Graphical representation of covariant-contravariant modal formulas

Luca Aceto Anna Ingólfsdóttir
ICE-TCS, School of Computer Science
Reykjavik University*
Iceland

Ignacio Fábregas David de Frutos Escrig Miguel Palomino
Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid†
Spain

Covariant-contravariant simulation is a combination of standard (covariant) simulation, its contravariant counterpart and bisimulation. We have previously studied its logical characterization by means of the covariant-contravariant modal logic. Moreover, we have investigated the relationships between this model and that of modal transition systems, where two kinds of transitions (the so-called may and must transitions) were combined in order to obtain a simple framework to express a notion of refinement over state-transition models. In a classic paper, Boudol and Larsen established a precise connection between the *graphical* approach, by means of modal transition systems, and the *logical* approach, based on Hennessy-Milner logic without negation, to system specification. They obtained a (graphical) representation theorem proving that a formula can be represented by a term if, and only if, it is consistent and prime. We show in this paper that the formulae from the covariant-contravariant modal logic that admit a "graphical" representation by means of processes, modulo the covariant-contravariant simulation preorder, are also the consistent and prime ones. In order to obtain the desired graphical representation result, we first restrict ourselves to the case of covariant-contravariant systems without bivariant actions. Bivariant actions can be incorporated later by means of an encoding that splits each bivariant action into its covariant and its contravariant parts.

1 Introduction

Modal transition systems (MTSs) were introduced in [8, 9] as a model of reactive computation based on states and transitions that naturally supports a notion of *refinement*. This is connected with the use of Hennessy-Milner Logic without negation as a specification language: a specification describes the collection of (good) properties that any implementation has to fulfil. More generally, a process p is considered to be better than q if the set of formulae satisfied by q is included in the set of formulae satisfied by p. The tight connections between these two ways of expressing the notions of specification and refinement were studied in [4]. There the authors talked about "graphical" representation (by means of one or several MTSs) of logical specifications, and completely characterized the collection of logical specification that can be "graphically represented". These are the so-called prime, consistent formulae.

There are two types of modal operators in Hennessy-Milner Logic: $\langle a\rangle$ and $[a]$, for each action a. Intuitively, a formula $\langle a\rangle\varphi$ indicates that it must be possible to execute a and reach a state that satisfies

*Research supported by the project 'Processes and Modal Logics' (project nr. 100048021) of the Icelandic Research Fund, and the Abel Extraordinary Chair programme within the NILS Mobility Project.

†Research supported by Spanish projects DESAFIOS10 TIN2009-14599-C03-01, TESIS TIN2009-14321-C02-01 and PROMETIDOS S2009/TIC-1465

Submitted to:
EXPRESS 2011

φ, while $[a]\varphi$ imposes that this will happen after any execution of a from the current state. It is well known that these two operators reflect the duality $\exists$-$\forall$, so that any process satisfying a $\langle a\rangle\varphi$ formula *must* include some a-labelled transition reaching a state satisfying φ, whereas the constraint expressed by a $[a]\varphi$ formula is better understood in a negative way: a process satisfying it *may not* contain an a-labelled transition reaching a state that does not satisfy φ. In particular, the formula $[a]\bot$ indicates that a process cannot execute a in its initial state, and therefore, using these formulae, we can limit the set of actions offered at any state.

In order to reflect these two kinds of constraints at the "operational" level, MTSs contain two kinds of transitions: the *may* transitions and the *must* transitions. Then we can use MTSs both as specifications or as implementations, and the notion of refinement imposes that, in order to implement correctly a specification, an implementation should exhibit all the *must* transitions in the MTS that describes the specification and may not include any transition that is not allowed by the specification: we cannot add any new *may* transition, although those in the specification could either disappear, be preserved or turned into *must* transitions. The relation between *may* and *must* is reflected in the formal definition of MTSs by requiring that each must transition is also a may transition.

The conditions defining the notion of refinement between MTSs obviously resemble those defining simulation and bisimulation. For may transitions we have a contravariant simulation condition, expressing the fact that no new (non-allowed) *may* transition can appear when refining a specification. Since we impose that *must* transitions induce the corresponding *may* transitions, we could think that they are related in a "bisimulation-like" style. However, this is not the case since the contravariant simulation condition imposed on the may part can be covered by a *may* transition without *must* counterpart. In fact, this is crucial in order to capture the principle that a *may* transition can be refined by a *must* transition.

Some of the authors of this paper thought that a more direct combination of simulation and bisimulation conditions could capture in a more flexible way all the ideas on which the specification of systems by means of modal systems and modal logics is based, and we looked for the clearest and most general framework to express those modal constraints. We found that covariant-contravariant systems (sometimes abbreviated to cc-systems) are a possible answer to this quest, combining pure (covariant) simulation, its contravariant counterpart and bisimulation.

We started the study of *covariant-contravariant simulation* in [5], and the modal logic characterizing it was presented in [7]. (In what follows, we refer to this logic as cc-modal logic.) In the most general case, we consider a partition of the set of actions into three sets: the collection of covariant actions, that of contravariant actions, and the set of bivariant actions. Intuitively, one may think of the covariant actions as being under the control of the specification LTS, and transitions with such actions as their label should be simulated by any correct implementation of the specification. On the other hand, the contravariant actions may be considered as being under the control of the implementation (or of the environment) and transitions with such actions as their label should be simulated by the specification. The bivariant actions are treated as in the classic notion of bisimulation.

We will see in this paper that, as in the MTS setting, the consistent and prime formulae from the cc-modal logic are exactly those that admit a "graphical" representation by means of processes modulo the covariant-contravariant simulation preorder. Moreover, each formula in the cc-modal logic can be represented "graphically" by a (possibly empty) finite set of processes.

The proofs of these representation results are inspired by the developments in [4]. There are, however, subtle differences because, in covariant-contravariant systems, each action has a single modality (covariant, contravariant, bivariant), while in MTSs we can combine both *may* and *must* transitions.

In fact, in order to obtain the desired graphical representation, for technical reasons we first restrict ourselves to the case of covariant-contravariant systems without bivariant actions. The reason that justi-

fies this constraint is that bivariant actions cannot be approximated in a non-trivial way (either we have one of them as itself, or we do not have it at all). Instead, covariant and contravariant actions behave in a more flexible way and we can obtain the desired characterization result by following the lead of the work done for MTSs.

Then we observe that bivariant actions can be seen as the combination of a covariant and a contravariant action. In fact, this also corresponds with the idea used in [1] when relating MTSs and cc-systems. Indeed, the constraint imposed on *must* transitions in MTSs, where they should always be accompanied by their *may* counterparts, tells us somehow that they have a "nearly" bivariant behaviour. (To be more precise, they are first covariant, but they are also "semi"-contravariant because when comparing two processes p and q, any *must* transition in q should fit with either a corresponding *must* transition in p, or at least with a *may* transition there.)

We could say that the very recent development of the notion of *partial bisimulation* in the setting of labelled transition systems (LTSs) presented in [3] has completed the spectrum of modal simulations. Partial bisimulation combines plain bisimulation [13, 14] and simulation, also by means of a partition of the set of actions. For the actions in the distinguished set B we have bisimulation-like conditions, while for the others we only impose simulation. Note that, instead, *may* transitions in MTSs corresponded to contravariant simulation conditions, and therefore, partial bisimulation can be seen as a dual of MTSs, and covariant-contravariant systems (cc-systems) as a unifying framework where we can combine the refinement ideas in the theory of MTSs with the explicit consideration of the constraints imposed by the environment, which is possible when partial bisimulation is used. Once we know that the formulae from the modal logic for cc-systems also afford a graphical representation, we will be able to integrate the logical formulae into the development of systems using any of the models discussed above.

The remainder of the paper is organized as follows. Section 2 is devoted to the necessary background on covariant-contravariant simulations, whereas in Section 3 we summarize the results on covariant-contravariant modal formulae. In Section 4 we develop the study of the graphical representation of cc-modal formulae for processes without bivariant actions. Afterwards, in Section 5, we show how we can work with cc-systems with bivariant actions. Finally, Section 6 concludes the paper and describes some future research that we plan to pursue.

2 Covariant-contravariant systems

We start the technical part of the paper by defining the covariant-contravariant simulation semantics for processes. Our semantics is defined over *Labelled Transition Systems* (LTS) $S = (\mathbf{P}, A, \longrightarrow)$, where $\mathbf{P}$ is a set of process states, A is a set of actions and $\longrightarrow \subseteq \mathbf{P} \times A \times \mathbf{P}$ is a transition relation on processes. We follow the standard practice and write $p \xrightarrow{a} q$ instead of $(p,a,q) \in \longrightarrow$. Because of the covariant-contravariant view, we assume that A is partitioned into A^l and A^r, expressed as $A = A^l \uplus A^r$. As we have already mentioned in the introduction, we will delay the consideration of the general case where we have also bivariant actions in a third class A^{bi} until Section 5.

Covariant-contravariant simulation can now be defined as follows:

Definition 1 *Let $S = (\mathbf{P}, A^l \uplus A^r, \longrightarrow)$ be an LTS. A* covariant-contravariant simulation *over S is a relation $R \subseteq \mathbf{P} \times \mathbf{P}$ such that, whenever $p, q \in \mathbf{P}$ and $p\ R\ q$, we have:*

- *For all $a \in A^r$ and all $p \xrightarrow{a} p'$, there exists some $q \xrightarrow{a} q'$ with $p'\ R\ q'$.*
- *For all $a \in A^l$ and all $q \xrightarrow{a} q'$, there exists some $p \xrightarrow{a} p'$ with $p'\ R\ q'$.*

We will write $p \lesssim_{cc} q$ if there exists a covariant-contravariant simulation R such that $p\ R\ q$.

It is well known that the relation $\lesssim_{cc}$ is a preorder.

In this study we will be mainly concerned with "finite" properties of systems, which will be either captured by (finite) logic formulae, or by finite processes that can be described by means of process terms.

Definition 2 *Assume that $A = A^l \uplus A^r$. Then the collection of process terms, ranged over by p, q etc. is given by the following syntax:*

$$p ::= 0 \mid \omega \mid a.p \mid p + p,$$

where $a \in A$. We denote the set of process terms by $\mathscr{P}$.

The size of a process term is its length in symbols.

We note that our set $\mathscr{P}$ of process terms is basically the set of *BCCSP* terms introduced in [15]. The only addition to the signature of BCCSP is the constant ω, which will be used to denote the least LTS modulo $\lesssim_{cc}$. However, we assume a classification of the actions in two (disjoint) sets, although this is not reflected in the syntactic structure of the terms. Even if $\mathscr{P}$ only contains finite terms, by means of ω we will obtain the full contravariant process which can execute any action at any time.

In [5, 6, 7] we used a more general definition for covariant-contravariant simulations which includes also bivariant actions, but since in the presence of these bivariant actions some technical problems appear (in particular the process ω will not be the least process with respect to the covariant-contravariant simulation preorder), we have preferred to first develop all the results without bivariant actions and, in Section 5, we will describe how they can be extended to a setting with bivariant actions.

Definition 3 *The operational semantics of $\mathscr{P}$ is defined by the following rules:*

- $\omega \xrightarrow{b} \omega$ *for all* $b \in A^l$,
- $a.p \xrightarrow{a} p$ *for all* $a \in A$,
- $p \xrightarrow{a} p'$ *implies* $p + q \xrightarrow{a} p'$,
- $q \xrightarrow{a} q'$ *implies* $p + q \xrightarrow{a} q'$.

Observe that if $p \neq \omega$ and $p \xrightarrow{a} p'$, then the size of p' is smaller than the size of p.

It is clear that ω is the least possible element with respect to the cc-simulation preorder. That is, we have $\omega \lesssim_{cc} p$ for any p.

In what follows we assume that A is finite.

3 The covariant-contravariant modal logic

Covariant-contravariant modal logic has been introduced and studied in [7].

Definition 4 Covariant-contravariant modal logic $\mathscr{L}$ *has the following syntax:*

$$\varphi ::= \bot \mid \top \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [b]\varphi \mid \langle a \rangle \varphi \qquad (a \in A^r, b \in A^l).$$

The operators $\bot$, $\top$, $\wedge$ and $\vee$ have the standard meaning whereas the semantics for the modal operators is defined as follows:

$p \models [b]\varphi$ *if* $p' \models \varphi$ *for all* $p \xrightarrow{b} p'$,

$p \models \langle a \rangle \varphi$ *if* $p' \models \varphi$ *for some* $p \xrightarrow{a} p'$.

We say that a formula φ is consistent if there is some p such that $p \models \varphi$.

The modal depth *of a formula is the maximum nesting of modal operators in it.*

The covariant-contravariant logic characterizes the covariant-contravariant simulation semantics over image-finite processes. Before we state this result formally we introduce some notation. We define the set of formulae that a process p satisfies by $\mathscr{L}(p) = \{\phi \mid p \models \phi\}$ and the logical preorder $\sqsubseteq_{\mathscr{L}}$ as follows: $p \sqsubseteq_{\mathscr{L}} q$ iff $\mathscr{L}(p) \subseteq \mathscr{L}(q)$. Recall that an LTS is *image finite* iff the set $\{p' \mid p \xrightarrow{a} p'\}$ is finite for each process p and action a.

Now we have the following theorem:

Theorem 1 ([7]) *If the LTS S is image finite then $\lesssim_{cc} = \sqsubseteq_{\mathscr{L}}$ over S.*

Clearly the processes in $\mathscr{P}$ are image finite.

4 Graphical representation of formulae

Whenever we have a (modal) logic characterizing some semantics for processes, we could look for a single formula that characterizes completely the behaviour of a process logically; this is a so-called *characteristic formula.* This subject has been studied by many authors in the literature, but we will just refer here to the book [2] for more details and further references to the original literature.

It is clear that, since we only allow for finite formulae without any fixed-point operator, we can only treat "finite" processes, such as those definable by our simple process algebra $\mathscr{P}$. However, the recursive definition of the characteristic formulae in what follows gives us immediately the framework for extending our results to finite-state processes following standard lines.

Definition 5 *A formula $\phi \in \mathscr{L}$ is a* characteristic formula *for a process p iff $p \models \phi$ and $\forall q.(q \models \phi \Rightarrow p \lesssim_{cc} q)$.*

In what follows, we write $\phi \leq \psi$ if $\{p \in P \mid p \models \phi\} \subseteq \{p \in P \mid p \models \psi\}$. We say that ϕ and ψ are logically equivalent, written $\phi \equiv \psi$, iff $\phi \leq \psi$ and $\psi \leq \phi$.

Lemma 1 *The following statements hold.*

1. *A formula $\phi \in \mathscr{L}$ is a characteristic formula for a process p iff $\forall q.(q \models \phi \Leftrightarrow p \lesssim_{cc} q)$.*
2. *Assume that $\chi(p)$ and $\chi(q)$ are characteristic formulae for processes p and q, respectively. Then, we have that*
$$p \lesssim_{cc} q \text{ iff } \chi(q) \leq \chi(p).$$
3. *A characteristic formula for a process p is unique up to logical equivalence.*

Proof.

1. First assume that ϕ is a characteristic formula for a process p. By definition $\forall q.(q \models \phi \Rightarrow p \lesssim_{cc} q)$ holds. We have to prove that $\forall q.(p \lesssim_{cc} q \Rightarrow q \models \phi)$. To this end, assume that $p \lesssim_{cc} q$. As $p \models \phi$, by Theorem 1 we have that $q \models \phi$ and we are done.

   For the converse, as $p \lesssim_{cc} p$ we have that $p \models \phi$ and the result follows.

2. Assume that $\chi(p)$ and $\chi(q)$ are characteristic formulae for processes p and q, respectively. First assume that $p \lesssim_{cc} q$ and that $r \models \chi(q)$. By Definition 5, $q \lesssim_{cc} r$ and thus $p \lesssim_{cc} r$. By the previous clause of the Lemma, also $r \models \chi(p)$. As r was arbitrary, this shows that $\chi(q) \leq \chi(p)$. Next, assume that $\chi(q) \leq \chi(p)$. As $q \models \chi(q)$ then $q \models \chi(p)$, and by definition of the characteristic formula, $p \lesssim_{cc} q$.

3. This claim follows directly from statement 2 above. □

As a characteristic formula for a process p is unique up to logical equivalence, we can denote it by $\chi(p)$ unambiguously. The next lemma tells us that $\chi(p)$ exists for each process $p \in \mathscr{P}$.

Lemma 2 *The characteristic formula for a process $p \in \mathscr{P}$ can be obtained recursively as*

$$\begin{aligned} \chi(p) &= \bigwedge_{p \xrightarrow{a} p', a \in A^r} \langle a \rangle \chi(p') \wedge \bigwedge_{b \in A^l} [b]\Big(\bigvee_{p \xrightarrow{b} p'} \chi(p') \Big), \text{ if } p \neq \omega. \\ \chi(\omega) &= \top. \end{aligned}$$

Proof. First we prove that $p \models \chi(p)$, for each p. This follows by a simple induction on the size of p.

Next we prove that, for any q, $q \models \chi(p)$ implies $p \lesssim_{cc} q$ by induction on the size of q.

First we note that if $p = \omega$ then $\chi(\omega) = \top$ and $\omega \lesssim_{cc} q$; hence we obtain the result. Also, for the case $p = 0$, we have that $\chi(0)$ is equivalent to $\bigwedge_{b \in A^l} [b]\bot$. Thus if $q \models \chi(0)$, then the process q cannot perform any $b \in A^l$. This yields that $0 \lesssim_{cc} q$.

Now, let p be a process different from 0 and ω, and assume that $q \models \chi(p)$. First suppose that $p \xrightarrow{a} p'$ for some p' and some $a \in A^r$. As $q \models \bigwedge_{p \xrightarrow{a} p', a \in A^r} \langle a \rangle \chi(p')$, this implies that there is some $q \xrightarrow{a} q'$ with $q' \models \chi(p')$. Then, by induction, $p' \lesssim_{cc} q'$.

Next, assume that $q \xrightarrow{b} q'$, for some q' and $b \in A^l$. As $q \models \bigwedge_{b \in A^l} [b](\bigvee_{p \xrightarrow{b} p'} \chi(p'))$, we can conclude that there is some q' such that $q \xrightarrow{b} q'$ and $q' \models \chi(p')$, for some p' with $p \xrightarrow{b} p'$. Again, by induction, we conclude $p' \lesssim_{cc} q'$. □

Next we consider the converse problem, we want to represent a formula by a process, or at least by a finite set of processes.

Definition 6 *A formula ϕ is represented by a (single) process p if*

$$\forall q \in \mathscr{P}.\ [q \models \phi \text{ iff } p \lesssim_{cc} q].$$

A formula ϕ is represented by a finite set $M \subseteq \mathscr{P}$ of processes if

$$\forall q \in \mathscr{P}.\ [q \models \phi \text{ iff } \exists p \in M.\ p \lesssim_{cc} q].$$

It is clear that p represents ϕ iff $\{p\}$ represents ϕ. Moreover, the empty set of processes represents the formula $\bot$.

The following lemma connects the notion of "graphical representation" of formulae with that of characteristic formula for processes.

Lemma 3 *We have the following properties:*

1. *p represents ϕ iff $\phi \equiv \chi(p)$.*
2. *If $M \subseteq \mathscr{P}$ is finite and ϕ is a formula then*

$$M \text{ represents } \phi \text{ iff } \phi \equiv \bigvee_{p \in M} \chi(p).$$

Proof.

1. It follows directly from the definitions of these two concepts and Lemma 1.

2. For any $q \in \mathscr{P}$ we proceed as follows:

$$\exists p \in M.p \lesssim_{cc} q \Leftrightarrow \exists p \in M.q \models \chi(p) \Leftrightarrow q \models \bigvee_{p \in M} \chi(p).$$

Now the statement of the lemma follows easily from this fact and Definition 6. □

We want to characterize the set of formulae that can be represented by a finite set of processes, and in particular by a single process. For this purpose we introduce some notions of normal form for logical formulae.

Definition 7 *1. A formula ϕ is in normal form if it has the form*

$$\phi = \bigvee_{i \in I} (\bigwedge_{j \in J_i} \langle a^i_j \rangle \phi^i_j \wedge \bigwedge_{k \in K_i} [b^i_k] \psi^i_k).$$

where all ϕ^i_j and ψ^i_k are also in normal form. In particular, $\bot$ is obtained when $I = \emptyset$ and $\top$ when $I = \{1\}$ and $J_1 = K_1 = \emptyset$.

2. A formula ψ is in strong normal form if it has the form

$$\psi = \bigvee_{i \in I} \phi_i \,,$$

where each ϕ_i is in unary strong normal form. A formula ϕ is in unary strong normal form if it is $\top$ or it has the form

$$\phi = \bigwedge_{j \in J} \langle a_j \rangle \phi_j \wedge \bigwedge_{b \in A^l} [b] \psi_b,$$

where every ϕ_j is in unary strong normal form and every ψ_b is in strong normal form.

We note that any unary strong normal form different from $\top$ can equivalently be written as

$$\phi = \bigwedge_{j \in J} \langle a_j \rangle \phi_j \wedge \bigwedge_{b \in A^l} [b] \bigvee_{k \in K_b} \psi^k_b,$$

where every ϕ_j and every ψ^k_b are in unary strong normal form, thus avoiding the introduction of strong normal forms.

Remark 1 *It is not hard to see that each unary strong normal form is consistent. See also Theorem 2 to follow.*

Clearly the characteristic formulae of processes are in unary strong normal form. Therefore, by Lemma 3, it is a necessary condition for a formula to be representable by a single process that it has an equivalent unary strong normal form. We will show that this is also a sufficient condition for this to hold for any consistent formula.

Theorem 2 *A unary strong normal form*

$$\phi = \bigwedge_{j \in J} \langle a_j \rangle \phi_j \wedge \bigwedge_{b \in A^l} [b] \bigvee_{k \in K_b} \psi^k_b$$

is represented by the process defined recursively by

$$\begin{aligned}\theta(\phi) &= \sum_{j\in J} a_j.\theta(\phi_j) + \sum_{b\in A^l}\sum_{k\in K_b} b.\theta(\psi_b^k), \quad \text{if } \phi \neq \top \\ \theta(\top) &= \omega.\end{aligned}$$

In particular ϕ is the characteristic formula for $\theta(\phi)$ (up to logical equivalence). Note that. even if in the formal expression above there is a summand for each $b \in A^l$, only those b's such that $K_b \neq \emptyset$ will finally appear as summands of $\theta(\phi)$.

Proof. First we prove that $\theta(\phi) \models \phi$ by induction on the modal depth of ϕ. If $\phi = \top$ we have that obviously $\theta(\phi) = \omega \models \phi = \top$. For the inductive step first we note that $\theta(\phi) \xrightarrow{a_j} \theta(\phi_j)$ for all $j \in J$. By induction, $\theta(\phi_i) \models \phi_i$. Next assume that $\theta(\phi) \xrightarrow{b} p$ for some $b \in A^l$ and some p. We have that $p = \theta(\psi_b^k)$ for some $k \in K_b$. By induction $\theta(\psi_b^k) \models \psi_b^k$ and therefore $\theta(\psi_b^k) \models \bigvee_{k\in K_b} \psi_b^k$.

Next we prove that if $q \models \phi$ then $\theta(\phi) \lesssim_{cc} q$. Towards proving this claim, assume that $q \models \phi$. Again we proceed by induction on the modal depth of ϕ.

First assume that $\theta(\phi) \xrightarrow{a} p'$ for some $a \in A^r$ and process term p'. Then $a = a_j$ for some $j \in J$ and $p' = \theta(\phi_j)$. As $q \models \phi$, we have that $q \xrightarrow{a_j} q'$ for some q' with $q' \models \phi_j$. By induction, $\theta(\phi_j) \lesssim_{cc} q'$, as required.

Now assume that $q \xrightarrow{b} q'$ for some $b \in A^l$. As $q \models \phi$ we have that $q' \models \psi_b^k$ for some $k \in K$. Now $\theta(\phi) \xrightarrow{b} \theta(\psi_b^k)$ and, by the induction hypothesis, we have $\theta(\psi_b^k) \lesssim_{cc} q'$, as required.

This proves that ϕ is the characteristic formula for $\theta(\phi)$ and therefore, by Lemma 3, that $\theta(\phi)$ represents ϕ. □

Next, we will show that any formula has an equivalent strong normal form and therefore can always be represented by a (possibly empty) finite set of processes. To derive this result we will use several standard equivalences between formulae.

Lemma 4 *The following statements hold.*

1. *$\wedge$ and $\vee$ are associative, commutative and idempotent.*
2. *$\wedge$ distributes over $\vee$, and $\vee$ distributes over $\wedge$.*
3. *$\phi \vee \top \equiv \top$, $\phi \vee \bot \equiv \phi$, $\phi \wedge \top \equiv \phi$, and $\phi \wedge \bot \equiv \bot$.*
4. *$[b]\top \equiv \top$.*
5. *$[b]\phi \wedge [b]\psi \equiv [b](\phi \wedge \psi)$ for $b \in A^l$.*
6. *$\langle a\rangle\phi \vee \langle a\rangle\psi \equiv \langle a\rangle(\phi \vee \psi)$ for $a \in A^r$.*

Proof. The first three collections of equalities are straightforward and well known, so we omit their proofs.

- $[b]\top \equiv \top$. We have $p \models [b]\top$ iff $p' \models \top$ for all $p \xrightarrow{b} p'$. Therefore, the condition is satisfied whenever $p \xrightarrow{b} p'$, and it is vacuously true when $p \not\xrightarrow{b}$.
- $[b]\phi \wedge [b]\psi \equiv [b](\phi \wedge \psi)$. We have $p \models ([b]\phi \wedge [b]\psi)$ iff $p' \models \phi$ for all $p \xrightarrow{b} p'$ and $p' \models \psi$ for all $p \xrightarrow{b} p'$, iff $p' \models (\phi \wedge \psi)$ for all $p \xrightarrow{b} p'$, iff $p \models [b](\phi \wedge \psi)$.

- $\langle a\rangle\phi \vee \langle a\rangle\psi \equiv \langle a\rangle(\phi \vee \psi)$. We have $p \models \langle a\rangle\phi \vee \langle a\rangle\psi$ iff there exists $p \xrightarrow{a} p'$ such that $p' \models \phi$ or there exists $p \xrightarrow{a} p''$ such that $p'' \models \psi$, that is, iff there exists some $p \xrightarrow{a} p'_0$ such that $p'_0 \models \phi$ or $p'_0 \models \psi$. This holds iff $p \models \langle a\rangle(\phi \vee \psi)$. □

Lemma 5 *Every formula ϕ has an equivalent strong normal form with no larger modal depth.*

Proof. First we prove by induction on the modal depth, using 1-3 of Lemma 4, that ϕ has an equivalent normal form with the same modal depth. To prove the main statement we can therefore assume that ϕ is in normal form. We proceed by induction on the modal depth $md(\phi)$. The base case $md(\phi) = 0$ ($\phi \equiv \bot$ and $\phi \equiv \top$) follows immediately.

Next let us assume that

$$\phi = \bigvee_{i\in I}\Big(\bigwedge_{j\in J_i}\langle a^i_j\rangle\phi^i_j \wedge \bigwedge_{k\in K_i}[b^i_k]\psi^i_k\Big).$$

By Lemma 4, using 4 and 5 and the standard laws described in 1-3, ϕ can be rewritten into an equivalent formula of the form

$$\phi = \bigvee_{i\in I}\Big(\bigwedge_{j\in J_i}\langle a^i_j\rangle\phi^i_j \wedge \bigwedge_{b\in A^l}[b]\psi^i_b\Big)$$

where $md(\psi^i_b) \leq \sup\{md(\psi^i_k) \mid k \in K_i\}$ (we note that some of the $[b]\psi^i_b$s may have the form $[b]\top$, which is equivalent to $\top$). Therefore, by the induction hypothesis, we may assume that ϕ^i_j and ψ^i_b are in strong normal form. Next we use Lemma 4.6 to remove all the occurrences of $\vee$ that are guarded by $\langle a\rangle$, for some $a \in A^r$ in each $\bigwedge_{j\in J_i}\langle a^i_j\rangle\phi^i_j$. The result for each i is of the form $\bigwedge_{j\in J_i}(\bigvee_{l\in L_j}\langle a^i_j\rangle\phi^{l,i}_j)$, where each $\phi^{l,i}_j$ is in a unary strong normal form. By repeated use of distributivity, the whole formula can be rewritten as

$$\phi = \bigvee_{r\in R}\Big(\bigwedge_{s\in S_r}\langle a^r_s\rangle\alpha^r_s \wedge \bigwedge_{b\in A^l}[b]\bigvee_{t\in T^r_b}\beta^r_{b,t}\Big)$$

where each α^s_r and $\beta^{b,t}_r$ is a unary strong normal form. Finally we note that the operations described above do not increase the modal depth. □

Now we will relate our result to the one in Boudol and Larsen's paper [4].

Definition 8 *A formula ϕ is prime if the following holds:*

$$\forall\phi_1,\phi_2 \in \mathscr{L}.\ \phi \leq \phi_1 \vee \phi_2 \text{ implies } \phi \leq \phi_1 \text{ or } \phi \leq \phi_2.$$

Theorem 3 *A formula ϕ can always be represented by a finite set of processes. It can be represented by a single process if and only if it is consistent and prime.*

Proof. By Lemma 5, $\phi \equiv \phi_1 \vee \ldots \vee \phi_n$ where each ϕ_i, $1 \leq i \leq n$, is in unary strong normal form. By Theorem 2, $\phi_i \equiv \chi(p_i)$ for some p_i for each $1 \leq i \leq n$, and therefore $\phi \equiv \chi(p_1) \vee \ldots \vee \chi(p_n)$. The first statement now follows from Lemma 3.2.

Towards proving the second statement, first assume that $\phi \equiv \chi(p_1) \vee \ldots \vee \chi(p_n)$ is prime. This implies that $\phi \leq \chi(p_i) \leq \phi$, for some $i \in \{1,\ldots,n\}$, which in turn implies that $\phi \equiv \chi(p_i)$.

Next assume that ϕ is represented by some process p or equivalently that $\phi \equiv \chi(p)$. Now assume that $\chi(p) \leq \phi_1 \vee \phi_2$. As $p \models \chi(p)$, this implies that $p \models \phi_1 \vee \phi_2$ or equivalently that either $p \models \phi_1$ or $p \models \phi_1$. Without loss of generality, we can assume that $p \models \phi_1$. Now assume that $r \models \chi(p)$. Then $p \lesssim_{cc} r$ and by Theorem 1 this implies that $r \models \phi_1$. Since r was arbitrary, this proves that $\phi \equiv \chi(p) \leq \phi_1$. Hence ϕ is prime, which was to be shown. □

5 Considering bivariant actions

Originally [5, 6, 7], the theory of covariant-contravariant semantics also considered bivariant actions in A^{bi}, so that we had a partition of A into $\{A^r, A^l, A^{bi}\}$ (called the signature of the LTS), and the definition of covariant-contravariant simulations imposed the following two conditions:

- For all $a \in A^r \cup A^{bi}$ and all $p \xrightarrow{a} p'$, there exists some $q \xrightarrow{a} q'$ with $p' \, R \, q'$.
- For all $a \in A^l \cup A^{bi}$ and all $q \xrightarrow{a} q'$, there exists some $p \xrightarrow{a} p'$ with $p' \, R \, q'$.

In [1] we presented transformations from LTSs to Modal Transition Systems (MTSs), and vice versa. Applying these two transformations in a row we did not obtain the identity function, but instead we could transform an LTS with bivariant actions into another (equivalent) LTS without them. To be precise, each $a \in A^{bi}$ is encoded by means of a pair of new actions (a^r, a^l) with $a^r \in A^r$ and $a^l \in A^l$. That transformation preserves and reflects the logic and the covariant-contravariant simulation preorder. However, as a consequence of the general mechanism to transform an LTS into an equivalent MTS, an additional $a^l \in A^l$ is introduced for each $a \in A^r$. Nevertheless, the addition of these new actions does not change the behaviour of the system, because we also add an "extra" state u whose behaviour is that defined by ω (as can be seen in Figure 1).

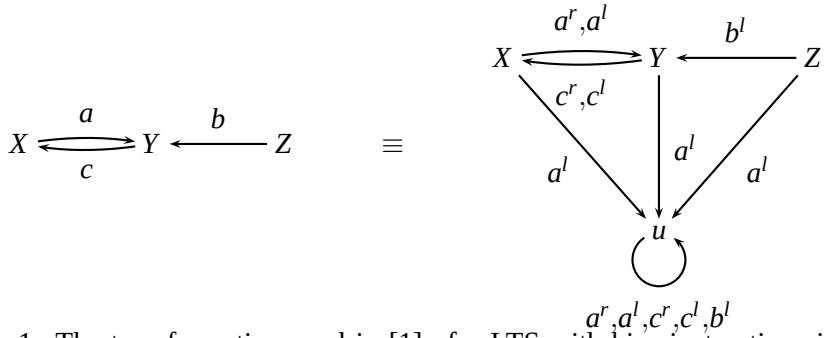


Figure 1: The transformation used in [1] of a LTS with bivariant actions into an equivalent one without them. $A^r = \{a\}$, $A^l = \{b\}$ and $A^{bi} = \{c\}$.

Based on this transformation, we have designed a direct encoding of LTSs over a set of actions A partitioned into $\{A^r, A^l, A^{bi}\}$ by means of an LTS over some set of actions $\overline{A}$ partitioned into $\{\overline{A}^r, \overline{A}^l, \emptyset\}$. For each bivariant action a in the signature of the original LTS, we introduce a pair of (new) actions, as the next definition makes precise.

Definition 9 *Let T be an LTS with signature $\{A^r, A^l, A^{bi}\}$. The LTS $\mathscr{T}(T)$, with signature $\{\overline{A}^r, \overline{A}^l\}$, where $\overline{A}^r = A^r \cup \{a^r \mid a \in A^{bi}\}$, $\overline{A}^l = A^l \cup \{a^l \mid a \in A^{bi}\}$, is constructed as follows:*

- *The set of states of $\mathscr{T}(T)$ is the same as that of T.*
- *All the transitions from T with label in $A^r \cup A^l$ are in $\mathscr{T}(T)$.*
- *For each transition $p \xrightarrow{a} p'$ in T with $a \in A^{bi}$, we add $p \xrightarrow{a^r} p'$ and $p \xrightarrow{a^l} p'$ to $\mathscr{T}(T)$.*

The transformation above produces LTSs without bivariant actions that faithfully represent any covariant-contravariant LTS (compare Figure 2 with Figure 1). Note that any LTS $S = (\mathbf{P}, \overline{A}^l \uplus \overline{A}^r, \longrightarrow)$ that is the representation of an LTS with signature $\{A^r, A^l, A^{bi}\}$ satisfies that $p \xrightarrow{a^r} p'$ iff $p \xrightarrow{a^l} p'$ for all $p, p' \in \mathbf{P}$ and all $a \in A^{bi}$.

For the case of modal formulae we have just to consider the right modality for the action as the following definition makes precise:

Definition 10 *Let us extend $\mathscr{T}$ to translate modal formulae over the modal logic for LTS over A into modal formulae over the modal logic for LTS over $\bar{A}$:*

- $\mathscr{T}(\bot) = \bot$.
- $\mathscr{T}(\top) = \top$.
- $\mathscr{T}(\varphi \wedge \psi) = \mathscr{T}(\varphi) \wedge \mathscr{T}(\psi)$.
- $\mathscr{T}(\varphi \vee \psi) = \mathscr{T}(\varphi) \vee \mathscr{T}(\psi)$.
- $\mathscr{T}(\langle a \rangle \varphi) = \langle a \rangle \mathscr{T}(\varphi)$, *if* $a \in A^r$.
- $\mathscr{T}(\langle a \rangle \varphi) = \langle a^r \rangle \mathscr{T}(\varphi)$, *if* $a \in A^{bi}$.
- $\mathscr{T}([b]\varphi) = [b]\mathscr{T}(\varphi)$, *if* $b \in A^l$.
- $\mathscr{T}([a]\varphi) = [a^l]\mathscr{T}(\varphi)$, *if* $a \in A^{bi}$.

Thus, we can apply all the results in Section 3 to the obtained systems and formulae, and in this way we (indirectly) apply all our results to any alphabet A, possibly including bivariant actions. Next we show that the transformation above reflects and preserves covariant-contravariant simulations.

Figure 2: The new transformation of a LTS with bivariant actions into an equivalent one without them. $A^r = \{a\}$, $A^l = \{b\}$ and $A^{bi} = \{c\}$.

Lemma 6 *$p \lesssim_{cc} q$ in T if and only if $p \lesssim_{cc} q$ in $\mathscr{T}(T)$.*

Proof. Let us first suppose that $p \lesssim_{cc} q$ in T. Let $p \xrightarrow{x} p'$ in $\mathscr{T}(T)$, with $x \in \bar{A}^r$. If $x = a \in A^r$ we are done, whereas if $x = a^r$, with $a \in A^{bi}$, there exists some q' such that $q \xrightarrow{a} q'$. Now, since $x = a^r \in \bar{A}^r$ we obtain that $q \xrightarrow{x} q'$. The case when $x \in \bar{A}^l$ and $q \xrightarrow{x} q'$ is analogous.

Now, let us suppose that $p \lesssim_{cc} q$ in $\mathscr{T}(T)$ and let $p \xrightarrow{a} p'$ with $a \in A^r \cup A^{bi}$ in $\mathscr{T}(T)$. The case for $a \in A^r$ is straightforward. If $a \in A^{bi}$ then we have $a^r \in \bar{A}^r$ and $p \xrightarrow{a^r} p'$ in $\mathscr{T}$. Therefore, there is some q' such that $q \xrightarrow{a^r} q'$ in $\mathscr{T}(T)$, with $p' \lesssim_{cc} q'$; from here it follows that $q \xrightarrow{a} q'$ in T. The case when $b \in A^l \cup A^{bi}$ and $q \xrightarrow{b} q'$ is analogous. □

After the representation of a bivariant action $a \in A^{bi}$ as a pair (a^r, a^l), we have that a^l under-approximates a, whereas a^r over-approximates a. This means in particular that we have $a^l \lesssim_{cc} a^l + a^r \lesssim_{cc} a^r$. From here we obtain the needed properties for the graphical representation results in the previous section.

To conclude the section we explore the set of systems for a signature $\bar{A}$. Some of them, but not all, are equivalent to the representation of a system for the original alphabet A. Whenever that is not the case we would need to remove (or add) some transitions labelled by the created actions in $\{a^r, a^l \mid a \in A^{bi}\}$ in order to obtain a system that is equivalent to the representation of some process. In the following

proposition we give an algorithm for obtaining a system for the original signature A to which a given system for the signature $\bar{A}$ is equivalent, whenever such a system exists. To make possible a proof by (structural) induction, we will only present the result for process terms in $\mathscr{P}$.

Proposition 1 *Let $A = \{A^r, A^l, A^{bi}\}$ signature and $\bar{A} = \{\bar{A}^r, \bar{A}^l\}$ the associated signature without bivariant actions. Let $p, q \in \mathscr{P}$ be processes terms for $\bar{A}$ such that q is the representation of some process for the signature A. Let us assume that $p \equiv cc_{cc} q$. Then it is possible to transform p into the representation p_{bi} of some process term for A, simply by adding or removing transitions in $\{a^r, a^l \mid a \in A^{bi}\}$ as indicated in the following proof.*

Proof. The idea is to consider each state p_s of the LTS that defines the semantics of p, and any transition $p_s \xrightarrow{b^r} p'_s$ (resp. $p_s \xrightarrow{b^l} p'_s$) has not a companion $p_s \xrightarrow{b^l} p'_s$ (resp. $p_s \xrightarrow{b^r} p'_s$), then we see if we can simply remove the unmatched transition or we should instead add its companion.

The proof is done by structural induction.

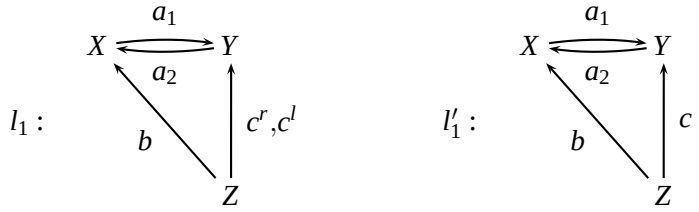
- If $p = 0$ or $p = \omega$ then we take $p_{bi} = p$.
- Let $p = p' + b^r p'_r$, where the term $b^r p'_r$ is not matched by any $b^l p'_r$ (the symmetric case $p = p' + b^l p'_l$ is analogous). Then, we check if $p' \equiv_{cc} p$. If that is the case, we simply remove the summand $b^r p'_r$. Otherwise, we can add the summand $b^l p'_r$ obtaining an equivalent term p_{bi}. This is indeed the case because, if $p \not\equiv_{cc} p'$ this means that the summand $b^r p'_r$ is maximal with respect to $\lesssim_{cc}$ on the set of summands $b^r p''_r$ of p. Then, from $p \lesssim_{cc} q$ we obtain that $p \xrightarrow{b^r} p'_r$ implies $q \xrightarrow{b^r} q'_r$ and $p'_r \lesssim_{cc} q'_r$. But, we have also $q \lesssim_{cc} p$, and then $q \xrightarrow{b^r} q'_r$ implies $p \xrightarrow{b^r} p'_r$ and $q'_r \lesssim_{cc} p'_r$, due to the maximality of the term $b^r p'_r$. Therefore, $p'_r \equiv_{cc} q'_r$. Since we also have that $q \xrightarrow{b^l} q'_r$, we conclude that if we take $p_{bi} = p + b^l p'_r$ we obtain $p_{bi} \equiv_{cc} q$, and therefore, $p \equiv_{cc} p_{bi}$.

  Hence, we can assume that any summand $b^r p'_r$ of p has a companion $b^l p'_r$. Then, we would be done if we could apply the induction hypothesis to each derivative of p. However, this cannot be (always) done, even for summands $a^r p'$ or $a^l p'$ with $a^r \in A^r$ and $a^l \in A^l$. The reason is that we cannot assert that any such derivative is equivalent to the representation of a term q'. Indeed, for any summand $a^r p'$ we should have $q \xrightarrow{a^r} q'_r$ with $p' \lesssim_{cc} q'_r$. Hence, from $q \xrightarrow{a^r} q'_r$ and $q \lesssim_{cc} p$, we have $p \xrightarrow{a^r} p''$, with $p' \lesssim_{cc} p''$. This means that if we start with a maximal summand of p, then we should have $p' = p''$, and we could apply the induction hypothesis. For the summands $a^r p'$ that are not maximal we can simply remove them in a similar way as we did before when we had $p \equiv_{cc} p'$. Of course, the treatment of the $a^l p'$ summands is dual, and in this case we can remove all the summands that are not minimal with respect to $\lesssim_{cc}$, applying the induction hypothesis to the others.

  Finally, we have also to take care of the $b^r p'_r$ and $b^l p'_r$ summands (Note that after the "balancing" step before, we can assume that the continuations after the b^r and the b^l summands are the same). In order to obtain an equivalent term we could apply to each summand $b^r p'_r$ the same reasoning that we applied to the summands $a^r p_r$, but in such a case we run the risk of removing one of the summands (e.g. $b^r p'_r$) leaving the other (e.g. $b^l p'_r$), so that we have ruined the balancing work in the step before. Instead, we will leave all the summands $b^r p'_r$ such that p'_r is either minimal or maximal with respect to $\lesssim_{cc}$, and the same for the summands $b^l p'_r$, and we remove all the intermediate summands. This way, we maintain the symmetry between between $b^r p'_r$ and $b^l p'_r$ summands, and for each of the remaining summands we can indeed apply the induction hypothesis, completing the proof. This is so, because depending on the maximal or minimal character of the summand,

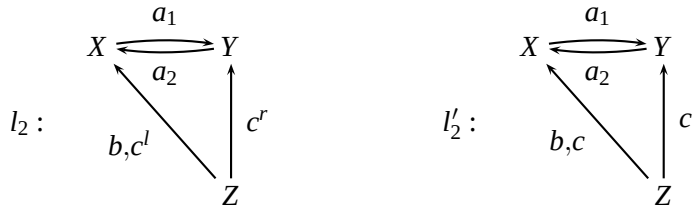
we can either apply the reasoning above for the maximal $a^r p'_r$ summands or that for the minimal $a^l p'_r$ summands. □

Example 1 *Let* $\bar{A}^r = \{a_1, a_2, c^r\}$ *and* $\bar{A}^l = \{b, c^l\}$.



We have that the LTS l_1 *is the representation of the LTS* l'_1*, where* $A^r = \{a_1, a_2\}$, $A^l = \{b\}$ *and* $A^{bi} = \{c\}$.

Instead,



the only possible system that l_2 *could represent is* l'_2*, but, since obviously* $l_2 \not\equiv_{cc} l'_2$*, the algorithm tells us that* l_2 *is not equivalent to the representation of any LTS.*

6 Conclusions and future work

In [1] we studied the relationships between the notion of refinement over modal transition systems, and the notions of covariant-contravariant simulation and partial bisimulation over labelled transition systems. Here we have continued that work by looking for the "graphical" representation of the covariant-contravariant modal formulae by means of terms, as it was done in [3] for the case of modal transition systems. For technical reasons, we had first to restrict ourselves to the case in which we have no bivariant actions. Afterwards, we argued that the general case can, in some sense, be "reduced" to the one we dealt with in Section 4 by defining a semantic-preserving transformation between covariant-contravariant systems with bivariant actions, and covariant-contravariant systems without them.

The idea was to separate each bivariant action into its covariant and its contravariant parts. As a matter of fact, we believe that this idea might be useful not only for obtaining theoretical results, as we have done here, but also for applications. Most of the studies on process algebras and their semantics assume the bivariant behaviour of all the actions. It is true that in some studies (see for example [12]) we have a classification of actions, as we have also done in [1] and in this paper. But now we are proposing to exploit the relationships between the different classes of actions.

As future work, it would be interesting to obtain a direct characterization of the formulae that are graphically representable in a setting with bivariant actions. Such a direct characterization will also pave the way towards a more general theory of "graphical characterizations" of formulae in modal logics of processes, of which the result by Boudol and Larsen and ours are special cases.

Of course, one of the directions in which we plan to continue our studies is that related with the logical characterization of the semantics, and in particular the connections between logical formulae and

terms established by characteristic formulae and graphical representations. The combination of these two frameworks is also an interesting challenge. In particular, we plan some extensions of the recent work by Lüttgen and Vogler [10, 11] to the case of covariant-contravariant systems.

References

[1] L. Aceto, I. Fábregas, D. de Frutos Escrig, A. Ingólfsdóttir, and M. Palomino. Relating modal refinements, covariant-contravariant simulations and partial bisimulations. In F. Arbab and M. Sirjani, editors, *Fundamentals of Software Engineering, FSEN 2011*, Lecture Notes in Computer Science. Springer, 2011.

[2] L. Aceto, A. Ingólfsdóttir, K. G. Larsen, and J. Srba. *Reactive Systems: Modelling, Specification and Verification*. Cambridge University Press, 2007.

[3] J. Baeten, D. van Beek, B. Luttik, J. Markovski, and J. Rooda. Partial bisimulation. SE Report 2010-04, Systems Engineering Group, Department of Mechanical Engineering, Eindhoven University of Technology, 2010.

[4] G. Boudol and K. G. Larsen. Graphical versus logical specifications. *Theoretical Computer Science*, 106(1):3–20, 1992.

[5] I. Fábregas, D. de Frutos-Escrig, and M. Palomino. Non-strongly stable orders also define interesting simulation relations. In A. Kurz, M. Lenisa, and A. Tarlecki, editors, *CALCO*, volume 5728 of *Lecture Notes in Computer Science*, pages 221–235. Springer, 2009.

[6] I. Fábregas, D. de Frutos-Escrig, and M. Palomino. Equational characterization of covariant-contravariant simulation and conformance simulation semantics. In L. Aceto and P. Sobociński, editors, *Proceedings Seventh Workshop on Structural Operational Semantics, Paris, France, 30 August 2010*, volume 32 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–14, 2010.

[7] I. Fábregas, D. de Frutos-Escrig, and M. Palomino. Logics for contravariant simulations. In J. Hatcliff and E. Zucca, editors, *Formal Techniques for Distributed Systems, Joint 12th IFIP WG 6.1 International Conference, FMOODS 2010 and 30th IFIP WG 6.1 International Conference, FORTE 2010, Amsterdam, The Netherlands, June 7-9, 2010. Proceedings*, volume 6117 of *Lecture Notes in Computer Science*, pages 224–231. Springer, 2010.

[8] K. G. Larsen. Modal specifications. pages 232–246, 1989.

[9] K. G. Larsen and B. Thomsen. A modal process logic. In *LICS*, pages 203–210. IEEE Computer Society, 1988.

[10] G. Lüttgen and W. Vogler. Safe reasoning with logic lts. In M. Nielsen, A. Kucera, P. B. Miltersen, C. Palamidessi, P. Tuma, and F. D. Valencia, editors, *SOFSEM*, volume 5404 of *Lecture Notes in Computer Science*, pages 376–387. Springer, 2009.

[11] G. Lüttgen and W. Vogler. Ready simulation for concurrency: It's logical! *Inf. Comput.*, 208(7):845–867, 2010.

[12] N. Lynch. I/O automata: A model for discrete event systems. *In 22nd Annual Conference on Information Sciences and Systems*, pages 29–38, 1988.

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

[14] D. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Theoretical Computer Science, 5th GI-Conference, Karlsruhe, Germany, March 23-25, 1981, Proceedings*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer, 1981.

[15] R. J. van Glabbeek. The linear time-branching time spectrum I: The semantics of concrete, sequential processes. In J. A. Bergstra, A. Ponse, and S. A. Smolka, editors, *Handbook of process algebra*, pages 3–99. North-Holland, 2001.