. It is not hard to see that $[\langle X_1 | E \rangle]$ is strongly bisimilar to $\mathcal{L}$. The mapping $[\![\ ]\!]$ is defined through this construction, with $[\![\langle X_1 | E \rangle]\!] = \mathcal{L}$.

■

Lemma 16. (Semantics)

1. $[\![\langle X | E \rangle]\!] \sqsubseteq_{(\equiv,\equiv)} [\langle X | E \rangle]$
2. $[\![\| g \|]\!] \sqsubseteq_{(\equiv,\equiv)} g$

■

5 The Axiomatization $\mathcal{A}_{(\rho,\rho^{-1})}$

Using $\mathcal{A}_{(\rho,\sigma)}$ to denote a complete axiomatization for $\sqsubseteq_{(\rho,\sigma)}$, we present in this section the axiomatization $\mathcal{A}_{(\rho,\rho^{-1})}$ where ρ is assumed to be a preorder. The axiomatization is proven complete for $\sqsubseteq_{(\rho,\rho^{-1})}$ over finite process graphs $\mathbb{R}$ relative to a complete axiomatization of ρ. A necessary condition for soundness is that $\mathcal{O}$ does not contain empty observables. In this case, $\sqsubseteq_{(\rho,\rho^{-1})}$ also turns out to be a congruence, allowing us to give a more succinct formulation of the axiomatization. It should be noted that all equivalence relations in the class of (ρ, σ)-bisimilarities have $\sigma = \rho^{-1}$ and ρ as a preorder. Thus $\mathcal{A}_{(\rho,\rho^{-1})}$ is in fact a complete axiomatization of all such equivalences defined over finite process graphs $\mathbb{R}$ with no empty observables.

Since ρ is a preorder, from proposition 2 we know that $\sqsubseteq_{(\rho,\rho^{-1})}$ contains $\sqsubseteq_{(\equiv,\equiv)}$. Taking $\mathcal{O} = \mathit{Act}$, strong bisimilarity has been shown to be equivalent to the natural bisimilarity relation $\sqsubseteq_{(\equiv,\equiv)}$ in [2]. We therefore refer to $\sqsubseteq_{(\equiv,\equiv)}$ as strong bisimilarity as well. The axiomatizations we consider are for relations no finer than strong bisimilarity. For this reason, we build our axiomatization upon the axiomatization BPA_{LR} shown in figure 6 which is a complete axiomatization for strong bisimilarity (see [6]).

It should be noted that the equality $=$ in BPA_{LR} is the relation on terms of BPA_{LR} that corresponds to strong bisimilarity $\sqsubseteq_{(\equiv,\equiv)}$

$x + y = y + x$		$A1$
$(x + y) + z = x + (y + z)$		$A2$
$x + x = x$		$A3$
$(x + y)z = xz + yz$		$A4$
$(xy)z = x(yz)$		$A5$
$\dfrac{p_i = \langle X_i \mid E \rangle,\ i = 1, \ldots, n}{p_1 = T_1(\overline{p})}$		$R1$
$\dfrac{p_i = T_i(\overline{p}),\ i = 1, \ldots, n}{p_1 = \langle X_1 \mid E \rangle}$	$T_i(\overline{X})$ is guarded	$R2$

Figure 6: BPA_{LR} - a proof system for $\sqsubseteq_{(\equiv,\equiv)}$

on process graphs. Similarly $=_\rho$ will stand for the relation on terms of $\mathcal{A}_{(\rho,\rho^{-1})}$ that corresponds to $\sqsubseteq_{(\rho,\rho^{-1})}$ on process graphs.

We briefly present the outline of one of the ways of building $\mathcal{A}_{(\rho,\rho^{-1})}$ on top of BPA_{LR} before moving on to our preferred formulation. The purpose of presenting this formulation is only to provide an alternative structuring of the axiomatization which might prove easier to extend. In particular, we believe that axiomatizations of other relations that are not congruences might have to proceed in this manner as this formulation does not rely on the congruence of $\sqsubseteq_{(\rho,\rho^{-1})}$.

The first formulation provides an axiom that cannot work in contexts, and is applicable only for canonical LR expressions. Every closed term of BPA_{LR}, whether recursive, non-recursive or an expression formed over these with the operators '.' and '+', can be proven to be strongly bisimilar to a canonical LR expression within the framework of BPA_{LR}. This boils down to proving parts 2,3,4 and 5 of proposition 13 within the scope of the BPA_{LR}, the proof of which is provided in [6]. Since strong bisimilarity will be the finest relation on processes considered here except for identity on processes, we may restrict our attention to the quotient structure $\mathbb{R}/\sqsubseteq_{(\equiv,\equiv)}$. Instead of needing to deal with closed terms of arbitrary forms, this allows us to deal with equivalent canonical LR expressions which have a very convenient form. At the same time, it allows us to add duplicate edges between two nodes while staying in the same equivalence class. On the quotient structure $\mathbb{R}/\sqsubseteq_{(\equiv,\equiv)}$, strong bisimilarity reduces to identity $\equiv$ on the equivalence classes. In this formulation, the full axiomatization would be composed of two "layers", BPA_{LR} forming the inner part and a second "layer" consisting of axioms that work on the quotient structure $\mathbb{R}/\sqsubseteq_{(\equiv,\equiv)}$. It turns out that we only need one more axiom (schema), the vertical axiom AV shown below.

$$\begin{array}{c} \langle X_1 | X_k = b.X_j + T, E_{-k} \rangle \\ =_\rho \\ \langle X_1 | X_k = a.X_j + b.X_j + T, E_{-k} \rangle \end{array} \qquad \textit{if } a \rho b$$

This axiom schema comfortably assumes each term in $\mathbb{R}/\underline{\square}_{(\equiv,\equiv)}$ to be a canonical LR expression (provable within the BPA_{LR} part) and therefore does not need to work in a context. This is particularly useful if $\underline{\square}_{(\rho,\rho^{-1})}$ was not a congruence. However, one of the conditions for the soundness of this axiom is not having empty observables in $\mathcal{O}$ which also ends up proving $\underline{\square}_{(\rho,\rho^{-1})}$ to be a congruence. For this reason we will prefer the second formulation which is more succinct.

In our preferred approach, we first translate BPA_{LR} to $BPA_{\rho LR}$ with terms of the two systems being identical. The translation of axioms of BPA_{LR} to $BPA_{\rho LR}$ involves the replacement of all $=$ by $=_\rho$, where $=_\rho$ on terms of $BPA_{\rho LR}$ corresponds to the relation $\underline{\square}_{(\rho,\rho^{-1})}$ on process graphs. In this manner we get the axioms $A1_\rho - A5_\rho$ and $R1_\rho, R2_\rho$ listed in figure 7. Note that the recursive equations E in a LR expression $\langle X|E\rangle$ remain of the form $\{X_i = T_i(\overline{X}) | i = 1, \ldots, n\}$ as the '=' in question here is a syntactical notation for specifying recursive equations and is not strong bisimilarity.

From the completeness of BPA_{LR}, we know that for all T, S in $Ter(BPA_{LR})$, $[T]\underline{\square}_{(\equiv,\equiv)}[S] \Longrightarrow BPA_{LR} \vdash T = S$. Taking the proof of $T = S$ in BPA_{LR} and replacing each application of the axioms of BPA_{LR} by the corresponding axioms of $BPA_{\rho LR}$, we end up constructing a proof of $T =_\rho S$ in $BPA_{\rho LR}$. That is, within $BPA_{\rho LR}$, we can prove the (ρ, ρ^{-1})-bisimilarity of every pair of strongly bisimilar processes. Using essentially the same arguments every closed term of $BPA_{\rho LR}$ can be proven to be (ρ, ρ^{-1})-bisimilar to a canonical LR expression within the framework of $BPA_{\rho LR}$.

In a similar manner, each instance of the *substitution* inference rule shown below is provable in $BPA_{\rho LR}$ by a translation of the corresponding proof [6] in BPA_{LR}.

$$\frac{T_k(\overline{X}) =_\rho T'_k(\overline{X})}{\langle X_1 | X_k = T_k(\overline{X}), E_{-k} \rangle =_\rho \langle X_1 | X_k = T'_k(\overline{X}), E_{-k} \rangle}$$

We introduce the axiom schema *Axiom-Vertical* AV_ρ (which is named so for reasons that will become clear in the proof of completeness) to obtain the axiomatization $\mathcal{A}_{(\rho,\rho^{-1})}$ in figure 7. The terms of $\mathcal{A}_{(\rho,\rho^{-1})}$ are the same as those of BPA_{LR}. The intuition of the axiom AV_ρ is similar to that of $A3_\rho$ which can prove that (with $x = b$) strong bisimilarity is insensitive to the addition of duplicate edges. AV_ρ says that $\underline{\square}_{(\rho,\rho^{-1})}$ is insensitive to the addition of edges that are

$b =_\rho a + b$ if $a\rho b$		AV_ρ
$x + y =_\rho y + x$		$A1_\rho$
$(x + y) + z =_\rho x + (y + z)$		$A2_\rho$
$x + x =_\rho x$		$A3_\rho$
$(x + y)z =_\rho xz + yz$		$A4_\rho$
$(xy)z =_\rho x(yz)$		$A5_\rho$
$\dfrac{p_i =_\rho \langle X_i \vert E\rangle,\ i = 1, \ldots, n}{p_1 =_\rho T_1(\overline{p})}$		$R1_\rho$
$\dfrac{p_i =_\rho T_i(\overline{p}),\ i = 1, \ldots, n}{p_1 =_\rho \langle X_1 \vert E\rangle}$	$T_i(\overline{X})$ is guarded	$R2_\rho$

Figure 7: $\mathcal{A}_{(\rho,\rho^{-1})}$

related by ρ to the existing ones.

In the case that ρ is also an equivalence relation, $a\rho b \Longrightarrow [b =_\rho a + b \wedge a =_\rho a + b] \Longrightarrow a =_\rho b$, which says that $=_\rho$ for such a ρ is indifferent to observables that lie in the same equivalence class with respect to ρ. Thus proving (ρ, ρ^{-1})-bisimilarity of graphs for such a ρ can be reduced to proving strong bisimilarity of graphs with labels from $\mathcal{O}/\rho$. This case not being so interesting, we shall only focus on the case where ρ is a preorder.

What should be easy to see is that all instances of AV as shown below, can be proven within the framework of this axiomatization.

$$\begin{array}{c} \langle X_1 | X_k = b.X_j + T, E_{-k}\rangle \\ =_\rho \\ \langle X_1 | X_k = a.X_j + b.X_j + T, E_{-k}\rangle \end{array} \qquad \textit{if } a\rho b$$

From $b =_\rho a+b \stackrel{A4_\rho}{\Longrightarrow} bX_j =_\rho aX_j + bX_j$, and using the congruence property of $=_\rho$, for any T we can prove that $b.X_j + T =_\rho a.X_j + b.X_j + T$. From the substitution rule, with $T_j(\overline{X}) \equiv b.X_j + T$ and $T'_j(\overline{X}) \equiv a.X_j + b.X_j + T$, we get the required result.

With the result that each LR expression can be proven to be equivalent to a canonical LR expression upto (ρ, ρ^{-1}) bisimilarity within the scope of $BPA_{\rho LR}$, we could have alternatively chosen the above inference rule instead of the axiom AV_ρ. The above inference rule will also be useful in providing an intuitive understanding of the procedure of saturation. In figure 8, $g \equiv \|\langle X_1 | X_k = b.X_j + T, E_{-k}\rangle\|$ and $g' \equiv \|\langle X_1 | X_k = a.X_j + b.X_j + T, E_{-k}\rangle\|$ and the nodes p and p' correspond to X_k and X_j respectively. It is not hard to see that g and g' are in fact related by $\sqsubseteq_{(\equiv,\rho^{-1})}$, and therefore also by $\sqsubseteq_{(\rho,\rho^{-1})}$.

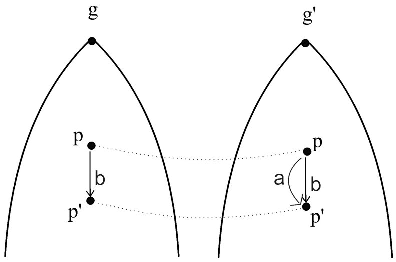

Figure 8: Graphical intuition for AV

The rest of the paper is devoted to the proof of soundness and completeness of the axiomatization. In section 7, we introduce the construction of derivatives of process graphs. The construction revolves around adding edges in the same way as the a edge is added in figure 8 to obtain g' from g.

It should be noted that the case of empty observables could lead to peculiar situations. For example, if a is an empty observable, then adding an a edge between p and p' has the effect of identifying p and p' and is not the addition of edges in any traditional sense. In fact, such addition corresponds to the addition of the term $a.X_j$ to X_k. However, the prefixing by an empty observable is undefined. This problem is summarized in definition 3 and part 3 of lemma 4. The presence of empty observables would force us to change the definition of non-recursive terms of BPA_{LR} to exclude empty observables, modify the mappings $[\ ]$ and $[\![\ ]\!]$ to introduce empty loops at each node and restrict AV_ρ to work only for non empty observables in order to be sound. Not being able to add a particular observable leads to the failure of the completeness proof.

6 Soundness of $\mathcal{A}_{(\rho,\rho^{-1})}$

Theorem 17. (Soundness) *For all $T, S \in Ter(\mathcal{A}_{(\rho,\rho^{-1})})$ in the absence of empty observables in $\mathcal{O}$,*

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash T \ =_\rho S \implies [T] \sqsubseteq_{(\rho,\rho^{-1})} [S]$$

PROOF. In order to prove the soundness of the axiom AV_ρ we need to prove $[b] \ \sqsubseteq_{(\rho,\rho^{-1})} \ [a+b]$ for any $a, b \in \mathcal{O}$ such that $a\rho b$.

From definition 12 we can see that the process graphs $[b]$ and $[a+b]$ are the ones given in figure 9. It is easily seen that $R : [b] \sqsubseteq_{(\rho,\rho^{-1})} [a+b]$ where $R = \{(p_0, q_0), (p_1, q_1), (p_1, q_2)\}$. Thus, AV_ρ is sound.

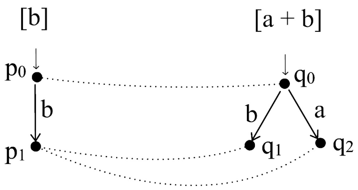

Figure 9: AV_ρ

The rest of the proof follows along the lines of the soundness proof in [6] with the observation that $\underline{\Box}_{(\rho,\rho^{-1})}$ is a congruence and is preserved by the *prefix, tree, collaps* and *root-unwinding* operators employed in definition 12. ■

7 Saturation of Process Graphs

The method of proving completeness of $\mathcal{A}_{(\rho,\rho^{-1})}$ is inspired by the completeness proof of $BPA_{\tau LR}$ [6]. In [6], Bergstra and Klop define the operators Δ and E on process graphs which preserve observational congruence while reducing any two graphs related by observational congruence to strongly bisimilar graphs.

$$\begin{array}{lcl} g & \approx^+ & h \\ \approx^+ & & \approx^+ \\ \Delta(g) & \approx^+ & \Delta(h) \\ \approx^+ & & \approx^+ \\ E(\Delta(g)) & \sim & E(\Delta(h)) \end{array}$$

Here, $\approx^+$ is observational congruence and $\sim$ is strong bisimilarity. The relation $g \approx^+ \Delta(g)$ is shown in a vertical fashion here. The proof of completeness of $BPA_{\tau LR}$ hinges upon such a diagram where $E(\Delta(g))$ and $E(\Delta(h))$ are strongly bisimilar if and only if g and h are observationally congruent.

Following a similar strategy of saturation, in this section we characterize certain relations between derivatives of graphs[‡] related by (ρ, σ)-bisimilarities. In particular, if $g \;\underline{\Box}_{(\rho,\rho^{-1})} h$ and ρ is a preorder, we are able to prove the existence of derivatives g' and h' such that $g \;\underline{\Box}_{(\equiv,\rho^{-1})} g'$, $h \;\underline{\Box}_{(\equiv,\rho^{-1})} h'$ and $g' \;\underline{\Box}_{(\equiv,\equiv)} h'$. Diagrammatically,

$$\begin{array}{lcl} g & \underline{\Box}_{(\rho,\rho^{-1})} & h \\ \underline{\Box}_{(\equiv,\rho^{-1})} & & \underline{\Box}_{(\equiv,\rho^{-1})} \\ g' & \underline{\Box}_{(\equiv,\equiv)} & h' \end{array}$$

[‡] A process graph derived from another in some manner.

This result is critical for the proof of completeness of the axiomatization $\mathcal{A}_{(\rho,\rho^{-1})}$ and this entire section deals with proving the existence of such g' and h'. Roughly speaking, the transformation from g and h to g' and h' requires that all transitions in g and h that are matched by the relations ρ and σ get matched in g' and h' by $\equiv$ also. We start by constructing the derivatives $g[\pi_g]$ and $h[\pi_g]$ such that the transition $\pi_g \in \mathit{trans}(g)$ gets matched by $\equiv$ in $h[\pi_g]$. Also, any newly added transitions in $g[\pi_g]$ and $h[\pi_g]$ also get matched by $\equiv$ in $h[\pi_g]$ and $g[\pi_g]$ respectively. In this manner, the derivative pair $g[\pi_g], h[\pi_g]$ has fewer transitions that cannot be matched by $\equiv$ as compared to the pair g, h. This result is summed up in the partial saturation lemma 24. Performing this construction for all transitions of g and h we are able to get the required g' and h', resulting in the saturation lemma 27 and theorem 30.

From lemma 2.8, a (ρ, σ)-bisimilarity is an equivalence relation if and only if $\sigma = \rho^{-1}$ and ρ is a preorder. While we intend to tackle only equivalence relations and this condition is sufficient for all results proved in this section, we will continue to make only the minimal necessary assumptions in each lemma.

7.1 Partial Saturation

Let $R \subseteq \mathit{nodes}(g) \times \mathit{nodes}(h)$ be a binary relation between the nodes of process graphs g and h and let ρ and σ be binary relations on the set $\mathcal{O}$ of observables.

We write $Q(\pi_g)$ for the **h-image** of $\pi_g \in \mathit{trans}(g)$ and $P(\pi_h)$ for the **g-image** of $\pi_h \in \mathit{trans}(h)$, defining them as:

$$Q(p \xrightarrow{c} p') = \{q \xrightarrow{b} q' \in \mathit{trans}(h) \mid pRq \ \wedge \ c\rho b \ \wedge p'Rq'\}$$

$$P(q \xrightarrow{b} q') = \{p \xrightarrow{c} p' \in \mathit{trans}(g) \mid pRq \ \wedge \ c\sigma b \ \wedge p'Rq'\}$$

For sets of transitions $G \subseteq \mathit{trans}(g)$ and $H \subseteq \mathit{trans}(h)$, we write $\mathcal{Q}(G)$ and $\mathcal{P}(H)$ for the h-image of G and g-image of H, respectively.

$$\mathcal{Q}(G) = \bigcup_{p \xrightarrow{c} p' \in G} Q(p \xrightarrow{c} p')$$

$$\mathcal{P}(H) = \bigcup_{q \xrightarrow{b} q' \in H} P(q \xrightarrow{b} q')$$

We write $T_i(\pi_g)$ and $U_i(\pi_g)$ for the sets of **level i matches** of $\pi_g \in \mathit{trans}(g)$ in g and h respectively.

$$T_o(\pi_g) = \{\pi_g\}$$

$$T_{i+1}(\pi_g) = \mathcal{P}(U_i(\pi_g)) \qquad U_i(\pi_g) = \mathcal{Q}(T_i(\pi_g))$$

Note that $U_0(\pi_g) = \mathcal{Q}(T_0(\pi_g))$. $T(\pi_g)$ and $U(\pi_g)$, the sets of **closure matches** of π_g in g and h, can be defined as infinite unions of T_i and U_i. Intuitively, $T(p_o \xrightarrow{a} p'_o)$ and $Q(p_o \xrightarrow{a} p'_o)$ are the sets of transitions where an extra $\xrightarrow{a}$ edge must be added for the purpose of partial saturation with respect to $p_o \xrightarrow{a} p'_o$.

$$T(\pi_g) = \bigcup_{i\geq 0} T_i(\pi_g) \qquad \text{and} \qquad U(\pi_g) = \bigcup_{i\geq 0} U_i(\pi_G)$$

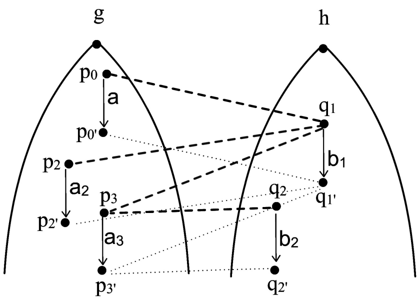

Figure 10: Construction of $T(\pi)$ and $U(\pi)$

In figure 10, with $\pi = p_0 \xrightarrow{a} p'_0$, we depict a case where $T_0(\pi) = \{p_0 \xrightarrow{a} p'_0\}$, $U_0(\pi) = \{q_1 \xrightarrow{b_1} q'_1\}$, $T_1(\pi) = \{p_2 \xrightarrow{a_2} p'_2,\ p_3 \xrightarrow{a_3} p'_3\}$, $U_1(\pi) = \{q_2 \xrightarrow{b_2} q'_2\}$ and $\forall i \geq 2 : T_i(\pi) = U_i(\pi) = \emptyset$. Thus, $T(\pi) = \{p_0 \xrightarrow{a} p'_0,\ p_2 \xrightarrow{a_2} p'_2,\ p_3 \xrightarrow{a_3} p'_3\}$ and $U(\pi) = \{q_1 \xrightarrow{b_1} q'_1,\ q_2 \xrightarrow{b_2} p'_2\}$. Here R contains (p_0, q_1), (p'_0, q'_1), (p_2, q_1), (p'_2, q'_1), (p_3, q_1), (p'_3, q'_1), (p_3, q_2), (p'_3, q'_2). Also, $(a, b_1), (a_3, b_2) \in \rho$ and $(a_2, b_1), (a_3, b_1) \in \sigma$.

Definition 18. *Given a relation $R \subseteq nodes(g) \times nodes(h)$ on two process graphs $g, h \in \mathbb{G}$, a pair of relations on observables ρ, σ and a transition $\pi_g = p_o \xrightarrow{a} p'_o \in trans(g)$ we define the* ***partial saturation of*** g ***and*** h ***with respect to*** π_g*, written as $g[\pi_g]$ and $h[\pi_g]$, as the process graphs that have the same set of nodes and root as g and h respectively, while the set of transitions are defined as follows*

1. $trans(g[\pi_g]) = trans(g) \cup \{p \xrightarrow{a} p' \mid p \xrightarrow{c} p' \in T(\pi_g), c \in \mathcal{O}\}$
2. $trans(h[\pi_g]) = trans(h) \cup \{q \xrightarrow{a} q' \mid q \xrightarrow{b} q' \in U(\pi_g), b \in \mathcal{O}\}$

A better intuition might be obtained by viewing this as a construction of $g[\pi_g]$ and $h[\pi_g]$ from g and h. We can refer to $g[\pi_g]$ as g' and $h[\pi_g]$ as h' as the transition in question $\pi_g = p_o \xrightarrow{a} p'_o$ is clear. Start with $g' := g$ and $h' := h$. The following steps correspond to adding the subgraph $a.p'$ at the node p if some transition $p \xrightarrow{c} p'$ is a closure match of π_g in g, and $a.q'$ at the node q if some transition $q \xrightarrow{b} q'$ is a closure match of π_g in h.

1. $\forall p \xrightarrow{c} p' \in T(p_o \xrightarrow{a} p'_o)$, add $p \xrightarrow{a} p'$ in g'
2. $\forall q \xrightarrow{b} q' \in U(p_o \xrightarrow{a} p'_o)$, add $q \xrightarrow{a} q'$ in h'

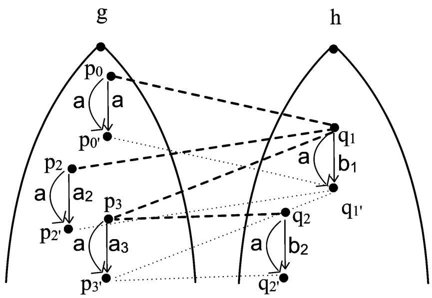

Figure 11: Construction of $g[p_0 \xrightarrow{a} p'_0]$ and $h[p_0 \xrightarrow{a} p'_0]$

Following the partial saturation process for the example of figure 10, we show the construction of $g[p_0 \xrightarrow{a} p'_0]$ and $h[p_0 \xrightarrow{a} p'_0]$ in figure 11. The axiom AV_ρ justifies each individual addition of this kind and we can break this construction into many single transition additions. However, in the presence of empty observables, it must be noted that it might not be possible to add an edge labelled by an arbitrary observable (see lemma 4 part 3). The inability to add edges at will, results in the failure of the various saturation results in this section.

Definition 19. *If $v \in \mathcal{O}$ and $g \in \mathbb{G}$, we define the v-**application of a transition** $p \xrightarrow{w} p'$ **to the graph** g, written $g_v\langle p \xrightarrow{w} p'\rangle$, as the graph obtained by adding the transition $p \xrightarrow{v} p'$ to g.*

We extend the definition to a set of transitions by the inductive definition $g_v\langle\{\pi\}\rangle = g_v\langle\pi\rangle$ and $g_v\langle G \cup \{\pi\}\rangle = g_v\langle\pi\rangle\langle G\rangle$.

Lemma 20. *If ρ is a preorder, $h \in \mathbb{G}, a \in \mathcal{O}, q \xrightarrow{b} q' \in trans(h)$ and $a\rho b$, then*

$$id_h : h \sqsubseteq_{(\equiv,\rho^{-1})} h_a\langle q \xrightarrow{b} q'\rangle$$

∎

Lemma 21.
If $g, h \in \mathbb{G}, R : g \underline{\square}_{(\rho,\rho^{-1})} h$ *and* $\pi = p_o \xrightarrow{a} p'_o \in trans(g)$ *then*

1. $g_a\langle T(\pi)\rangle \equiv g[\pi]$
2. $h_a\langle U(\pi)\rangle \equiv h[\pi]$

■

Lemma 22. *The following properties of* $\mathcal{Q}$ *and* $\mathcal{P}$ *follow trivially from their definitions*

1. $\forall q \xrightarrow{b} q' \in \mathcal{Q}(T) : \ \exists p \xrightarrow{c} p' \in T : \ c\rho b$
2. $\forall p \xrightarrow{c} p' \in \mathcal{P}(U) : \ \exists q \xrightarrow{b} q' \in U : \ b\sigma^{-1}c$

■

Lemma 23. *If* $\rho \circ \sigma^{-1} \subseteq \rho$, $\rho \circ \rho \subseteq \rho$ *(*ρ *is transitive) and* $\equiv \subseteq \rho$ *(*ρ *is reflexive) then* $\forall i \geq 0$,

1. $\forall\ p \xrightarrow{c} p' \in T_i(p_o \xrightarrow{a} p'_o) : a\rho c$
2. $\forall\ q \xrightarrow{b} q' \in U_i(p_o \xrightarrow{a} p'_o) : a\rho b$

■

Lemma 23 says that, under the stated assumptions, the labels of all closure matches of a transition $p_o \xrightarrow{a} p'_o \in trans(g)$ are related to a through ρ. This can be proven by simultaneous induction on i.

Note that we will be dealing only with $\mathbb{R}$ in further sections although some results in this section are valid for $\mathbb{G}$. Therefore in sections 7.2 and 8, $trans(g)$ and $trans(h)$ are finite and hence $T(\pi_g) \subseteq trans(g)$ and $U(\pi_g) \subseteq trans(h)$ are also finite. Thus the partial saturations $g[\pi_g]$ and $h[\pi_g]$ are also finite process graphs.

7.2 Properties of Saturation

Lemma 24. (Partial Saturation) *For finite process graphs* $g, h \in \mathbb{R}$ *and a relation* R *such that* $R : g\underline{\square}_{(\rho,\sigma)} h$ *where* ρ, σ *are such that* $\equiv \subseteq \rho, \sigma$ *(both are reflexive),* $\rho \circ \rho \subseteq \rho$ *(*ρ *is transitive) and* $\rho \circ \sigma^{-1} \subseteq \rho$*, the following diagram holds for all* $\pi_g = p_o \xrightarrow{a} p'_o \in trans(g)$.

$$\begin{array}{ccc} g & \underline{\square}_{(\rho,\sigma)} & h \\ \underline{\square}_{(\equiv,\rho^{-1})} & & \underline{\square}_{(\equiv,\rho^{-1})} \\ g[\pi_g] & \underline{\square}_{(\rho,\sigma)} & h[\pi_g] \end{array}$$

Moreover,

1. $id_g : \ g\ \underline{\square}_{(\equiv,\rho^{-1})} g[\pi_g]$
2. $id_h : \ h\ \underline{\square}_{(\equiv,\rho^{-1})} h[\pi_g]$
3. $R : \ g[\pi_g]\ \underline{\square}_{(\rho,\sigma)}\ h[\pi_g]$

4. (a) *For all* q *in* h' *such that* $(p_o, q) \in R$*,* $\exists q' : q \xrightarrow{a} q' \in h' \wedge (p'_o, q') \in R$

(b) *For all transitions* $p \xrightarrow{a} p'$ *in* g' *that are newly added* [§] *and for all* q *in* h' *such that* $(p, q) \in R$*,* $\exists q' : q \xrightarrow{a} q' \in h' \wedge (p', q') \in R$

(c) *For all transitions* $q \xrightarrow{a} q'$ *in* h' *that are newly added and for all* p *in* g' *such that* $(p, q) \in R$*,* $\exists p' : p \xrightarrow{a} p' \in g' \wedge (p', q') \in R$

PROOF. The proofs of parts 1 and 2 follow from lemma 20 and 21. If $U(\pi_g) = \{\pi_1, \ldots, \pi_n\}$ then from lemma 20 we get

$$h \sqsubseteq_{(\equiv,\rho^{-1})} h_a\langle\pi_1\rangle \sqsubseteq_{(\equiv,\rho^{-1})} \cdots \sqsubseteq_{(\equiv,\rho^{-1})} h_a\langle\pi_1\rangle \ldots \langle\pi_n\rangle$$

From the transitivity of $\sqsubseteq_{(\equiv,\rho^{-1})}$, the observation that by definition $h_a\langle\pi_1\rangle \ldots \langle\pi_n\rangle \equiv h_a\langle U(\pi)\rangle$, and from lemma 21, we may show $id_h : h \sqsubseteq_{(\equiv,\rho^{-1})} h[\pi_g]$. In a similar manner we prove part 1. Part 3 can be seen by noting $id_g^{-1} \circ R \circ id_h : g' \sqsubseteq_{(\rho,\sigma)} h'$ and $id_g^{-1} \circ R \circ id_h = R$.

To prove part 4.4c, consider the newly added transition $\pi_{h'} = q \xrightarrow{a} q'$ in $h' = h[\pi_g]$. Since $\pi_{h'}$ is newly added, it has been added by an a-application of some $q \xrightarrow{b} q' \in U(\pi_g)$ since $h[\pi_g] = h_a\langle U(\pi_g)\rangle$. Since $q \xrightarrow{b} q' \in U_i(\pi_g)$ for some i, we get $a\rho b$ by lemma 23. By the fact that $R : g \sqsubseteq_{(\rho,\sigma)} h$, for any p such that $(p, q) \in R$ we know that $\exists p', c[p \xrightarrow{c} p' \in g \wedge c\rho b \wedge (p', q') \in R]$. From the definition of T_{i+1}, we know that $p \xrightarrow{c} p' \in T_{i+1}(\pi_g) \subseteq T(\pi_g)$. The fact that g' is obtained from g by the a-application of $T(\pi_g)$ gives us $p \xrightarrow{a} p' \in g'$ as required. This proves part 4.4c. The part 4.4b is proven on symmetric lines.

For part 4.4a, notice that from $R : g \sqsubseteq_{(\rho,\sigma)} h$ and $(p_o, q) \in R$ we know that $\exists q', b[q \xrightarrow{b} q' \in g \wedge a\rho b \wedge (p'_o, q') \in R]$. On similar lines as the proof for part 4.4c, note that $q \xrightarrow{b} q' \in U_i(\pi_g)$ for some i. Thus the graph h' which is obtained by an a-application of $U(\pi_g)$ also contains the transition $q \xrightarrow{a} q'$ as required.

■

The construction of $g[p_0 \xrightarrow{a} p'_0]$ and $h[p_0 \xrightarrow{a} p'_0]$ is intended to make g and h closer in terms of bisimilarity. For the transition $p_0 \xrightarrow{a} p'_0$ in g, we add an edge $\xrightarrow{a}$ at the h matches of $p_0 \xrightarrow{a} p'_0$ to make sure that at least the transition $p_0 \xrightarrow{a} p'_0$ can be matched by the identity relation $\equiv$ on observables. This is captured by lemma 24.4a. These additions correspond to the additions due to U_0. For the example considered in figure 10 in section 7.1, we get the figure

[§] Newly added transitions in g' and h' are $trans(g') - trans(g)$ and $trans(h') - trans(h)$ respectively.

12 upon the additions due to U_0.

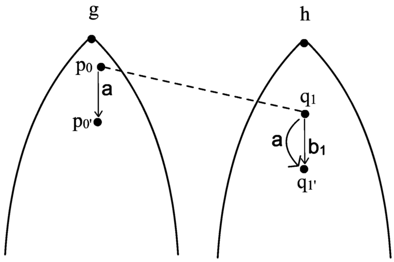

Figure 12: Additions due to U_0

This addition triggers off the need to add corresponding transitions (as given by T_1) in g to ensure that the newly added transitions in h also get matched in g by the identity on observables.

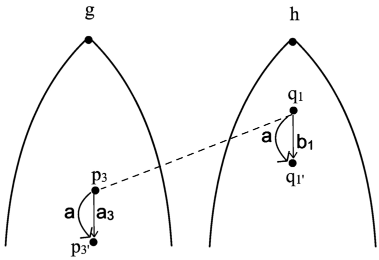

Figure 13: An addition due to T_1

In figure 13 we show one of the additions due to T_1. This addition, in turn, further triggers the need for additions due to U_1 as shown in figure 14.

Proceeding in this manner, in the end the transition $p_0 \xrightarrow{a} p_0'$ and all the newly added transitions end up being matched by $\equiv$ (identity on observables) in figure 11. This is captured by lemma 24.4b, 24.4c. Repeating the procedure for each transition $\pi \in g$ we end up with a g' and h' where each transition in g', old or new, is matched by transitions in h by $\equiv$. In this manner we can reduce $\sqsubseteq_{(\rho,\sigma)}$ to $\sqsubseteq_{(\equiv,\sigma)}$.

Definition 25. *Given a relation R, a pair of relations ρ, σ and process graphs $g, h \in \mathbb{R}$ such that $R : g \sqsubseteq_{(\rho,\sigma)} h$, we define the* ***partial saturation of*** g ***and*** h ***with respect to the set*** $G \subseteq trans(g)$*, written as $g[G]$ and $h[G]$, as follows.*

1. $g[\emptyset] = g, h[\emptyset] = h$

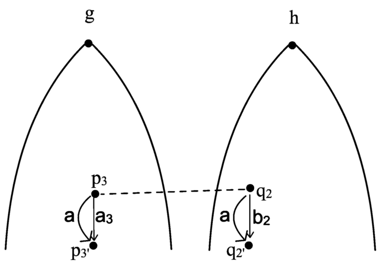

Figure 14: An addition due to U_1

2. $g[G \cup \{\pi_g\}] = (g[G])[\pi_g], h[G \cup \{\pi_g\}] = (h[G])[\pi_g]$

Lemma 26. (Semi Saturation) *For finite process graphs $g, h \in \mathbb{R}$ and a relation R such that $R : g \underline{\Box}_{(\rho,\sigma)} h$ where ρ, σ are such that $\equiv \subseteq \rho, \sigma$ (both are reflexive), $\rho \circ \rho \subseteq \rho$ (ρ is transitive) and $\rho \circ \sigma^{-1} \subseteq \rho$, the following diagram holds.*

$$\begin{array}{ccc} g & \underline{\Box}_{(\rho,\sigma)} & h \\ \underline{\Box}_{(\equiv,\rho^{-1})} & & \underline{\Box}_{(\equiv,\rho^{-1})} \\ g' & \underline{\Box}_{(\equiv,\sigma)} & h' \end{array}$$

where $g' = g[trans(g)]$ and $h' = h[trans(g)]$. The following hold as well.

1. $id_g : \ g \ \underline{\Box}_{(\equiv,\rho^{-1})} g'$
2. $id_h : \ h \ \underline{\Box}_{(\equiv,\rho^{-1})} h'$
3. $R : \ g' \ \underline{\Box}_{(\equiv,\sigma)} \ h'$

Proof. The key to the proof of the semi saturation lemma is the "tiling" of the partial saturation diagram. From the partial saturation lemma 24, if $R : g[G_k] \ \underline{\Box}_{(\rho,\sigma)} h[G_k]$ for some $G_k \subseteq trans(g)$ then for $\pi_k \in trans(g)$ we get the following diagram

$$\begin{array}{ccc} g[G_k] & \underline{\Box}_{(\rho,\sigma)} & h[G_k] \\ \underline{\Box}_{(\equiv,\rho^{-1})} & & \underline{\Box}_{(\equiv,\rho^{-1})} \\ g[G_k][\pi_k] & \underline{\Box}_{(\equiv,\sigma)} & h[G_k][\pi_k] \end{array}$$

It helps to note that $trans(g) \subseteq trans(g[G_k])$ and therefore $\pi_k \in trans(g[G_k])$ as well and the above diagram comes directly from partial saturation lemma. Enumerating $trans(g) = \{\pi_1, \pi_2, \ldots, \pi_n\}$, we define $G_k = \{\pi_i \in trans(g) | i < k\}$. It should be obvious that $G_1 = \emptyset, G_{k+1} = G_k \cup \{\pi_k\}$ and $i < j \implies G_i \subset G_j$. Since $G_1 = \emptyset$, we get $g[G_1] \equiv g$ and $h[G_1] \equiv h$. Therefore, from the knowledge that $R : g \sqsubseteq_{(\rho,\sigma)} h$, the above diagram holds for $k = 1$. Note that $g[G_k][\pi_k] = g[G_k \cup \{\pi_k\}] = g[G_{k+1}]$ and similarly for h. By induction on k, we can show that the diagram holds $\forall k \geq 1$. By a composition of these diagrams, we can get the following diagram.

$$\begin{array}{ccc}
g[G_1] & \sqsubseteq_{(\rho,\sigma)} & h[G_1] \\
\sqsubseteq_{(\equiv,\rho^{-1})} & & \sqsubseteq_{(\equiv,\rho^{-1})} \\
g[G_2] & \sqsubseteq_{(\rho,\sigma)} & h[G_2] \\
\sqsubseteq_{(\equiv,\rho^{-1})} & & \sqsubseteq_{(\equiv,\rho^{-1})} \\
\vdots & \vdots & \vdots \\
\sqsubseteq_{(\equiv,\rho^{-1})} & & \sqsubseteq_{(\equiv,\rho^{-1})} \\
g[G_{n+1}] & \sqsubseteq_{(\rho,\sigma)} & h[G_{n+1}]
\end{array}$$

Note that since $G_{n+1} = trans(g)$, the last terms we obtain in this process are $g[trans(g)]$ and $h[trans(g)]$ as desired in the lemma. Using transitivity of $\sqsubseteq_{(\equiv,\rho^{-1})}$ we (nearly) get the diagram of semi saturation lemma. The other observation that g' and h' are related by $\sqsubseteq_{(\equiv,\sigma)}$ comes from using part 4 of the lemma 24 which implies that having each transition in g and all new ones in g' being matched by $\equiv$ in h', all transitions in g' end up matched by $\equiv$ in h'. For proving this, we first prove two subproofs.

1. Consider $\pi_{g'} = p \xrightarrow{a_k} p' \in g[G_{n+1}] - g[G_1]$ such that π'_g was added in the construction of $g[G_k][\pi_k]$ from $g[G_k]$ for some k, with $\pi_k = p_k \xrightarrow{a_k} p'_k$. From lemma 24 part 4.4b, we know that $\forall (p,q) \in R\ \exists q' : q \xrightarrow{a_k} q' \in h[G_k][\pi_k] \wedge p'Rq'$. Since $trans(h[G_k][\pi_k]) \subseteq trans(h[G_{n+1}])$, $q \xrightarrow{a_k} q'$ is also a transition in $h[G_{n+1}]$.
2. If $\pi_g \in g[G_1]$ then $\pi_g = \pi_k = p_k \xrightarrow{a_k} p'_k$ for some k. In the construction of $h[G_k][\pi_k]$ from $h[G_k]$, we can use lemma 24 part 4.4a to see that $\forall (p_k, q) \in R\ \exists q' : q \xrightarrow{a_k} q' \in h[G_k][\pi_k] \wedge p'_k R q'$. Again, $q \xrightarrow{a_k} q'$ is also a transition in $h[G_{n+1}]$.

As a result of the above two subproofs, we know that for for all $(p,q) \in R$ and any $\pi = p \xrightarrow{c} p' \in g[G_{n+1}]$, there is a transition $q \xrightarrow{c} q'$ in $h[G_{n+1}]$ such that $(p',q') \in R$. This proves the first clause of the definition of $R : g' \sqsubseteq_{(\equiv,\rho^{-1})} h'$ where $g' = g[G_{n+1}]$ and $h' = h[G_{n+1}]$. The second part trivially follows from $R : g \sqsubseteq_{(\rho,\rho^{-1})} h$. ■

By another application of the semi saturation lemma we can have every transition in h' also matched by $\equiv$ to some transition in g'. Thus we can reduce $\sqsubseteq_{(\equiv,\sigma)}$ further to $\sqsubseteq_{(\equiv,\equiv)}$, which is strong bisimilarity.

Lemma 27. (Saturation) *For finite process graphs $g, h \in \mathbb{R}$ and a relation R such that $R : g\underline{\Box}_{(\rho,\sigma)}h$, where ρ, σ are preorders and $\rho \circ \sigma^{-1} \subseteq \rho$, there exist derivatives g', h', g'', h'' such that the following diagram holds.*

$$\begin{array}{ccc} g & \underline{\Box}_{(\rho,\sigma)} & h \\ \underline{\Box}_{(\equiv,\rho^{-1})} & & \underline{\Box}_{(\equiv,\rho^{-1})} \\ g' & \underline{\Box}_{(\equiv,\sigma)} & h' \\ \underline{\Box}_{(\equiv,\sigma)} & & \underline{\Box}_{(\equiv,\sigma)} \\ g'' & \underline{\Box}_{(\equiv,\equiv)} & h'' \end{array}$$

Moreover, we have

1. $id_g : g \, \underline{\Box}_{(\equiv,\rho^{-1})} g'$ *and* $id_g : g' \, \underline{\Box}_{(\equiv,\sigma)} g''$
2. $id_h : h \, \underline{\Box}_{(\equiv,\rho^{-1})} h'$ *and* $id_h : h' \, \underline{\Box}_{(\equiv,\sigma)} h''$
3. $R : g' \, \underline{\Box}_{(\equiv,\sigma)} \, h'$ *and* $R : g'' \, \underline{\Box}_{(\equiv,\equiv)} \, h''$

PROOF. The first part (g and h to g' and h') follows directly from lemma 26. We get $R : g' \, \underline{\Box}_{(\equiv,\sigma)} \, h'$. By proposition 2.2, we get $R^{-1} : h' \, \underline{\Box}_{(\sigma^{-1},\equiv)} \, g'$. Since $\equiv$ and σ^{-1} are both reflexive and transitive and we also have $\sigma^{-1} \circ \equiv^{-1} \subseteq \sigma^{-1}$, we use lemma 26 to get $R^{-1} : h'' \, \underline{\Box}_{(\equiv,\equiv)} \, g''$ which is the same as $R : g'' \, \underline{\Box}_{(\equiv,\equiv)} \, h''$. Parts 1, 2 and 3 follow easily as consequences of corresponding parts in lemma 26.

■

Corollary 28. *For finite process graphs $g, h \in \mathbb{R}$ and a relation R such that $R : g\underline{\Box}_{(\rho,\sigma)}h$, where ρ, σ are preorders and $\sigma^{-1} \subseteq \rho$, there exist derivatives g'', h'' such that the following diagram holds.*

$$\begin{array}{ccc} g & \underline{\Box}_{(\rho,\sigma)} & h \\ \underline{\Box}_{(\equiv,\rho^{-1})} & & \underline{\Box}_{(\equiv,\rho^{-1})} \\ g'' & \underline{\Box}_{(\equiv,\equiv)} & h'' \end{array}$$

Also

1. $id_g : g \, \underline{\Box}_{(\equiv,\rho^{-1})} g''$
2. $id_h : h \, \underline{\Box}_{(\equiv,\rho^{-1})} h''$
3. $R : g'' \, \underline{\Box}_{(\equiv,\equiv)} \, h''$

■

In the above corollary, we have shown how to obtain strongly bisimilar g'' and h'' from g and h by traversing down $\underline{\Box}_{(\equiv,\rho^{-1})}$. This result is not restricted to only the cases where $\sigma^{-1} \subseteq \rho$. In the symmetrical case where $\rho^{-1} \subseteq \sigma$, we can also obtain strongly bisimilar graphs¶.

¶ $\sigma^{-1} \subseteq \rho$ or $\rho^{-1} \subseteq \sigma$ is not an unusual condition. Simulation, strong

Corollary 29. *For finite process graphs $g, h \in \mathbb{R}$ and a relation R such that $R : g \underline{\square}_{(\rho,\sigma)} h$, where ρ, σ are preorders and $\rho^{-1} \subseteq \sigma$, there exist derivatives g'', h'' such that the following diagram holds.*

$$\begin{array}{ccc} g & \underline{\square}_{(\rho,\sigma)} & h \\ \underline{\square}_{(\equiv,\sigma)} & & \underline{\square}_{(\equiv,\sigma)} \\ g'' & \underline{\square}_{(\equiv,\equiv)} & h'' \end{array}$$

Moreover,

1. $id_g : \ g \ \underline{\square}_{(\equiv,\sigma)} g''$
2. $id_h : \ h \ \underline{\square}_{(\equiv,\sigma)} h''$
3. $R : \ g'' \ \underline{\square}_{(\equiv,\equiv)} \ h''$

■

Theorem 30. *For finite process graphs $g, h \in \mathbb{R}$ and a relation R such that $R : g \underline{\square}_{(\rho,\sigma)} h$, where ρ, σ are preorders and $\rho^{-1} = \sigma$, there exist derivatives g'', h'' such that the following diagram holds.*

$$\begin{array}{ccc} g & \underline{\square}_{(\rho,\rho^{-1})} & h \\ \underline{\square}_{(\equiv,\rho^{-1})} & & \underline{\square}_{(\equiv,\rho^{-1})} \\ g'' & \underline{\square}_{(\equiv,\equiv)} & h'' \end{array}$$

The following hold as well

1. $id_g : \ g \ \underline{\square}_{(\equiv,\rho^{-1})} g''$
2. $id_h : \ h \ \underline{\square}_{(\equiv,\rho^{-1})} h''$
3. $R : \ g'' \ \underline{\square}_{(\equiv,\equiv)} \ h''$
4. $S : g \underline{\square}_{(\rho,\rho^{-1})} h$ *where* $S = id_g \circ R \circ id_h^{-1}$

■

Theorem 30.4 says that the diagram commutes. Using proposition 2.3, we can show that $id_g \circ R \circ id_h^{-1} \ : \ g \underline{\square}_{(\equiv \circ \equiv \circ \rho, \rho^{-1} \circ \equiv \circ \equiv)} h$. So we have another relation $S = id_g \circ R \circ id_h^{-1}$ such that $S : g \underline{\square}_{(\rho,\rho^{-1})} h$.

That the diagram commutes is critical for the completeness proof where we are able to prove $g \ \underline{\square}_{(\rho,\rho^{-1})} g'', h \ \underline{\square}_{(\rho,\rho^{-1})} h''$ and $h'' \ \underline{\square}_{(\rho,\rho^{-1})} g''$ within the framework of the axiomatization and use the transitivity of $\underline{\square}_{(\rho,\rho^{-1})}$ to prove $g \ \underline{\square}_{(\rho,\rho^{-1})} h$ within the framework of the axiomatization. The reason we are unable to prove completeness for preorders is because when $\sigma \neq \rho^{-1}$ the diagram does not necessarily commute.

8 Completeness of $\mathcal{A}_{(\rho,\rho^{-1})}$

The proof of completeness of $\mathcal{A}_{(\rho,\rho^{-1})}$ hinges on reducing the two graphs to be compared g and h, to graphs g' and h' such that

---

bisimulation and observational equivalence have ρ and σ such that one of these conditions hold [5] .

$g \sqsubseteq_{(\rho,\rho^{-1})} g' \wedge h \sqsubseteq_{(\rho,\rho^{-1})} h' \wedge g' \sqsubseteq_{(\equiv,\equiv)} h' \iff g \sqsubseteq_{(\rho,\rho^{-1})} h$. That is, the graphs g' and h' turn out to be strongly bisimilar if and only if g and h were (ρ, ρ^{-1})-bisimilar. This relation between g, g', h, h' can also be shown diagrammatically in the following manner.

$$\begin{array}{ccc} g & \sqsubseteq_{(\rho,\rho^{-1})} & h \\ \sqsubseteq_{(\rho,\rho^{-1})} & & \sqsubseteq_{(\rho,\rho^{-1})} \\ g' & \sqsubseteq_{(\equiv,\equiv)} & h' \end{array}$$

The strong bisimilarity of g' and h' can be checked within the framework of BPA_{LR} as it is complete, while the relation between the pairs g, g' and h, h' can be proven using the axiom schema AV_ρ or AV.

Lemma 31. *For any graph $h \in \mathbb{R}$, transition $\pi = q \xrightarrow{b} q' \in trans(h)$ and $a\rho b$,*

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|h\| =_\rho \|h_a\langle q \xrightarrow{b} q'\rangle\|$$

Proof. Let $\|h\| \equiv \langle X_1|E\rangle$. Let X_k and X_j be the variables that correspond to the node q and q' respectively in the construction of $\|h\|$ (see lemma 15). Then T_k must be of the form $b.X_j + T$ for some term T. Thus $\|h\| \equiv \langle X_1|X_k = b.X_j + T, E_{-k}\rangle$. Consider the process graph $h_a\langle q \xrightarrow{b} q'\rangle$ which is derived from h by adding a single transition $q \xrightarrow{a} q'$ in the graph h, where $a\rho b$. Following the algorithm of $\|\ \|$ given in lemma 15, it should be easy to see that $\|h_a\langle q \xrightarrow{b} q'\rangle\| \equiv \langle X_1|X_k = a.X_j + b.X_j + T, E_{-k}\rangle$. Thus in order to prove

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|h\| =_\rho \|h_a\langle q \xrightarrow{b} q'\rangle\|$$

we need to prove $\mathcal{A}_{(\rho,\rho^{-1})} \vdash \langle X_1|X_k = b.X_j + T, E_{-k}\rangle =_\rho \langle X_1|X_k = a.X_j + b.X_j + T, E_{-k}\rangle$

Since $a\rho b$, the above is a direct application of AV (or application of AV_ρ, $A4_\rho$, congruence and substitution rule) and is hence provable.

Theorem 32. (Completeness)

If $g, h \in \mathbb{R}$ and $\rho \subseteq \mathcal{O} \times \mathcal{O}$ is a preorder where $\mathcal{O}$ has no empty observables, then

$$g \sqsubseteq_{(\rho,\rho^{-1})} h \implies \mathcal{A}_{(\rho,\rho^{-1})} \vdash \|g\| =_\rho \|h\|$$

Proof. If $g \sqsubseteq_{(\rho,\rho^{-1})} h$, there exists [5] a relation R such that $R : g \sqsubseteq_{(\rho,\rho^{-1})} h$. For $\pi_g = p_o \xrightarrow{a} p'_o \in trans(g)$ we can construct $T(\pi_g)$ and $U(\pi_g)$ as shown in section 7. Since it is finite, let $U(\pi_g) =$

$\{\pi_1, \pi_2, \ldots \pi_n\}$ for some n. From lemma 23 we know that for any $q \xrightarrow{b} q' \in U(\pi_g)$, we have $a\rho b$. Hence by applying lemma 31 we have‖

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|h\| \ =_\rho \|h_a\langle\pi_1\rangle\|$$

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|h_a\langle\pi_1\rangle\| \ =_\rho \|h_a\langle\pi_1\rangle\langle\pi_2\rangle\|$$

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|h_a\langle\pi_1\rangle \ldots \langle\pi_{k-1}\rangle\| \ =_\rho \|h_a\langle\pi_1\rangle \ldots \langle\pi_k\rangle\|$$

Using the transitivity of $=_\rho$ (being an equational axiomatization $\mathcal{A}_{(\rho,\rho^{-1})}$ has the appropriate axioms for transitivity, reflexivity and symmetry but for brevity these are not mentioned) and the definition of application of a set of transitions (definition 19), we get

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|h\| \ =_\rho \|h_a\langle U_i(\pi_g)\rangle\|$$

Similarly, let $T(\pi_g) = \{\chi_1, \chi_2, \ldots \chi_m\}$. From lemma 23 we know that for any $p \xrightarrow{c} p' \in T(\pi_g)$, we have $a\rho c$. Hence by lemma 31,

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|g\| \ =_\rho \|g_a\langle\chi_1\rangle\|$$

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|g_a\langle\chi_1\rangle\| \ =_\rho \|g_a\langle\chi_1\rangle\langle\chi_2\rangle\|$$

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|g_a\langle\chi_1\rangle \ldots \langle\chi_{m-1}\rangle\| \ =_\rho \|g_a\langle\chi_1\rangle \ldots \langle\chi_m\rangle\|$$

Again, using the transitivity of $=_\rho$ we get

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|g\| \ =_\rho \|g_a\langle T_i(\pi_g)\rangle\|$$

From lemma 21 we have $g_a\langle T_i(\pi_g)\rangle \equiv g[\pi_g]$ and $h_a\langle U_i(\pi_g)\rangle \equiv h[\pi_g]$ and thus,

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|h\| \ =_\rho \|h[\pi_g]\|$$

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|g\| \ =_\rho \|g[\pi_g]\|$$

However, we are unable to prove the following even though from partial saturation lemma 24, we know $g[\pi_g] \sqsubseteq_{(\rho,\rho^{-1})} h[\pi_g]$ to be true.

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|g[\pi_g]\| \ =_\rho \|h[\pi_g]\|$$

Thus, the vertical relations between $g, g[\pi_g], h, h[\pi_g]$ in the diagram of partial saturation lemma (with the additional restriction $\sigma = \rho^{-1}$) can be proven within the context of $\mathcal{A}_{(\rho,\rho^{-1})}$ while the parts

‖Note that $trans(h) \subseteq h_a\langle q \xrightarrow{b} q'\rangle$ and hence if a transition is in h, it is also in $h_a\langle q \xrightarrow{b} q'\rangle$. For instance the transition π_2 is also in $h_a\langle\pi_1\rangle$ and hence lemma 31 can be applied to it.

3 and 4, which claim that $g[\pi_g]$ and $h[\pi_g]$ are related by $\sqsubseteq_{(\rho,\rho^{-1})}$ and in fact are "closer in terms of strong bisimilarity", cannot be proven in this manner directly. To be accurate, we have only been able to prove $g\sqsubseteq_{(\rho,\rho^{-1})}g'$ whereas the relation between g and g' in partial saturation lemma is $g\sqsubseteq_{(\equiv,\rho^{-1})}g'$. However, for our purposes this will prove adequate.

Since the diagram of semi saturation lemma 26 is constructed by tiling up partial saturation lemma diagrams, each vertical relation in the semi saturation lemma diagram can also be proven within the context of $\mathcal{A}_{(\rho,\rho^{-1})}$ but we cannot prove $\|g'\| =_\rho \|h'\|$ although $g'\sqsubseteq_{(\equiv,\rho^{-1})}h'$ is true. To be precise, we are able to prove $\|g[G_k]\| =_\rho \|g[G_k][\pi_k]\|$ within the framework for every $k \geq 1$, where G_k and π_k are as defined in the proof outline of semi saturation lemma. Using transitivity, we can prove $\|g[G_1]\| =_\rho \|g[G_{n+1}]\|$ and since $g[G_1] \equiv g, g[G_{n+1}] \equiv g'$, we get our desired result.

By similar reasoning, the vertical relations in the saturation lemma 27 diagram which are obtained by tiling two semi saturation lemma diagrams can also be proven within the context of the axiomatization. Once again we know $g''\sqsubseteq_{(\equiv,\equiv)}h''$ to be true but cannot prove it in this manner. We now rely on BPA_{LR}, which is a complete axiomatization for $\sqsubseteq_{(\equiv,\equiv)}$ and therefore the following can be proven in its framework

$$BPA_{LR} \vdash \|g''\| = \|h''\|$$

By a translation of the above proof,

$$BPA_{\rho LR} \vdash \|g''\| =_\rho \|h''\|$$

Since $BPA_{\rho LR} \subset \mathcal{A}_{(\rho,\rho^{-1})}$, we can also prove

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \|g''\| =_\rho \|h''\|$$

Now we use the fact that the diagram of theorem 30 commutes. Using symmetry of $=_\rho$, and putting together the relations between g, g'', h, h'' proven above, we get

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \; \|g\| \quad =_\rho \quad \|g''\| \quad =_\rho \quad \|h''\| \quad =_\rho \quad \|h\|$$

By transitivity of $=_\rho$, we get

$$\mathcal{A}_{(\rho,\rho^{-1})} \vdash \; \|g\| =_\rho \|h\|$$

■

We have proven $g\sqsubseteq_{(\rho,\rho^{-1})}h$ in the framework of the axiomatization. Hence $\mathcal{A}_{(\rho,\rho^{-1})}$ is sound complete axiomatization of $\sqsubseteq_{(\rho,\rho^{-1})}$ for finite process graphs, where ρ is an axiomatizable preorder on a set of observables that excludes all empty observables.

9 Conclusion

Under the assumption that the set of observables does not contain any empty observable, and relative to an axiomatization for the relation on observables ρ, we have been able to provide an axiomatization that is complete for all equivalence relations in the class of (ρ, σ)-bisimilarities. The axiomatization builds upon an existing axiomatization for strong bisimilarity [6], and after a simple translation of this, only requires one additional axiom AV_ρ. The intuition of this axiom is similar to that of $x = x + x$ which can prove that strong bisimilarity is insensitive to addition of a duplicate edge. AV_ρ says that $\sqsubseteq_{(\rho,\rho^{-1})}$ is insensitive to the addition of all edges that are related by ρ to an existing edge.

We have also shown that under the absence of an empty observable, $\sqsubseteq_{(\rho,\rho^{-1})}$ also turns out to be a congruence. In the presence of empty observables, a study of the nature of observables that are related to an empty action by ρ and σ might lead to a characterization of the smallest precongruence or congruence containing a given (ρ, σ) preorder or equivalence bisimilarity. Such observables are termed ρ-preemptive and σ-preemptive as they may occur without being matched by any observable (other than an empty action) of a related process which ends up preserving choice in contrast to the process that does a preemptive action.

The generalization presented in this paper cannot provide an axiomatization of observational equivalence because of the presence of empty observables in its characterization as $\sqsubseteq_{(\hat{=}, \hat{=})}$, which was presented in [5]. The other problem is that while $\approx$ might be considered on finite process graphs, the use of $\mathcal{O} = Act^*$ in the definition of $\sqsubseteq_{(\hat{=}, \hat{=})}$ converts the finite process graphs that have cycles into infinitely branching graphs for which our axiomatization has not been proven complete.

Another problem worth inquiring into is the inequational axiomatization of (ρ, σ) precongruences and preorders. Our method of saturation which is similar to the Δ-saturation method in [6] does not preserve the preorder in the sense that the diagram in lemma 27 does not commute unless $\sigma = \rho^{-1}$. A construction of derivatives that leads to a commuting diagram may be the key to finding a complete axiomatization for precongruences. An attempt to find such saturation methods for precongruences such as conformance [4] and efficiency precongruence [3] have led us to characterize some properties of the kernels of these relations. Presented in [12], these observations might also prove useful for an inequational extension of the axiomatization.

References

[1] Luca Aceto. On relating concurrency and nondeterminism. In Stephen D. Brookes, Michael G. Main, Austin Melton, Michael W. Mislove, and David A. Schmidt, editors, *Mathematical Foundations of Programming Semantics*, Lecture Notes in Computer Science, pages 376–402. Springer-Verlag, Pittsburgh, PA, USA, 1991.

[2] S. Arun-Kumar. On bisimilarities induced by relations on actions. In *Proceedings 4th IEEE International Conference on Software Engineering and Formal Methods, Pune, India*. IEEE Computer Society Press, 2006.

[3] S. Arun-Kumar and M. Hennessy. An efficiency preorder for processes. *Acta Informatica*, 29:737–760, 1992.

[4] S. Arun-Kumar and V. Natarajan. Conformance: A precongruence close to bisimilarity. In *STRICT, Berlin 1995*, volume number 526 in Workshops in Computing Series, pages 55–68. Springer-Verlag, 1995.

[5] S. Arun-Kumar, Amarinder Singh Randhawa, and Pranav Singh. Parametrised bisimulations: Some characterisations. submitted.

[6] J.A. Bergstra and J.W. Klop. A complete inference system for regular processes with silent moves. In *F.R. Drake, J.K. Truss (Editors), Proceedings of Logic Colloquium 1986, North Holland, Amsterdam*, pages 21–81. 1988.

[7] Flavio Corradini, Roberto Gorrieri, and Marco Roccetti. Performance preorder and competitive equivalence. *Acta Informatica*, 34(11):805–835, 1997.

[8] Robin Milner. A complete axiomatisation for observational congruence of finite-state behaviours. *Information and Computation*, 81(2).

[9] Robin Milner. A complete inference system for a class of regular behaviors. *Journal of Computer and System Sciences*, 28:439–466, 1984.

[10] Robin Milner. *Communication and Concurrency.* Prentice-Hall International, 1989.

[11] Arto Salomaa. Two complete axiom systems for the algebra of regular events. *J. ACM*, 13(1):158–169, 1966.

[12] Pranav Singh. Axiomatizations of bisimulation based relations. Master's thesis, Indian Institute of Technology, Delhi, Department of Computer Science and Engineering, 2005.