Forward and Backward Simulations
Part I: Untimed Systems

Nancy Lynch
MIT Laboratory for Computer Science
Cambridge, MA 02139, USA
lynch@theory.lcs.mit.edu

Frits Vaandrager
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands
fritsv@cwi.nl
University of Amsterdam, Programming Research Group
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

August 19, 1994

Abstract

A comprehensive presentation of simulation techniques is given in terms of a simple (untimed) automaton model. In particular, we discuss (1) refinements, (2) forward and backward simulations, (3) forward-backward and backward-forward simulations, and (4) history and prophecy relations. History and prophecy relations are new and are abstractions of the history and prophecy variables of Abadi and Lamport, as well as the auxiliary variables of Owicki and Gries. We give elegant and short proofs of soundness and completeness results for complicated simulations in terms of soundness and (partial) completeness results for simple simulations. In Part II of this paper, it is shown how most of the results for untimed automata can be carried over smoothly to the setting of timed automata.

1991 Mathematics Subject Classification: 68Q60, 68Q68.

1991 CR Categories: F.1.1, F.3.1.

Keywords and Phrases: Simulations, automata, refinement mappings, forward simulations, backward simulations, forward-backward simulations, backward-forward simulations, history variables, prophecy variables, history relations, prophecy relations.

Notes: This work was supported by ONR contracts N00014-85-K-0168 and N00014-91-J-1988, by NSF grant CCR-8915206, and by DARPA contracts N00014-89-J-1988 and N00014-92-J-4033. Part of this work took place while the second author was employed by the Ecole des Mines, CMA, Sophia Antipolis, France. The second author also received partial support from the ESPRIT Basic Research Action 7166, CONCUR2. An earlier verion of this paper (Part I+II) appeared as [22].

1 Introduction

In this paper, we give a comprehensive presentation of forward and backward simulation methods for proving trace inclusion relationships between concurrent systems. The concurrent systems we treat in this paper do not involve timing; there is a second paper, [23], following this one in this same journal issue, which extends the ideas of this paper to timing-based systems.

We present all the methods in terms of a simple and general automaton model, which includes internal actions. We define several kinds of simulations including *refinements*, *forward simulations*, *backward simulations*, and hybrid versions that we call *forward-backward* and *backward-forward simulations*. We prove basic results for these kinds of simulations, in particular, soundness and completeness theorems. We also define *history relations* and *prophecy relations*, which are abstract versions of the history and prophecy variables, respectively, of Abadi and Lamport [1]; history relations are also abstract versions of the auxiliary variables of Owicki and Gries [26]. We prove theorems describing the properties of these various kinds of simulations and relating the different kinds of simulations to each other.

The simulations we consider are derived from simulations studied in many places in the research literature. The simplest kind of simulation we consider is a *refinement*, which is a functional simulation similar to those studied in [16] and very similar to a homomorphism between automata in the sense of classical automata theory [4]. A refinement from an automaton A to another automaton B is a function from states of A to states of B such that (a) the image of every start state of A is a start state of B, and (b) every step of A has a corresponding sequence of steps of B that begins and ends with the images of the respective beginning and ending states of the given step, and that has the same external actions. This notion of refinement implies that the traces of A are also traces of B. We give soundness and partial completeness results.

We then consider *forward simulations* and *backward simulations*, which are generalizations of refinements that allow a set of states of B to correspond to a single state of A. Forward simulations are similar to the the simulations of [27, 8], the possibilities mappings of [19, 21], the downward simulations of [7, 12, 5], the forward simulations of [11], and the history measures of [14]. The correspondence conditions (a) and (b) above are generalized so that (a) every start state of A has *some* image that is a start state of B, and (b) every step of A and every state of B corresponding to the *beginning* state of the step yield a corresponding sequence of steps of B ending with the image of the *ending* state of the given step. Again, we give soundness and partial completeness results.

Backward simulations occurred first in [7] under the name of upward simulations and were used later in the setting of CSP in [12, 5]. In [24] and [10], where they are called prophecy mappings and backwards simulations, respectively, it is observed that they are closely related to the prophecy variables first defined in [1]. In the case of a backward simulation, conditions (a) and (b) are generalized so that (a) *all* images of every start state of A are start states of B, and (b) every step of A and every state of B corresponding to the *ending* state of the step yield a corresponding sequence of steps of B *beginning* with the image of the beginning state of the given step. Again, we give soundness and partial completeness results.

We also consider *forward-backward* and *backward-forward simulations*, which are essentially compositions of one forward and one backward simulation, in the two possible orders. The definition of a forward-backward simulation has been inspired by the work of Klarlund and Schneider [13, 14], for the case without internal actions. The notion of a backward-forward simulation is suggested by symmetry with forward-backward simulations. While some of the results for this case are symmetric with the forward-backward case, others (notably, certain completeness results) do not hold.

We also provide redefinitions of the *history variable* and *prophecy variable* notions of [1], and generalize these to new notions of *history relation* and *prophecy relation*. We prove equivalence between these definitions and our notions of forward and backward simulations. Finally, we show how reachability can be integrated into the various simulation proof methods.

The usefulness of refinement mappings and forward simulations in proving correctness has been well demonstrated. Abstraction mappings, which are essentially refinement mappings, comprise a basic proof method for implementations of abstract data types [6, 18]. Typical examples of forward simulation proofs appear in [20]. Backward simulations have been much less widely used. Abadi and Lamport [1] demonstrate the usefulness of prophecy variables (and hence backward simulations), with some simple examples, while [17] contains a more interesting example. There has not been much work on applying the hybrid forward and backward methods.

As far as the classification of simulations is concerned, our work is closely related to and extends that of Jonsson [11]. Jonsson, however, addresses liveness issues, which we do not do. Also, Jonsson has more powerful notion of backward simulation, which we prefer not to use since it fails to reduce global reasoning about infinite behaviors to local reasoning about states and actions.

We consider the main contributions of this paper to be the following. First, we give a comprehensive presentation, in terms of a very simple and abstract automaton model, of a wide range of important simulation techniques, together with their basic soundness and completeness properties. We present the various simulation techniques in a "bottom-up" order, starting with simple ones such as forward and backward simulations and building up to more complicated simulations such as forward-backward simulations and history relations. We give elegant and short proofs of soundness and completeness results for complicated simulations in terms of soundness and (partial) completeness results for simple simulations. We show how to incorporate invariant assertions into the simulations. Second, there are several specific new definitions and results, notably: (1) The definition of a notion of composition of forward-backward simulations. This allows us to prove that image-finite forward-backward simulations induce a preorder on the domain of general automata. (2) The introduction of backward-forward simulations. Although these simulations do not lead to a complete proof method, they are sound and possibly useful in practice. They arise naturally as the dual notion of forward-backward simulations. (3) The notions of history and prophecy relations.

In Part II [23], we extend the results of this paper to timing-based systems. We do this by defining a new notion of automaton called a *timed automaton*, and using it to present all the definitions and results for timed automata. The results for the timed setting turn out to be analogous to those for the untimed setting. In most cases, our results for the timed setting are derived from those for the untimed setting, while in the remaining cases, new proofs analogous to those in this paper are presented.

The rest of this paper is organized as follows. Section 2 contains some mathematical preliminaries. Section 3 contains basic definitions and results for untimed automata. Section 4 contains the development of the basic simulation techniques: refinements, forward simulations and backward simulations. Section 5 contains the development of the hybrid techniques: forward-backward and backward-forward simulations. Section 6 contains the results on history and prophecy relations. Section 7 shows how reachability can be included in the simulations. Finally, Section 8 contains some conclusions.

2 Preliminaries

We begin with some basic mathematical preliminaries.

2.1 Sequences

Let K be any set. The sets of finite and infinite sequences of elements of K are denoted by K^* and K^ω, respectively. Concatenation of a finite sequence with a finite or infinite sequence is denoted by juxtaposition; λ denotes the empty sequence and the sequence containing one element $a \in K$ is denoted a. We say that a sequence σ is a *prefix* of a sequence ρ, notation $\sigma \leq \rho$, if either $\sigma = \rho$, or σ is finite and $\rho = \sigma\sigma'$ for some sequence σ'. A set S of sequences is *prefix closed* if, whenever some sequence is in S, all its prefixes are also. If σ is a nonempty sequence then $\mathit{first}(\sigma)$ returns the first element of σ, and $\mathit{tail}(\sigma)$ returns σ with its first element removed. Moreover, if σ is finite, then $\mathit{last}(\sigma)$ returns the last element of σ. If σ is a sequence over K and $L \subseteq K$, then $\sigma\lceil L$ denotes the sequence obtained by projecting σ on L. If S is a set of sequences, $S\lceil L$ is defined as $\{\sigma\lceil L \mid \sigma \in S\}$.

2.2 Sets, Relations and Functions

A *relation* over sets X and Y is defined to be any subset of $X \times Y$. If f is a relation over X and Y, then we define the *domain* of f to be $\mathit{domain}(f) \triangleq \{x \in X \mid (x, y) \in f \text{ for some } y \in Y\}$, and the *range* of f to be $\mathit{range}(f) \triangleq \{y \in Y \mid (x, y) \in f \text{ for some } x \in X\}$. A relation f over X and Y is *total* over X if $\mathit{domain}(f) = X$. If X is any set, we let $\mathit{id}(X)$ denote the identity relation over X and X, i.e., $\{(x, x) \mid x \in X\}$. We define *composition* of relations in the usual way, i.e., if f and g are relations over X and Y and over Y and Z, respectively, then $g \circ f$ denotes the relation over X and Z consisting of all pairs (x, z) such that there exists $y \in Y$ with $(x, y) \in f$ and $(y, z) \in g$. For all relations f, g and h, $f \circ (g \circ h) = (f \circ g) \circ h$. Also, for $X \supseteq \mathit{domain}(f)$ and $Y \supseteq \mathit{range}(f)$, $\mathit{id}(X) \circ f = f \circ \mathit{id}(Y) = f$. If f is a relation over X and Y, then the *inverse* of f, written f^{-1}, is defined to be the relation over Y and X consisting of those pairs (y, x) such that $(x, y) \in f$. Recall that for any pair of relations f and g, $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$. If f is a relation over X and Y, and Z is a set, then $f\lceil Z$ is the relation over $X \cap Z$ and Y given by $f\lceil Z \triangleq f \cap (Z \times Y)$. If f is a relation over X and Y and $x \in X$, we define $f[x] = \{y \in Y \mid (x, y) \in f\}$. We say that a relation f over X and Y is a *function from X to Y*, and write $f : X \to Y$, if $|f[x]| = 1$ for all $x \in X$; in this case, we write $f(x)$ to denote the unique element of $f[x]$. A function c from X to Y is a *choice* *function* for a relation f over X to Y provided that $c \subseteq f$ (i.e., $c(x) \in f[x]$ for all $x \in X$). If X is a set, $\mathbf{P}(X)$ denotes the powerset of X, i.e., the set of subsets of X, and $\mathbf{N}(X)$ the set of nonempty subsets of X, i.e., the set $\mathbf{P}(X) - \{\emptyset\}$. We say that a relation f over X and Y is *image-finite* if $f[x]$ is finite for all x in X. If f is a relation over X and $\mathbf{P}(Y)$, then we say that f is *image-set-finite* if every set in the range of f is finite.

2.3 A Basic Graph Lemma

We require the following lemma, a generalization of König's Lemma [15]. If G is a digraph, then a *root* of G is defined to be a node with no incoming edges.

Lemma 2.1 *Let G be an infinite digraph that satisfies the following properties.*

1. *G has finitely many roots.*
2. *Each node of G has finite outdegree.*
3. *Each node of G is reachable from some root of G.*

Then there is an infinite path in G starting from some root.

Proof: The usual proof for König's Lemma extends to this case. ■

3 Untimed Automata and Their Behaviors

This section presents the basic definitions and results for untimed automata. It also defines certain restricted kinds of automata that are useful in our proofs, and characterizes the structures that can be obtained as the behaviors of automata.

3.1 Automata

We begin with the definition of an (untimed) automaton. An *automaton* A consists of:

- a set $states(A)$ of states,
- a nonempty set $start(A) \subseteq states(A)$ of start states,
- a set $acts(A)$ of actions that includes a special element τ, and
- a set $steps(A) \subseteq states(A) \times acts(A) \times states(A)$ of steps.

We let $s, s', u, u', ..$ range over states, and $a, ..$ over actions. We let $ext(A)$, the *external actions*, denote $acts(A) - \{\tau\}$. We call τ the *internal action*. The term *event* refers to an occurrence of an action in a sequence. If σ is a sequence of actions then $\widehat{\sigma}$ is the sequence obtained by deleting all τ events from σ. We write $s' \xrightarrow{a}_A s$, or just $s' \xrightarrow{a} s$ if A is clear from the context, as a shorthand for $(s', a, s) \in steps(A)$. In this part of the paper, A, B,..

range over automata. In Part II, however, we will use these symbols to range over timed automata.

An *execution fragment* of A is a finite or infinite alternating sequence $s_0 a_1 s_1 a_2 s_2 \cdots$ of states and actions of A, beginning with a state, and if it is finite also ending with a state, such that for all i, $s_i \xrightarrow{a_{i+1}} s_{i+1}$. We denote by $\mathit{frag}^*(A)$, $\mathit{frag}^\omega(A)$ and $\mathit{frag}(A)$ the sets of finite, infinite, and all execution fragments of A, respectively. An *execution* of A is an execution fragment that begins with a start state. We denote by $\mathit{execs}^*(A)$, $\mathit{execs}^\omega(A)$ and $\mathit{execs}(A)$ the sets of finite, infinite, and all executions of A, respectively. A state s of A is *reachable* if $s = \mathit{last}(\alpha)$ for some finite execution α of A.

Suppose $\alpha = s_0 a_1 s_1 a_2 s_2 \cdots$ is an execution fragment of A. Let γ be the sequence consisting of the actions in α: $\gamma = a_1 a_2 \ldots$. Then $\mathit{trace}(\alpha)$ is defined to be the sequence $\hat{\gamma}$. A finite or infinite sequence β of actions is a *trace* of A if A has an execution α with $\beta = \mathit{trace}(\alpha)$. We write $\mathit{traces}^*(A)$, $\mathit{traces}^\omega(A)$ and $\mathit{traces}(A)$ for the sets of finite, infinite and all traces of A, respectively. These notions induce three *preorders* (i.e., reflexive and transitive relations). For A and B automata, we define $A \leq_{*\mathrm{T}} B \triangleq \mathit{traces}^*(A) \subseteq \mathit{traces}^*(B)$, $A \leq_{\omega\mathrm{T}} B \triangleq \mathit{traces}^\omega(A) \subseteq \mathit{traces}^\omega(B)$, and $A \leq_{\mathrm{T}} B \triangleq \mathit{traces}(A) \subseteq \mathit{traces}(B)$. Recall that the *kernel* of a preorder $\sqsubseteq$ is the equivalence $\equiv$ defined by $x \equiv y \triangleq x \sqsubseteq y \wedge y \sqsubseteq x$. We denote by $\equiv_{*\mathrm{T}}$, $\equiv_{\omega\mathrm{T}}$ and $\equiv_{\mathrm{T}}$, the respective kernels of the preorders $\leq_{*\mathrm{T}}$, $\leq_{\omega\mathrm{T}}$ and $\leq_{\mathrm{T}}$.

Suppose A is an automaton, s' and s are states of A, and β is a finite sequence over $\mathit{ext}(A)$. We say that (s', β, s) is a *move* of A, and write $s' \overset{\beta}{\Longrightarrow}_A s$, or just $s' \overset{\beta}{\Longrightarrow} s$ when A is clear, if A has a finite execution fragment α with $\mathit{first}(\alpha) = s'$, $\mathit{trace}(\alpha) = \beta$ and $\mathit{last}(\alpha) = s$.

Example 3.1 The automata A and B of Figure 1 illustrate the difference between $\leq_{*\mathrm{T}}$ and $\leq_{\mathrm{T}}$.

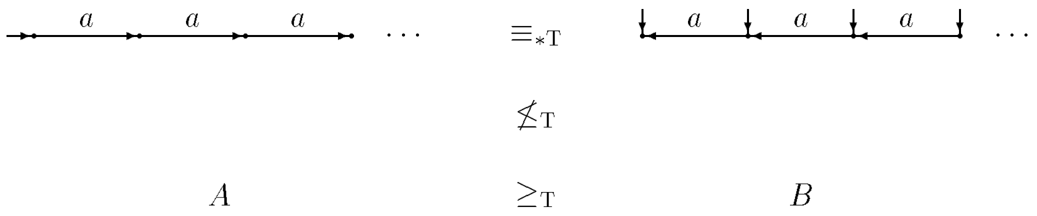

Figure 1: $\leq_{*\mathrm{T}}$ versus $\leq_{\mathrm{T}}$.

3.2 Restricted Kinds of Automata

Automaton A is *deterministic* if $|\mathit{start}(A)| = 1$, and for any state s' and any finite sequence β over $\mathit{ext}(A)$, there is at most one state s such that $s' \overset{\beta}{\Longrightarrow} s$. A deterministic automaton is characterized uniquely by the property that $|\mathit{start}(A)| = 1$, every τ step is of the form (s, τ, s) for some s, and for all states s' and all actions a there is at most one state s such that $s' \xrightarrow{a}_A s$.

A has *finite invisible nondeterminism (fin)* if $\mathit{start}(A)$ is finite, and for any state s' and any finite sequence β over $\mathit{ext}(A)$, there are only finitely many states s such that $s' \overset{\beta}{\Longrightarrow}_A s$.

A is a *forest* if for each state of A there is a unique execution that leads to it. A forest is characterized uniquely by the property that all states of A are reachable, start states have no incoming steps and each of the other states has exactly one incoming step.

The relation $\mathit{after}(A)$ consists of the pairs (β, s) for which there is a finite execution of A with trace β and last state s.

$$\mathit{after}(A) \triangleq \{(\beta, s) \mid \exists \alpha \in \mathit{execs}^*(A) : \mathit{trace}(\alpha) = \beta \text{ and } \mathit{last}(\alpha) = s\}.$$

The relation $\mathit{past}(A) \triangleq \mathit{after}(A)^{-1}$ relates a state s of A to the traces of finite executions of A that lead to s.

Lemma 3.2

1. *If A is deterministic then $\mathit{after}(A)$ is a function from $\mathit{traces}^*(A)$ to $\mathit{states}(A)$.*

2. *If A has fin then $\mathit{after}(A)$ is image-finite.*

3. *If A is a forest then $\mathit{past}(A)$ is a function from $\mathit{states}(A)$ to $\mathit{traces}^*(A)$.*

Example 3.3 In Figure 1, automaton A is deterministic (and so has fin), and is a forest. Automaton B has none of these three properties.

3.3 Trace Properties

For A an automaton, its *behavior*, $\mathit{beh}(A)$, is defined by $\mathit{beh}(A) \triangleq (\mathit{ext}(A), \mathit{traces}(A))$. In this subsection, we characterize the structures that can be obtained as the behavior $\mathit{beh}(A)$ for some automaton A as *trace properties*.

A *trace property* P is a pair (K, L) with K a set and L a nonempty, prefix closed set of (finite or infinite) sequences over K. We will refer to the constituents of P as $\mathit{sort}(P)$ and $\mathit{traces}(P)$, respectively. Also, we write $\mathit{traces}^*(P) \triangleq K^* \cap L$ and $\mathit{traces}^\omega(P) \triangleq K^\omega \cap L$. For P and Q trace properties, we define $P \leq_{*\mathrm{T}} Q \triangleq \mathit{traces}^*(P) \subseteq \mathit{traces}^*(Q)$, $P \leq_{\omega\mathrm{T}} Q \triangleq \mathit{traces}^\omega(P) \subseteq \mathit{traces}^\omega(Q)$, and $P \leq_{\mathrm{T}} Q \triangleq \mathit{traces}(P) \subseteq \mathit{traces}(Q)$. With $\equiv_{*\mathrm{T}}$, $\equiv_{\omega\mathrm{T}}$ and $\equiv_{\mathrm{T}}$, we denote the kernels of the preorders $\leq_{*\mathrm{T}}$, $\leq_{\omega\mathrm{T}}$ and $\leq_{\mathrm{T}}$, respectively. A trace property P is *limit-closed* if an infinite sequence is in $\mathit{traces}(P)$ whenever all its finite prefixes are.

Lemma 3.4 *Suppose P and Q are trace properties with Q limit-closed. Then $P \leq_{*\mathrm{T}} Q \Leftrightarrow P \leq_{\mathrm{T}} Q$.*

Lemma 3.5

1. *$\mathit{beh}(A)$ is a trace property.*

2. *If A has fin then $\mathit{beh}(A)$ is limit-closed.*

3. *$A \leq_{*\mathrm{T}} B \Leftrightarrow \mathit{beh}(A) \leq_{*\mathrm{T}} \mathit{beh}(B)$, $A \leq_{\omega\mathrm{T}} B \Leftrightarrow \mathit{beh}(A) \leq_{\omega\mathrm{T}} \mathit{beh}(B)$, and $A \leq_{\mathrm{T}} B \Leftrightarrow \mathit{beh}(A) \leq_{\mathrm{T}} \mathit{beh}(B)$.*

Proof: It is easy to see that $beh(A)$ is a trace property.

For Part 2, suppose A has fin. We use Lemma 2.1 to show that $beh(A)$ is limit-closed. Suppose β is an infinite sequence over $ext(A)$ such that all finite prefixes of β are in $traces(A)$. Consider the digraph G whose nodes are pairs $(\gamma, s) \in after(A)$, where γ is a finite prefix of β; there is an edge from node (γ', s') to node (γ, s) exactly if γ is of the form $\gamma' a$, where $a \in ext(A)$, and where $s' \overset{a}{\Longrightarrow}_A s$. Then G satisfies the hypotheses of Lemma 2.1, which implies that there is an infinite path in G starting at a root. This corresponds directly to an execution α having $trace(\alpha) = \beta$. Hence, $\beta \in traces(A)$.

Part 3 is immediate from the definitions. ■

Proposition 3.6 *If B has fin then $A \leq_{*\mathrm{T}} B \Leftrightarrow A \leq_{\mathrm{T}} B$.*

Proof: Immediate from Lemma 3.4 and Lemma 3.5. ■

Example 3.7 In Figure 1, $A \leq_{*\mathrm{T}} B$ but $A \not\leq_{\mathrm{T}} B$. Note that B does not have fin.

We close this section with the construction of the *canonical automaton*[1] for a given trace property. For P a trace property, the associated *canonical* automaton $can(P)$ is the structure A given by

- $states(A) = traces^*(P)$,
- $start(A) = \{\lambda\}$,
- $acts(A) = sort(P) \cup \{\tau\}$, and
- for $\beta', \beta \in states(A)$ and $a \in acts(A)$, $\beta' \xrightarrow{a}_A \beta \quad \Leftrightarrow \quad a \in ext(A) \wedge \beta' \, a = \beta$.

Lemma 3.8

1. *$can(P)$ is a deterministic forest.*
2. *$beh(can(P)) \equiv_{*\mathrm{T}} P$.*
3. *$P \leq_{\mathrm{T}} beh(can(P))$.*
4. *If P is limit-closed then $beh(can(P)) \equiv_{\mathrm{T}} P$.*

Proof: Parts 1 and 2 follow easily from the definitions. Since $can(P)$ is deterministic it certainly has fin, so it follows by Lemma 3.5 that $beh(can(P))$ is limit-closed. Now 3 and 4 follow by combination of 2 and Lemma 3.4. ■

Lemma 3.9

1. *$can(beh(A))$ is a deterministic forest.*

[1] This terminology is due to He Jifeng [5].

2. *$can(beh(A)) \equiv_{*\mathrm{T}} A$.*

3. *$A \leq_{\mathrm{T}} can(beh(A))$.*

4. *If A has fin then $can(beh(A)) \equiv_{\mathrm{T}} A$.*

Proof: By combining Lemma 3.5 and Lemma 3.8. ■

4 Basic Simulations

In this section, we develop the basic simulation techniques for untimed automata: refinements and forward and backward simulations.

4.1 Refinements

The simplest type of simulation we consider is a *refinement*. A *refinement* from A to B is a function r from states of A to states of B that satisfies the following two conditions:

1. If $s \in start(A)$ then $r(s) \in start(B)$.

2. If $s' \xrightarrow{a}_A s$ then $r(s') \overset{\hat{a}}{\Longrightarrow}_B r(s)$.

We write $A \leq_{\mathrm{R}} B$ if there exists a refinement from A to B.

This notion is similar to that of a *homomorphism* in classical automata theory; see for instance Ginzberg [4]. Besides our additional treatment of internal actions, a difference between the two notions is that the classical notion involves a mapping between the action sets of the automata, whereas our refinements do not.

Example 4.1 Figure 2 presents some canonical examples of $\leq_{\mathrm{R}}$.

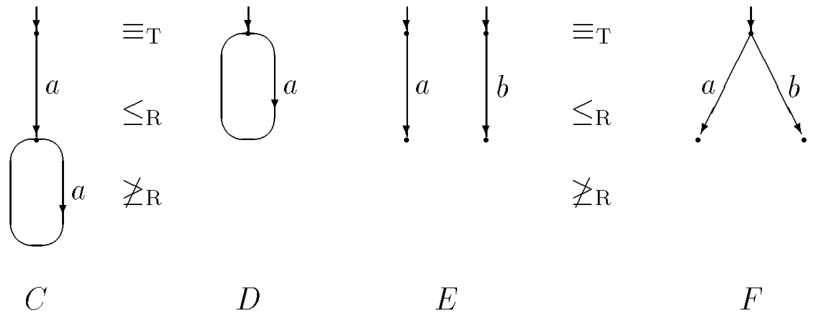

Figure 2: Refinements.

The following technical lemma is a straightforward consequence of the definition of a refinement.

Lemma 4.2 *Suppose r is a refinement from A to B and $s' \overset{\beta}{\Longrightarrow}_A s$. Then $r(s') \overset{\beta}{\Longrightarrow}_B r(s)$.*

Proposition 4.3 *$\leq_{\mathrm{R}}$ is a preorder (i.e., is transitive and reflexive).*

Proof: The identity function $id(states(A))$ is a refinement from A to itself. This implies that $\leq_{\mathrm{R}}$ is reflexive. Using Lemma 4.2, transitivity follows from the observation that if r is a refinement from A to B and r' is a refinement from B to C, $r' \circ r$ is a refinement from A to C. ■

Theorem 4.4 *(Soundness of refinements) $A \leq_{\mathrm{R}} B \;\Rightarrow\; A \leq_{\mathrm{T}} B$.*

Proof: Suppose $A \leq_{\mathrm{R}} B$. Let r be a refinement from A to B, and let e be a function that maps each move (s', β, s) of B to a finite execution fragment of B from s' to s with trace β. Suppose $\beta \in traces(A)$. Then there exists an execution $\alpha = s_0 a_1 s_1 a_2 s_2 \cdots$ of A with $\beta = trace(\alpha)$. By the first condition in the definition of a refinement, $r(s_0)$ is a start state of B, and by the second condition, $r(s_i) \overset{a_{i+1}}{\Longrightarrow}_B r(s_{i+1})$ for all i. For $i \geq 0$, define $\alpha_i = e((r(s_i), \widehat{a_{i+1}}, r(s_{i+1})))$. Next define sequence α' to be the (infinitary) concatenation $\alpha_0 tail(\alpha_1) tail(\alpha_2) \cdots$. By construction, α' is an execution of B with $trace(\alpha') = trace(\alpha) = \beta \in traces(B)$. ■

Theorem 4.5 *(Partial completeness of refinements) Suppose A is a forest, B is deterministic and $A \leq_{*\mathrm{T}} B$. Then $A \leq_{\mathrm{R}} B$.*

Proof: The relation $r \triangleq after(B) \circ past(A)$ is a refinement from A to B. ■

4.2 Forward Simulations

A *forward simulation* from A to B is a relation f over $states(A)$ and $states(B)$ that satisfies:

1. If $s \in start(A)$ then $f[s] \cap start(B) \neq \emptyset$.
2. If $s' \xrightarrow{a}_A s$ and $u' \in f[s']$, then there exists a state $u \in f[s]$ such that $u' \overset{\hat{a}}{\Longrightarrow}_B u$.

We write $A \leq_{\mathrm{F}} B$ if there exists a forward simulation from A to B.

Example 4.6 Let C, D, E, F be as in Figure 2. Then $D \leq_{\mathrm{F}} C$ and $F \not\leq_{\mathrm{F}} E$.

Proposition 4.7 *$A \leq_{\mathrm{R}} B \Rightarrow A \leq_{\mathrm{F}} B$.*

Proof: Any refinement relation is a forward simulation. ■

The following lemma is the analogue for forward simulations of Lemma 4.2.

Lemma 4.8 *Suppose f is a forward simulation from A to B and $s' \overset{\beta}{\Longrightarrow}_A s$. If $u' \in f[s']$, then there exists a state $u \in f[s]$ such that $u' \overset{\beta}{\Longrightarrow}_B u$.*

Proposition 4.9 *$\leq_{\mathrm{F}}$ is a preorder.*

Proof: For reflexivity, observe that the identity function $id(states(A))$ is a forward simulation from A to itself. For transitivity, use Lemma 4.8 to show that if f and f' are forward simulations from A to B and from B to C, respectively, $f' \circ f$ is a forward simulation from A to C. ■

Theorem 4.10 *(Soundness of forward simulations, [21, 9, 30])* $A \leq_{\mathrm{F}} B \Rightarrow A \leq_{\mathrm{T}} B$.

Proof: Versions of this proof appears in the cited papers. The proof is similar to that of Theorem 4.4. ■

Theorem 4.11 *(Partial completeness of forward simulations) Suppose B is deterministic and $A \leq_{*\mathrm{T}} B$. Then $A \leq_{\mathrm{F}} B$.*

Proof: The relation $f \triangleq after(B) \circ past(A)$ is a forward simulation from A to B. ■

The following proposition allows us to give an alternative proof of the partial completeness result for refinements (Theorem 4.5): if A is a forest, B is deterministic and $A \leq_{*\mathrm{T}} B$, then $A \leq_{\mathrm{F}} B$ by Theorem 4.11, and from that $A \leq_{\mathrm{R}} B$ follows using Prop. 4.12. Interestingly, Prop. 4.12 is the only result for which we have not been able to prove an analogue in the timed case.

Proposition 4.12 *Suppose A is a forest and $A \leq_{\mathrm{F}} B$. Then $A \leq_{\mathrm{R}} B$.*

Proof: Let f be a forward simulation from A to B. We construct a choice function r for f, and prove that r is a refinement from A to B.

For $n \geq 0$, let $Layer_n$ be the set of states s of A for which the (unique) execution leading to it contains n actions. Then the sets $Layer_n$ $(n \geq 0)$ partition the set $states(A)$ and $Layer_0 = start(A)$. We define functions $r_n : Layer_n \rightarrow states(B)$ inductively such that $r_n(s) \in f[s]$. By Condition 1 in the definition of a forward simulation, there exists a function $r_0 : Layer_0 \rightarrow start(B)$ satisfying $r_0(s) \in f[s]$. Suppose that r_i has been defined for $i \leq n$. By Condition 2 in the definition of a forward simulation, there exists a function $r_{n+1} : Layer_{n+1} \rightarrow states(B)$ such that if s is in $Layer_{n+1}$ and $s' \xrightarrow{a}_A s$ is the unique incoming step of s, we have $r_n(s') \overset{\hat{a}}{\Longrightarrow}_B r_{n+1}(s)$ and $r_{n+1}(s) \in f[s]$. By construction, the union r of the functions r_n is a refinement from A to B with $r(s) \in f[s]$. ■

4.3 Backward Simulations

In many respects, backward simulations are the dual of forward simulations. Whereas a forward simulation requires that *some* state in the image of each start state should be a start state, a backward simulation requires that *all* states in the image of a start state be start states. Also, a forward simulation requires that forward steps in the source automaton can be simulated from related states in the target automaton, whereas the corresponding condition for a backward simulations requires that backward steps can be simulated. However, the two notions are not completely dual: the definition of a backward simulation contains a nonemptiness condition, and also, in order to imply soundness in general, backward simulations also require a finite image condition. The mismatch is due to the asymmetry in our automata between future and past: from any given state, all the possible histories are finite executions, whereas the possible futures can be infinite.

A *backward simulation* from A to B is a total relation b over $states(A)$ and $states(B)$ that satisfies:

1. If $s \in start(A)$ then $b[s] \subseteq start(B)$.

2. If $s' \xrightarrow{a}_A s$ and $u \in b[s]$, then there exists a state $u' \in b[s']$ such that $u' \overset{\hat{a}}{\Longrightarrow}_B u$.

We write $A \leq_{\mathrm{B}} B$ if there exists a backward simulation from A to B, and $A \leq_{\mathrm{iB}} B$ if there exists an image-finite backward simulation from A to B.

Example 4.13 Let A, B be as in Figure 1. Then $A \leq_{\mathrm{B}} B$ but $A \not\leq_{\mathrm{iB}} B$. If C, D, E, F are as in Figure 2, then $D \not\leq_{\mathrm{B}} C$ and $F \leq_{\mathrm{iB}} E$.

Proposition 4.14 $A \leq_{\mathrm{R}} B \Rightarrow A \leq_{\mathrm{iB}} B$.

The following lemma is useful in the proofs of the preorder properties and of soundness.

Lemma 4.15 *Suppose b is a backward simulation from A to B and $s' \overset{\beta}{\Longrightarrow}_A s$. If $u \in b[s]$, then there exists a state $u' \in b[s']$ such that $u' \overset{\beta}{\Longrightarrow}_B u$.*

Proposition 4.16 *$\leq_{\mathrm{B}}$ and $\leq_{\mathrm{iB}}$ are preorders.*

Proof: The identity function $id(states(A))$ is a backward simulation from A to itself. Using Lemma 4.15 one can easily show that if b is backward simulation from A to B and b' is a backward simulation from B to C, $b' \circ b$ is a backward simulation from A to C. Moreover, if both b and b' are image-finite, then $b' \circ b$ is image-finite too. ■

Theorem 4.17 *(Soundness of backward simulations)*

1. $A \leq_{\mathrm{B}} B \Rightarrow A \leq_{*\mathrm{T}} B$.

2. $A \leq_{\mathrm{iB}} B \Rightarrow A \leq_{\mathrm{T}} B$.

Proof: Suppose b is a backward simulation from A to B and suppose $\beta \in traces^*(A)$. Then there is a move $s' \overset{\beta}{\Longrightarrow}_A s$, where s' is a start state of A. Since b is a backward simulation it is a total relation, so there exists a state $u \in b[s]$. By Lemma 4.15, there exists $u' \in b[s']$ with $u' \overset{\beta}{\Longrightarrow}_B u$. By the first condition of the definition of a backward simulation, $u' \in start(B)$. Therefore, $\beta \in traces^*(B)$, which shows the first part of the proposition.

For the second part, suppose that b is image-finite. We have already established $A \leq_{*\mathrm{T}} B$, so it is sufficient to show $A \leq_{\omega\mathrm{T}} B$. Suppose that $\beta \in traces^\omega(A)$, and let $\alpha = s_0 a_1 s_1 a_2 \cdots$ be an infinite execution of A with $trace(\alpha) = \beta$.

Consider the digraph G whose nodes are pairs (u, i) such that $(s_i, u) \in b$ and in which there is an edge from (u', i') to (u, i) exactly if $i = i' + 1$ and $u' \overset{\widehat{a_i}}{\Longrightarrow}_B u$. Then G satisfies the hypotheses of Lemma 2.1, which implies that there is an infinite path in G starting at a root. This corresponds directly to an execution α' of B having $trace(\alpha') = trace(\alpha) = \beta$. Hence, $\beta \in traces(B)$. ■

In a recent paper, Jonsson [11] considers a weaker image-finiteness condition for backward simulations. Translated into our setting, the key observation of Jonsson is that in order to prove $A \leq_{\mathrm{T}} B$, it is enough to give a backward simulation b from A to B with the property that each infinite execution of A contains infinitely many states s with $b[s]$ finite. We do not explore this extension in this paper, primarily because it lacks a key feature of simulation techniques. Namely, it fails to reduce global reasoning about infinite behaviors to local reasoning about states and actions.

The following partial completeness result slightly generalizes a similar result of Jonsson [10] in that it also allows for τ-steps in the B automaton.

Theorem 4.18 *(Partial completeness of backward simulations) Suppose A is a forest and $A \leq_{*\mathrm{T}} B$. Then*

1. *$A \leq_{\mathrm{B}} B$, and*

2. *if B has fin then $A \leq_{\mathrm{iB}} B$.*

Proof: We define a relation b over $states(A)$ and $states(B)$. Suppose s is a state of A. Since A is a forest there is a unique trace leading up to s, say β. Now define

$$b[s] = \{u \mid \exists \alpha \in execs^*(B)\colon trace(\alpha) = \beta,\ last(\alpha) = u \wedge [\alpha' < \alpha \Rightarrow trace(\alpha') \neq \beta]\}.$$

By letting $b[s]$ consist only of those states of B which can be reached via a *minimal* execution with trace β, we achieve that, if s is a start state, all the states in $b[s]$ are start states of B. It is also the case that b satisfies the other conditions in the definition of a backward simulation.

Lemma 3.2 implies that b is image-finite if B has fin. ■

The next proposition is the dual of Prop. 4.12, and provides us with yet another proof of the partial completeness result for refinements (Theorem 4.5), now using Theorem 4.18. Unlike Prop. 4.12, Prop. 4.19 does have an analogue in the timed case.

Proposition 4.19 *Suppose all states of A are reachable, B is deterministic and $A \leq_{\mathrm{B}} B$. Then $A \leq_{\mathrm{R}} B$.*

Proof: Let b be a backward simulation from A to B and let s be a reachable state of A. We will prove that $b[s]$ contains exactly one element. Because all states of A are reachable, it follows that b is functional. But any functional backward simulation trivially is a refinement, and so we obtain $A \leq_{\mathrm{R}} B$.

Since b is a backward simulation, it is a total relation, so we know $b[s]$ contains at least one element. Suppose that both $u \in b[s]$ and $u' \in b[s]$; we prove $u = u'$. Since s is reachable, there exists a finite execution α of A with last state s. By carrying out the same construction as in the proof of Theorem 4.17, we can construct two finite executions γ and γ' of B with

γ ending in u and γ' ending in u' such that α, γ and γ' all have the same trace, say β. Thus both u and u' are in the set $\mathit{after}(B)[\beta]$. But this means that they are equal because since B is deterministic, $\mathit{after}(B)[\beta]$ contains only a single element according to Lemma 3.2. ■

Proposition 4.20 *Suppose all states of A are reachable, B has fin and $A \leq_{\mathrm{B}} B$. Then $A \leq_{\mathrm{iB}} B$.*

Proof: Let b be a backward simulation from A to B and let s be a state of A. Since s is reachable we can find a trace $\beta \in \mathit{past}(A)[s]$. From the fact that b is a backward simulation it follows that $b[s] \subseteq \mathit{after}(B)[\beta]$. But since B has fin, $\mathit{after}(B)[\beta]$ is finite by Lemma 3.2. This implies that b is image-finite. ■

Example 4.21 Figure 3 shows that the reachability assumption in Prop. 4.20 is essential. There is a backward simulation from G to H, but even though H is deterministic there is no image-finite backward simulation.

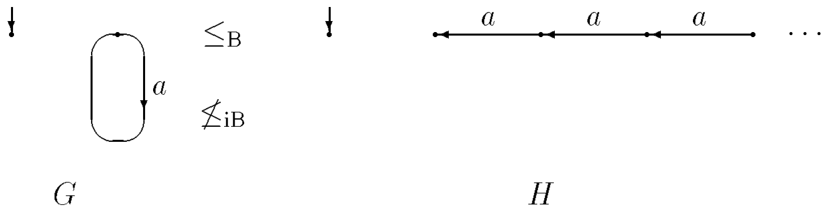

Figure 3: $\leq_{\mathrm{B}}$ and $\leq_{\mathrm{iB}}$ are different, even for automata with fin.

4.4 Combined Forward and Backward Simulations

Several authors have observed that forward and backward simulations together give a complete proof method (see [7, 5, 12, 10, 11, 14]): if $A \leq_{*\mathrm{T}} B$ then there exists an intermediate automaton C with a forward simulation from A to C and a backward simulation from C to B. We prove this below by taking C to be the canonical automaton of A, as defined in Section 3. Alternative proofs can be given using different intermediate automata, for example the automaton obtained by applying the classical subset construction on B (see [11, 14]).

Theorem 4.22 *(Completeness of forward and backward simulations) If $A \leq_{*\mathrm{T}} B$ then the following are true.*

1. *$\exists C : A \leq_{\mathrm{F}} C \leq_{\mathrm{B}} B$.*
2. *If B has fin then $\exists C : A \leq_{\mathrm{F}} C \leq_{\mathrm{iB}} B$.*

Proof: Let $C = \mathit{can}(\mathit{beh}(A))$. By Lemma 3.9, C is a deterministic forest and $A \equiv_{*\mathrm{T}} C$. Since C is deterministic, $A \leq_{\mathrm{F}} C$ by Theorem 4.11, and because C is a forest, $C \leq_{\mathrm{B}} B$ follows by Theorem 4.18(1). If B has fin then $C \leq_{\mathrm{iB}} B$ follows by Theorem 4.18(2). ■

5 Hybrid Simulations

5.1 Forward-Backward Simulations

Forward-backward simulations were introduced by Klarlund and Schneider who call them *invariants* in [13] and *ND measures* in [14]. They also occur in the work of Jonsson [11] under the name *subset simulations*, and are related to the *failure simulations* of Gerth [3]. Forward-backward simulations combine in a single relation both a forward and a backward simulation. Below we present simple proofs of their soundness and completeness by making this connection explicit.

Formally, a *forward-backward simulation* from A to B is a relation g over $\mathit{states}(A)$ and $\mathbf{N}(\mathit{states}(B))$ that satisfies:

1. If $s \in \mathit{start}(A)$ then there exists $S \in g[s]$ such that $S \subseteq \mathit{start}(B)$.
2. If $s' \xrightarrow{a}_A s$ and $S' \in g[s']$, then there exists a set $S \in g[s]$ such that for every $u \in S$ there exists $u' \in S'$ with $u' \overset{\hat{a}}{\Longrightarrow}_B u$.

We write $A \leq_{\mathrm{FB}} B$ if there exists a forward-backward simulation from A to B, and $A \leq_{\mathrm{iFB}} B$ if there exists an image-set-finite forward-backward simulation from A to B.

The following theorem says that a forward-backward simulation is essentially just a combination of a forward and a backward simulation.

Theorem 5.1

1. $A \leq_{\mathrm{FB}} B \Leftrightarrow (\exists C : A \leq_{\mathrm{F}} C \leq_{\mathrm{B}} B)$.

2. $A \leq_{\mathrm{iFB}} B \Leftrightarrow (\exists C : A \leq_{\mathrm{F}} C \leq_{\mathrm{iB}} B)$.

Proof: "$\Rightarrow$" Let g be a forward-backward simulation from A to B, which is image-set-finite if $A \leq_{\mathrm{iFB}} B$. Define C to be the automaton given by:

- $\mathit{states}(C) = \mathit{range}(g)$,
- $\mathit{start}(C) = \mathit{range}(g) \cap \mathbf{P}(\mathit{start}(B))$,
- $\mathit{acts}(C) = \mathit{acts}(B)$, and
- for $S', S \in \mathit{states}(C)$ and $a \in \mathit{acts}(C)$, $S' \xrightarrow{a}_C S \quad \Leftrightarrow \quad \forall u \in S \; \exists u' \in S' : u' \overset{\hat{a}}{\Longrightarrow}_B u$.

Then g is a forward simulation from A to C. Also, $\{(S, u) \mid S \in \mathit{states}(C) \text{ and } u \in S\}$ is a backward simulation from C to B, which is image finite if g is image-set-finite.

"$\Leftarrow$" Suppose f is a forward simulation from A to C, and b is a backward simulation from C to B. Then the relation g over $\mathit{states}(A)$ and $\mathbf{N}(\mathit{states}(B))$ defined by $g = \{(s, b[u]) \mid (s, u) \in f\}$ is a forward-backward simulation from A to B. If b is image-finite then g is image-set-finite. ■

Proposition 5.2

1. $A \leq_{\mathrm{F}} B \Rightarrow A \leq_{\mathrm{iFB}} B$.
2. $A \leq_{\mathrm{B}} B \Rightarrow A \leq_{\mathrm{FB}} B$.
3. $A \leq_{\mathrm{iB}} B \Rightarrow A \leq_{\mathrm{iFB}} B$.

Proof: Immediate from Theorem 5.1, using that $\leq_{\mathrm{iB}}$ and $\leq_{\mathrm{F}}$ are reflexive. ■

In order to show that $\leq_{\mathrm{FB}}$ and $\leq_{\mathrm{iFB}}$ are preorders, we require a definition of composition for forward-backward simulations, and a transitivity lemma.

If g is a relation over X and $\mathbf{N}(Y)$ and g' is a relation over Y and $\mathbf{N}(Z)$ then the composition $g' \bullet g$ is a relation over X and $\mathbf{N}(Z)$ defined as follows.

$$(x, S') \in g' \bullet g \Leftrightarrow \exists S \in g[x], \exists c, \text{ a choice function for } g' \lceil S : S' = \bigcup \{c(y) : y \in S\}.$$

(The nonemptiness assumptions for g and g' immediately imply the nonemptiness assumption for $g' \bullet g$.)

Lemma 5.3 *Suppose g is a forward-backward simulation from A to B, and g' is a forward-backward simulation from B to C. Then $g' \bullet g$ is a forward-backward simulation from A to C. Moreover, if g and g' are image-set-finite then $g' \bullet g$ is also image-set-finite.*

Proof: For Condition 1 of the definition of a forward-backward simulation, suppose $s \in$ *start*(A). Because g is a forward-backward simulation, there is a set $S \in g[s]$ with $S \subseteq$ *start*(B). Since g' is a forward-backward simulation, it is possible to find, for each $u \in S$, a set $S_u \in g'[u]$ with $S_u \subseteq$ *start*(C). Hence all states in the set $S' = \bigcup\{S_u \mid u \in S\}$ are start states of C. Now let c be the function with domain S given by $c(u) = S_u$. Then c is a choice function for $g' \lceil S$. From the definition of $\bullet$ it now follows that $(s, S') \in g' \bullet g$. This shows that $g' \bullet g$ satisfies Condition 1.

Now we show Condition 2 of the definition of a forward-backward simulation. Suppose $s' \xrightarrow{a}_A s$ and $(s', S') \in g' \bullet g$. By definition of $g' \bullet g$, there exist $T' \in g[s']$ and a choice function c' for $g' \lceil T'$ such that $S' = \bigcup\{c'(u') : u' \in T'\}$. Because g is a forward-backward simulation from A to B, there is a set $T \in g[s]$ such that for each $u \in T$ there exists $u' \in T'$ with $u' \overset{\hat{a}}{\Longrightarrow}_B u$. Consider any particular $u \in T$. Choose $u' \in T'$ with $u' \overset{\hat{a}}{\Longrightarrow}_B u$. Because g' is a forward-backward simulation, there exists a set $S_u \in g'[u]$ such that for every $v \in S_u$ there exists a $v' \in c'(u')$ with $v' \overset{\hat{a}}{\Longrightarrow}_C v$. Define a choice function c for $g' \lceil T$ by taking $c(u)$ to be the set S_u.

Now consider the set $S = \bigcup\{c(u) : u \in T\}$. Then $(s, S) \in g' \bullet g$ by definition. By construction, we can find, for each $v \in S$, a state $v' \in S'$ with $v' \overset{\hat{a}}{\Longrightarrow}_C v$. Thus S has the required property to show Condition 2.

Finally, it is immediate from the definitions that, if g and g' are image-set-finite, $g' \bullet g$ is also image-set-finite. ■

Proposition 5.4 *$\leq_{\mathrm{FB}}$ and $\leq_{\mathrm{iFB}}$ are preorders.*

Proof: By Lemma 5.3. ■

Theorem 5.5 *(Soundness of forward-backward simulations, [13])*

1. $A \leq_{\mathrm{FB}} B \Rightarrow A \leq_{*\mathrm{T}} B$.
2. $A \leq_{\mathrm{iFB}} B \Rightarrow A \leq_{\mathrm{T}} B$.

Proof: For part 1, suppose $A \leq_{\mathrm{FB}} B$. By Theorem 5.1, there exists an automaton C with $A \leq_{\mathrm{F}} C \leq_{\mathrm{B}} B$. By soundness of forward simulations, Theorem 4.10, $A \leq_{\mathrm{T}} C$, and by soundness of backward simulations, Theorem 4.17, $C \leq_{*\mathrm{T}} B$. This implies $A \leq_{*\mathrm{T}} B$. Part 2 is similar. ■

Theorem 5.6 *(Completeness of forward-backward simulations, [13]) Suppose $A \leq_{*\mathrm{T}} B$. Then*

1. *$A \leq_{\mathrm{FB}} B$, and*
2. *if B has fin then $A \leq_{\mathrm{iFB}} B$.*

Proof: By Theorem 4.22, there exists an automaton C with $A \leq_{\mathrm{F}} C \leq_{\mathrm{B}} B$. Moreover, if B has fin then $A \leq_{\mathrm{F}} C \leq_{\mathrm{iB}} B$. Then Theorem 5.1 implies the needed conclusions. ■

5.2 Backward-Forward Simulations

Having studied forward-backward simulations, we find it natural to define and study a dual notion of backward-formulation simulation.

A *backward-forward simulation* from A to B is a total relation g over $\mathit{states}(A)$ and $\mathbf{P}(\mathit{states}(B))$ that satisfies:

1. If $s \in \mathit{start}(A)$ then, for all $S \in g[s]$, $S \cap \mathit{start}(B) \neq \emptyset$.
2. If $s' \xrightarrow{a}_A s$ and $S \in g[s]$, then there exists a set $S' \in g[s']$ such that for every $u' \in S'$ there exists a $u \in S$ with $u' \overset{\hat{a}}{\Longrightarrow}_B u$.

We write $A \leq_{\mathrm{BF}} B$ if there exists a backward-forward simulation from A to B, and $A \leq_{\mathrm{iBF}} B$ if there exists an image-finite backward-forward simulation from A to B.

As for forward-backward simulations, backward-forward simulations can be characterized as combinations of forward and backward simulations.

Theorem 5.7

1. $A \leq_{\mathrm{BF}} B \Leftrightarrow (\exists C : A \leq_{\mathrm{B}} C \leq_{\mathrm{F}} B)$.
2. $A \leq_{\mathrm{iBF}} B \Leftrightarrow (\exists C : A \leq_{\mathrm{iB}} C \leq_{\mathrm{F}} B)$.

Proof: "$\Rightarrow$" Let g be a backward-forward simulation from A to B, which is image-finite if $A \leq_{\mathrm{iBF}} B$. Define C to be the automaton given by:

- $\mathit{states}(C) = \mathit{range}(g)$,

- $start(C) = range(g\lceil start(A))$,
- $acts(C) = acts(B)$, and
- for $S', S \in states(C)$ and $a \in acts(C)$, $S' \xrightarrow{a}_C S \quad \Leftrightarrow \quad \forall u' \in S' \; \exists u \in S : u' \overset{\hat{a}}{\Longrightarrow}_B u$.

Then g is a backward simulation from A to C (and image-finiteness carries over). Also, the relation $\{(S, u) \mid S \in states(C) \text{ and } u \in S\}$ is a forward simulation from C to B.

"$\Leftarrow$" Easy. ■

Proposition 5.8

1. $A \leq_{\mathrm{F}} B \Rightarrow A \leq_{\mathrm{iBF}} B$.

2. $A \leq_{\mathrm{B}} B \Rightarrow A \leq_{\mathrm{BF}} B$.

3. $A \leq_{\mathrm{iB}} B \Rightarrow A \leq_{\mathrm{iBF}} B$.

Proof: Immediate from Theorem 5.7, using the fact that $\leq_{\mathrm{iB}}$ and $\leq_{\mathrm{F}}$ are reflexive. ■

In order to show the properties of backward-forward simulations, it is useful to relate them to forward-backward simulations.

Theorem 5.9

1. $A \leq_{\mathrm{BF}} B \Leftrightarrow A \leq_{\mathrm{FB}} B$.

2. $A \leq_{\mathrm{iBF}} B \Rightarrow A \leq_{\mathrm{iFB}} B$.

Proof: For one direction of 1, suppose that $A \leq_{\mathrm{BF}} B$. Then by Theorem 5.7, there exists an automaton C with $A \leq_{\mathrm{B}} C \leq_{\mathrm{F}} B$. By Prop. 5.2, $A \leq_{\mathrm{FB}} C$ and $C \leq_{\mathrm{FB}} B$. Now $A \leq_{\mathrm{FB}} B$ follows by Prop. 5.4. The proof of 2 is similar.

For the other direction of 1, suppose that f is a forward-backward simulation from A to B. Given a state s of A, we define $g[s]$ to be exactly the set of subsets S of $states(B)$ such that S intersects each set in $f[s]$ in at least one element. Then g is a backward-forward simulation. ■

Example 5.10 In general it is not the case that $A \leq_{\mathrm{iFB}} B$ implies $A \leq_{\mathrm{iBF}} B$. A counterexample is presented in Figure 4. The diagram shows two automata I and J. In the diagram a label $> i$ next to an arc means that in fact there are infinitely many steps, labeled $i+1$, $i+2$, $i+3$, etc..

We claim that the relation g given by

$$\begin{aligned} g[0] &= \{\{0\}, \{0', 1\}, \{0', 1', 2\}, \ldots\} \\ g[n] &= \{\{\omega\}, \{\omega'\}\} \quad \text{for } n > 0 \end{aligned}$$

is an image-set-finite forward-backward simulation from I to J.

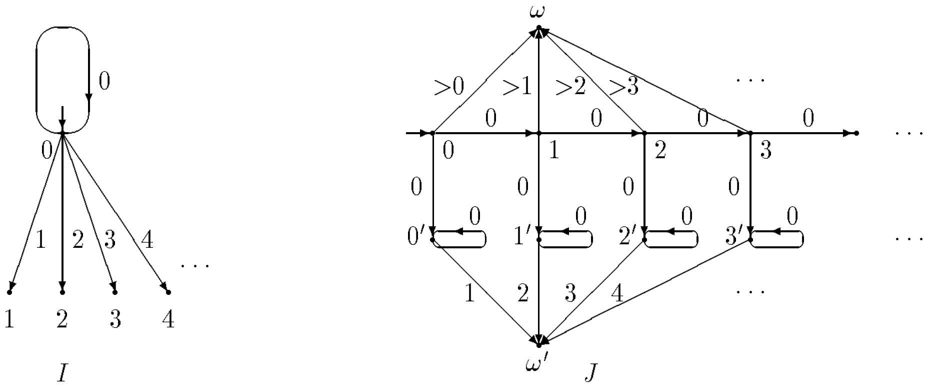

Figure 4: $I \leq_{\text{iFB}} J$ but $I \not\leq_{\text{iBF}} J$.

However, there is no image-finite backward-forward simulation from I to J. We see this as follows. Suppose g is an image-finite backward-forward simulation from I to J. In order to prove that this assumption leads to a contradiction, we first establish that $g[0]$ does not contain a finite subset X of N. First note that by the first condition in the definition of a backward-forward simulation, all sets in $g[0]$ are nonempty. The proof proceeds by induction on the maximal element of X. For the induction base, observe that $\{0\} \notin g[0]$, since 0 has an incoming 0-step in I but not in J. For the induction step, suppose that we have established that $g[0]$ contains no finite subset of N with a maximum less than n, and suppose $X \in g[0]$ with X a finite subset of N with maximum n. Using that 0 has an incoming 0-step in I, the second condition in the definition of a backward-forward simulation gives that $g[0]$ contains an element of $g[0]$ which is a subset of N with a maximum less than n. This contradicts the induction hypothesis.

Pick some state $n > 0$ of I and a set $S' \in g[n]$. Since $0 \xrightarrow{n}_I n$, there exists a set $S \in g[0]$ such that every state in S has an outgoing n-step. Then S must be a subset of $\{0, \ldots, n-1, (n-1)'\}$. Since $g[0]$ does not contain the empty set or a finite subset of N, it follows that $(n-1)' \in S$. But since n was chosen arbitrarily (besides being positive) it follows that $g[0]$ has an infinite number of elements. This gives a contradiction with the assumption that g is image-finite.

Proposition 5.11 $\leq_{\text{BF}}$ *is a preorder. (However,* $\leq_{\text{iBF}}$ *is not a preorder.)*

Proof: The fact that $\leq_{\text{BF}}$ is a preorder, is trivially implied by Theorem 5.9 and Prop. 5.4.

The counterexample of Figure 4 tells us that $\leq_{\text{iBF}}$ is not a preorder in general. If we take the two automata I and J from the example, then we can find an automaton C with $I \leq_{\text{F}} C \leq_{\text{iB}} J$, using Theorem 4.22. By Prop. 5.8, $I \leq_{\text{iBF}} C$ and $C \leq_{\text{iBF}} J$. Hence it cannot be that $\leq_{\text{iBF}}$ is transitive, because this would imply $I \leq_{\text{iBF}} J$. ■

Soundness and completeness results for backward-forward simulations now follow from those for forward-backward simulations.

Theorem 5.12 *(Soundness of backward-forward simulations)*

1. $A \leq_{\mathrm{BF}} B \Rightarrow A \leq_{*\mathrm{T}} B$.

2. $A \leq_{\mathrm{iBF}} B \Rightarrow A \leq_{\mathrm{T}} B$.

Proof: By Theorem 5.9 and Theorem 5.5. ■

Theorem 5.13 *(Completeness of backward-forward simulations)* $A \leq_{*\mathrm{T}} B \Rightarrow A \leq_{\mathrm{BF}} B$.

Proof: By Theorem 5.6 and Theorem 5.9. ■

Example 5.10 falsifies the completeness result that one might expect here. That is, Theorem 5.13 does not have a second case saying that if B has fin and $A \leq_{*\mathrm{T}} B$, then $A \leq_{\mathrm{iBF}} B$.

6 Auxiliary Variable Constructions

In this section, we present two new types of relations, history relations and prophecy relations, which correspond to the notions of history and prophecy variable of Abadi and Lamport [1]. We show that there is a close connection between history relations and forward simulations, and also between prophecy relations and backward simulations. Using these connections together with the earlier results of this section, we can easily derive a completeness theorem for refinements similar to the one of Abadi and Lamport [1]. In fact, in the setting of this paper, the combination of history and prophecy relations and refinements gives exactly the same verification power as the combination of forward and backward simulations.

6.1 History Relations

A relation h over $states(A)$ and $states(B)$ is a *history relation* from A to B if h is a forward simulation from A to B and h^{-1} is a refinement from B to A. We write $A \leq_{\mathrm{H}} B$ if there exists a history relation from A to B. Thus $A \leq_{\mathrm{H}} B$ implies $A \leq_{\mathrm{F}} B$ and $B \leq_{\mathrm{R}} A$.

We give an example of a history relation, using the construction of the *unfolding* of an automaton; the unfolding of an automaton augments the automaton by remembering information about the past.

The *unfolding* of an automaton A, notation $unfold(A)$, is the automaton B defined by

- $states(B) = execs^*(A)$,
- $start(B) =$ the set of finite executions of A that consist of a single start state,
- $acts(B) = acts(A)$, and
- for $\alpha', \alpha \in states(B)$ and $a \in acts(B)$, $\alpha' \xrightarrow{a}_B \alpha \quad \Leftrightarrow \quad \alpha = \alpha' \, a \, last(\alpha)$.

Proposition 6.1 *$unfold(A)$ is a forest and $A \leq_{\mathrm{H}} unfold(A)$.*

Proof: Clearly, *unfold*(A) is a forest. The function *last* which maps each finite execution of A to its last state is a refinement from *unfold*(A) to A, and the relation $last^{-1}$ is a forward simulation from A to *unfold*(A). ■

Example 6.2 For C, D, E, F as in Example 4.1, $C \not\leq_{\mathrm{H}} D$, $D \leq_{\mathrm{H}} C$, $E \not\leq_{\mathrm{H}} F$ and $F \not\leq_{\mathrm{H}} E$.

Proposition 6.3 *$\leq_{\mathrm{H}}$ is a preorder.*

Proof: Reflexivity is trivial. For transitivity, suppose h is a history relation from A to B and h' is a history relation from B to C. Then h is a forward simulation from A to B and h' is a forward simulation from B to C, so $h' \circ h$ is a forward simulation from A to C, by Prop. 4.9. Also, since h'^{-1} is a refinement from C to B and h^{-1} is a refinement from B to A, $(h' \circ h)^{-1} = h^{-1} \circ h'^{-1}$ is a refinement from C to A by Prop. 4.3. It now follows that $h' \circ h$ is a history relation from A to C. ■

The notion of a history relation is a new contribution of this paper. It provides a simple and abstract view of the *history variables* of Abadi and Lamport [1] (which in turn are abstractions of the *auxiliary variables* of Owicki and Gries [26]). Translated into the setting of this paper, history variables can be simply defined in terms of history relations, as follows.

An automaton B is obtained from an automaton A by *adding a history variable* if there exists a set V such that

- *states*$(B) \subseteq$ *states*$(A) \times V$, and
- the relation $\{(s,(s,v)) \mid (s,v) \in \mathit{states}(B)\}$ is a history relation from A to B.

Whenever B is obtained from A by adding a history variable, then $A \leq_{\mathrm{H}} B$ by definition. The following proposition states that the converse is also true if one is willing to consider automata up to isomorphism.

Proposition 6.4 *Suppose $A \leq_{\mathrm{H}} B$. Then there exists an automaton C that is isomorphic to B and obtained from A by adding a history variable.*

Proof: Let h be a history relation from A to B. Define automaton C by

- *states*$(C) = h$,
- $(s,u) \in \mathit{start}(C) \;\Leftrightarrow\; u \in \mathit{start}(B)$,
- *acts*$(C) =$ *acts*(B), and
- for $(s',u'), (s,u) \in \mathit{states}(C)$ and $a \in \mathit{acts}(C)$, $(s',u') \xrightarrow{a}_C (s,u) \;\Leftrightarrow\; u' \xrightarrow{a}_B u$.

Clearly, the projection function π_2 that maps a state (s,u) of C to the state u of B is an isomorphism between C and B.

In order to show that C is obtained from A by adding a history variable, let *states*(B) play the role of the set V required in the definition of a history variable. It is easy to check that relation $\{(s,(s,v)) \mid (s,v) \in \mathit{states}(C)\}$ is a history relation from A to C. ■

Prop. 6.4 shows that history relations already capture the essence of history variables. For this reason and also because history relations have nicer theoretical properties, we will state all our results in this subsection in terms of relations, and will not mention the auxiliary variables any further.

Theorem 6.5 *(Soundness of history relations)* $A \leq_{\mathrm{H}} B \Rightarrow A \equiv_{\mathrm{T}} B$.

Proof: Immediate from the soundness of refinements and forward simulations. ■

In fact, a history relation from A to B is just a functional *bisimulation* between A and B in the sense of Park [27] and Milner [25]. This implies that if there exists a history relation from A to B, both automata are *bisimulation equivalent*. Hence, history relations preserve the behavior of automata in a very strong sense.

We can now state and prove the completeness results of Sistla [29].

Theorem 6.6 *(Completeness of history relations and backward simulations, [29]) Suppose* $A \leq_{*\mathrm{T}} B$. *Then*

1. $\exists C : A \leq_{\mathrm{H}} C \leq_{\mathrm{B}} B$, *and*
2. *if* B *has fin then* $\exists C : A \leq_{\mathrm{H}} C \leq_{\mathrm{iB}} B$.

Proof: By Prop. 6.1, *unfold*(A) is a forest and $A \leq_{\mathrm{H}}$ *unfold*(A). Since $A \leq_{*\mathrm{T}} B$, also *unfold*$(A) \leq_{*\mathrm{T}} B$ by the soundness of history relations (Theorem 6.5). Next we can apply the partial completeness result for backward simulations (Theorem 4.18) to conclude (1) *unfold*$(A) \leq_{\mathrm{B}} B$, and (2) if B has fin then *unfold*$(A) \leq_{\mathrm{iB}} B$. ■

Suppose k is a relation over *states*(A) and *states*(B) satisfying $k \cap (\mathit{start}(A) \times \mathit{start}(B)) \neq \emptyset$. (Typically, k will be a forward or a backward simulation.) The *superposition sup*(A, B, k) of B onto A via k is the automaton C defined by

- $\mathit{states}(C) = k$,
- $\mathit{start}(C) = k \cap (\mathit{start}(A) \times \mathit{start}(B))$,
- $\mathit{acts}(C) = \mathit{acts}(A) \cap \mathit{acts}(B)$, and
- for $(s', v'), (s, v) \in \mathit{states}(C)$ and $a \in \mathit{acts}(C)$,

$$(s', v') \xrightarrow{a}_C (s, v) \quad \Leftrightarrow \quad s' \overset{\hat{a}}{\Longrightarrow}_A s \wedge v' \overset{\hat{a}}{\Longrightarrow}_B v.$$

Lemma 6.7 *Suppose* f *is a forward simulation from* A *to* B. *Let* $C = \mathit{sup}(A, B, f)$ *and let* π_1 *and* π_2 *be the projection functions that map states of* C *to their first and second components, respectively. Then* π_1^{-1} *is a history relation from* A *to* C *and* π_2 *is a refinement from* C *to* B.

Theorem 6.8 $A \leq_{\mathrm{F}} B \Leftrightarrow (\exists C : A \leq_{\mathrm{H}} C \leq_{\mathrm{R}} B)$.

Proof: For the implication "⇒", suppose $A \leq_{\mathrm{F}} B$. Let f be a forward simulation from A to B. Take $C = sup(A, B, f)$. The result follows by Lemma 6.7. For the implication "⇐", suppose that $A \leq_{\mathrm{H}} C \leq_{\mathrm{R}} B$. Then $A \leq_{\mathrm{F}} C$ by the definition of history relations, and $C \leq_{\mathrm{F}} B$ because any refinement is a forward simulation. Now $A \leq_{\mathrm{F}} B$ follows by the fact that $\leq_{\mathrm{F}}$ is a preorder. ■

6.2 Prophecy Relations

Now we will present prophecy relations and show that they correspond to backward simulations, very similarly to the way in which history relations correspond to forward simulations.

A relation p over $states(A)$ and $states(B)$ is a *prophecy relation* from A to B if p is a backward simulation from A to B and p^{-1} is a refinement from B to A. We write $A \leq_{\mathrm{P}} B$ if there exists a prophecy relation from A to B, and $A \leq_{\mathrm{iP}} B$ if there is an image-finite prophecy relation from A to B. Thus $A \leq_{\mathrm{iP}} B$ implies $A \leq_{\mathrm{iB}} B$ and $A \leq_{\mathrm{P}} B$, and $A \leq_{\mathrm{P}} B$ implies $A \leq_{\mathrm{B}} B$ and $B \leq_{\mathrm{R}} A$. We give an example of a prophecy relation, using the construction of the *guess* of an automaton. This construction is a kind of dual to the unfolding construction of the previous subsection in that the states contain information about the future rather than about the past.[2]

The *guess* of an automaton A, notation $guess(A)$, is the automaton B defined by

- $states(B) = frag^*(A)$,
- $start(B) = execs^*(A)$,
- $acts(B) = acts(A)$, and
- for $\alpha', \alpha \in states(B)$ and $a \in acts(B)$, $\alpha' \xrightarrow{a}_B \alpha \;\Leftrightarrow\; first(\alpha')\, a\, \alpha = \alpha'$.

Proposition 6.9 $A \leq_{\mathrm{P}} guess(A)$.

Proof: The function *first* which maps each execution fragment of A to its first state is a refinement from $guess(A)$ to A, and the relation $first^{-1}$ is a backward simulation from A to $guess(A)$. ■

Example 6.10 For the automata of Figure 2 we have $C \not\leq_{\mathrm{P}} D$, $D \not\leq_{\mathrm{P}} C$, $E \not\leq_{\mathrm{P}} F$ and $F \leq_{\mathrm{iP}} E$. The difference between $\leq_{\mathrm{P}}$ and $\leq_{\mathrm{iP}}$ is illustrated by the automata of Figure 3: $G \leq_{\mathrm{P}} H$ but $G \not\leq_{\mathrm{iP}} H$. The automata A and B of Figure 1 cannot be used directly to show the difference between $\leq_{\mathrm{P}}$ and $\leq_{\mathrm{iP}}$ since neither $A \leq_{\mathrm{P}} B$ nor $B \leq_{\mathrm{P}} A$. However, we obtain a counterexample by unfolding the B automaton: $A \leq_{\mathrm{P}} unfold(B)$ but $A \not\leq_{\mathrm{iP}} unfold(B)$.

[2] Just as the unfolding operation gives rise to a forest, the guess construction leads to the dual notion of a *backward forest*, i.e., an automaton with the property that for each state there is a unique maximal execution that starts in it. Also, similar to the partial completeness result for backward simulations that requires one of the automata to be a forest, there is a partial completeness result for forward simulations that involves backward forests. Since the guess construction appears to be useful only in proving finite trace inclusion, we decided not to work out the forward/backward duality completely at this point.

Proposition 6.11 *$\leq_{\mathrm{P}}$ and $\leq_{\mathrm{iP}}$ are preorders.*

The following proposition sheds some more light on the relationship between $\leq_{\mathrm{P}}$ and $\leq_{\mathrm{iP}}$.

Proposition 6.12 *Suppose all states of A are reachable, B has fin and $A \leq_{\mathrm{P}} B$. Then $A \leq_{\mathrm{iP}} B$.*

Proof: Let p be a prophecy relation from A to B. Then p is a backward simulation. Now the proof of Prop. 4.20 implies that p is image-finite. Thus p is a bounded prophecy relation and $A \leq_{\mathrm{iP}} B$. ∎

We will now show that prophecy relations capture the essence of prophecy variables, just as history relations capture the essence of history variables.

An automaton B is obtained from an automaton A by *adding a prophecy variable* if there exists a set V such that

- $\mathit{states}(B) \subseteq \mathit{states}(A) \times V$, and
- the relation $\{(s, (s, v)) \mid (s, v) \in \mathit{states}(B)\}$ is a prophecy relation from A to B.

A prophecy variable is *bounded* if the underlying prophecy relation is image-finite.

Proposition 6.13 *Suppose $A \leq_{\mathrm{P}} B$. Then there exists an automaton C that is isomorphic to B and obtained from A by adding a prophecy variable, which is bounded if $A \leq_{\mathrm{iP}} B$.*

Again, we will state all further results in this subsection in terms of relations, and not mention the auxiliary variables any further.

Theorem 6.14 *(Soundness of prophecy relations)*

1. $A \leq_{\mathrm{P}} B \Rightarrow A \equiv_{*\mathrm{T}} B$.
2. $A \leq_{\mathrm{iP}} B \Rightarrow A \equiv_{\mathrm{T}} B$.

Proof: Immediate from the soundness of refinements and backward simulations. ∎

Lemma 6.15 *Suppose b is a backward simulation from A to B. Let $C = \mathit{sup}(A, B, b)$ and let π_1 and π_2 be the projection functions that map states of C to their first and second components, respectively. Then π_1^{-1} is a prophecy relation from A to C and π_2 is a refinement from C to B. If b is image-finite then so is π_1^{-1}.*

Theorem 6.16

1. $A \leq_{\mathrm{B}} B \Leftrightarrow (\exists C : A \leq_{\mathrm{P}} C \leq_{\mathrm{R}} B)$.
2. $A \leq_{\mathrm{iB}} B \Leftrightarrow (\exists C : A \leq_{\mathrm{iP}} C \leq_{\mathrm{R}} B)$.

Proof: The proof of 1 is analogous to that of Theorem 6.8, using Lemma 6.15. 2 can be proved similarly. ■

The following result is dual to Sistla's completeness result.

Theorem 6.17 *(Completeness of prophecy relations and forward simulations)* $A \leq_{*\mathrm{T}} B \Rightarrow \exists C : A \leq_{\mathrm{P}} C \leq_{\mathrm{F}} B$.

Proof:

$$\begin{aligned}
A \leq_{*\mathrm{T}} B &\;\Rightarrow\; \text{(By Theorem 5.13)}\\
A \leq_{\mathrm{BF}} B &\;\Rightarrow\; \text{(By Theorem 5.7)}\\
\exists E : A \leq_{\mathrm{B}} E \leq_{\mathrm{F}} B &\;\Rightarrow\; \text{(By Theorem 6.16)}\\
\exists E, C : A \leq_{\mathrm{P}} C \leq_{\mathrm{R}} E \leq_{\mathrm{F}} B &\;\Rightarrow\; \text{(By Propositions 4.7 and 4.9)}\\
\exists C : A \leq_{\mathrm{P}} C \leq_{\mathrm{F}} B.&
\end{aligned}$$

■

6.3 Completeness of History and Prophecy Relations

We finish this section with versions of the completeness results of Abadi and Lamport [1].

Theorem 6.18 *(Completeness of history relations, prophecy relations and refinements, [1]) Suppose* $A \leq_{*\mathrm{T}} B$. *Then*

1. $\exists C, D : A \leq_{\mathrm{H}} C \leq_{\mathrm{P}} D \leq_{\mathrm{R}} B$, *and*

2. *if* B *has fin then* $\exists C, D : A \leq_{\mathrm{H}} C \leq_{\mathrm{iP}} D \leq_{\mathrm{R}} B$.

Proof: By Sistla's result (Theorem 6.6), there exists an automaton C with $A \leq_{\mathrm{H}} C \leq_{\mathrm{B}} B$. Next, Theorem 6.16 yields the required automaton D with $C \leq_{\mathrm{P}} D \leq_{\mathrm{R}} B$, which proves 1. Now statement 2 is routine. ■

Similarly, we obtain the dual result:

Theorem 6.19 $A \leq_{*\mathrm{T}} B \Rightarrow \exists C, D : A \leq_{\mathrm{P}} C \leq_{\mathrm{H}} D \leq_{\mathrm{R}} B$.

7 Reachability

Whether or not there exists a trace inclusion relation between two automata A and B, is fully determined by the reachable parts of these automata, and so the behavior in the unreachable parts is irrelevant. From the various completeness results we have proved thus far in this paper we know that, at least in theory, our simulation proof techniques can be applied irrespective of whether or not there are unreachable states. Still, several of the individual types of simulation we have discussed are sensitive to the presence of unreachable states. For instance, there exists no refinement from K to L in Figure 5, but if we restrict K to its

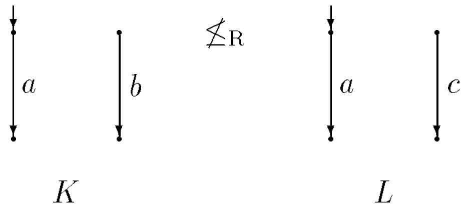

Figure 5: The impact of unreachable states.

reachable part then there is one. For this reason, some of the other work on simulations (e.g., [24]) includes reachability restrictions in the simulation definitions. We have avoided doing this so far, in order to avoid cluttering up our results and proofs. However, we would like to be able to use our results to justify the soundness of methods that allow use of reachability conditions.

In this section, we show how to incorporate reachability into the simulation definitions, and show that the soundness of the resulting simulations follows from the soundness results we have already proved.

7.1 The Reachable Subautomaton

For any automaton A and for any set I of states of A with $I \cap \mathit{start}(A) \neq \emptyset$, let the *subautomaton* of A induced by I, notation A/I, be the automaton defined as follows.

- $\mathit{states}(A/I) = I$,
- $\mathit{start}(A/I) = \mathit{start}(A) \cap I$,
- $\mathit{acts}(A/I) = \mathit{acts}(A)$, and
- $\mathit{steps}(A/I) = \mathit{steps}(A) \cap (I \times \mathit{acts}(A) \times I)$.

For any automaton A, let $\mathit{rstates}(A)$ be the set of reachable states of A. We write $R(A)$ for the *reachable subautomaton* of A, i.e., the automaton $A/\mathit{rstates}(A)$.

Lemma 7.1 $A \leq_{\mathrm{H}} R(A)$.

Proof: The relation $\{(s, s) \mid s \in \mathit{rstates}(A)\}$ is a history relation from A to $R(A)$. ■

In the rest of this subsection we investigate what happens if the various types of simulations are restricted to the reachable subautomata of the automata on which they are defined. Basically, the result we obtain is that the results of such restrictions are again simulation relations.

Lemma 7.2 *Suppose s' is a reachable state of A. Then $s' \overset{\beta}{\Longrightarrow}_A s \Leftrightarrow s' \overset{\beta}{\Longrightarrow}_{R(A)} s$.*

Lemma 7.3

1. *Suppose r is a refinement from A to B. Then $r' = r\lceil \mathit{rstates}(A)$ is a refinement from $R(A)$ to $R(B)$.*
2. *Suppose f is a forward simulation from A to B. Then $f' = f \cap (\mathit{rstates}(A) \times \mathit{rstates}(B))$ is a forward simulation from $R(A)$ to $R(B)$.*
3. *Suppose b is a backward simulation from A to B. Then $b' = b \cap (\mathit{rstates}(A) \times \mathit{rstates}(B))$ is a backward simulation from $R(A)$ to $R(B)$.*
4. *Suppose g is a forward-backward simulation from A to B. Then $g' = g \cap (\mathit{rstates}(A) \times \mathbf{N}(\mathit{rstates}(B)))$ is a forward-backward simulation from $R(A)$ to $R(B)$.*
5. *Suppose g is a backward-forward simulation from A to B. Then*

   $$g' = \{(s, S \cap \mathit{rstates}(B)) \mid s \in \mathit{rstates}(A) \textit{ and } (s, S) \in g\}$$

   is a backward-forward simulation from $R(A)$ to $R(B)$.
6. *Suppose h is a history relation from A to B. Then $h' = h \cap (\mathit{rstates}(A) \times \mathit{rstates}(B))$ is a history relation from $R(A)$ to $R(B)$.*
7. *Suppose p is a prophecy relation from A to B. Then $p' = p \cap (\mathit{rstates}(A) \times \mathit{rstates}(B))$ is a prophecy relation from $R(A)$ to $R(B)$.*

Proof:

1. We first establish that r' is a function from $\mathit{rstates}(A)$ to $\mathit{rstates}(B)$.

   Suppose s is a reachable state of A. Then there exists a finite execution α of A with $\mathit{last}(\alpha) = s$. By induction on the length of α we prove that $r(s)$ is a reachable state of B.

   If α has length 1 then s is a start state of A. Since r is a refinement, this means that $r(s)$ is a start state of B, and hence also a reachable state of B.

   For the induction step, suppose that α has length $n + 1$. Then α has a prefix α' of length n, with a last state s' such that, for some action a, $s' \xrightarrow{a}_A s$. By induction hypothesis, $r(s')$ is a reachable state of B, and because r is a refinement, $r(s') \overset{\hat{a}}{\Longrightarrow}_B r(s)$. Now Lemma 7.2 gives $r(s') \overset{\hat{a}}{\Longrightarrow}_{R(B)} r(s)$. Thus $r(s)$ is a reachable state of B.

   Next we show that r' satisfies the two conditions of a refinement.

   For Condition 1, suppose that $s \in \mathit{start}(R(A))$. Then $s \in \mathit{start}(A)$ and thus $r'(s) \in \mathit{start}(B)$. Hence $r'(s) \in \mathit{start}(R(B))$.

   For Condition 2, suppose that $s' \xrightarrow{a}_{R(A)} s$. Then $s' \xrightarrow{a}_A s$ and both s' and s are reachable. Since r is a refinement, $r'(s') \overset{\hat{a}}{\Longrightarrow}_B r'(s)$. By Lemma 7.2, $r'(s') \overset{\hat{a}}{\Longrightarrow}_{R(B)} r'(s)$.

2. For Condition 1, suppose $s \in \mathit{start}(R(A))$. Then $s \in \mathit{start}(A)$. Since f is a forward simulation, $f[s] \cap \mathit{start}(B) \neq \emptyset$. Thus $f'[s] \cap \mathit{start}(R(B)) = (f[s] \cap \mathit{rstates}(B)) \cap \mathit{start}(B) = f[s] \cap \mathit{start}(B) \neq \emptyset$.

   For Condition 2, suppose $s' \overset{a}{\Longrightarrow}_{R(A)} s$ and $u' \in f'[s']$. Then $u' \in f[s']$ and $u' \in \mathit{rstates}(B)$. Since f is a forward simulation there exists a state $u \in f[s]$ with $u' \overset{\hat{a}}{\Longrightarrow}_B u$. By Lemma 7.2, $u' \overset{\hat{a}}{\Longrightarrow}_{R(B)} u$ and $u \in f'[s]$.

3. We first establish that b' is a total relation over $\mathit{rstates}(A)$ and $\mathit{rstates}(B)$.

   Suppose s is a reachable state of A. Then there exists a finite execution α of A with $\mathit{last}(\alpha) = s$. By induction on the length of α we prove that all states in $b[s]$ are reachable states of B. Since $b[s]$ is nonempty, this implies that $b'[s]$ is nonempty.

   If α has length 1 then s is a start state of A. Since b is a backward simulation, this means that all states in $b[s]$ are start states of B, and hence also reachable states of B.

   For the induction step, suppose that α has length $n + 1$. Then α has a prefix α' of length n, with a last state s' such that, for some action a, $s' \xrightarrow{a}_A s$. Suppose $u \in b[s]$. Because b is a backward simulation, there exists a $u' \in b[s']$ with $u' \overset{\hat{a}}{\Longrightarrow}_B u$. By induction hypothesis, all states in $b[s']$, u in particular, are reachable states of B. Now Lemma 7.2 gives $u' \overset{\hat{a}}{\Longrightarrow}_{R(B)} u$. Thus u is a reachable state of B. Since u has been chosen arbitrarily, it follows that all states in $b[s]$ are reachable states of B.

   Now it is routine to check that b' satisfies the two conditions of a backward simulation.

4. For Condition 1, suppose $s \in \mathit{start}(R(A))$. Then $s \in \mathit{start}(A)$, and so there exists $S \in g[s]$ with $S \subseteq \mathit{start}(B)$. Since all start states are reachable, $S \in g'[s]$, and since $\mathit{start}(B) = \mathit{start}(R(B))$, $S \subseteq \mathit{start}(R(B))$.

   For Condition 2, suppose $s' \xrightarrow{a}_{R(A)} s$ and $S' \in g'[s']$. Then $s' \xrightarrow{a}_A s$ and $S' \in g[s']$. Using that g is a forward-backward simulation, we can find a set $S \in g[s]$ such that for every $u \in S$ there exists $u' \in S'$ with $u' \overset{\hat{a}}{\Longrightarrow}_B u$. Since all states in S' are reachable states of B, it follows by Lemma 7.2 that for every $u \in S$ there exists $u' \in S'$ with $u' \overset{\hat{a}}{\Longrightarrow}_{R(B)} u$. Consequently, all states in S are reachable states of B as well and $S \in g'[s]$.

5. Relation g' is a total relation because relation g is total. Checking that g' satisfies the two conditions of a backward-forward simulation is routine.

6. By (2), h' is a forward simulation from $R(A)$ to $R(B)$, and by (1), h'^{-1} is a refinement from $R(B)$ to $R(A)$. Thus h' is a history relation from $R(A)$ to $R(B)$.

7. By (3), p' is a backward simulation from $R(A)$ to $R(B)$, and by (1), p'^{-1} is a refinement from $R(B)$ to $R(A)$. Thus p' is a prophecy relation from $R(A)$ to $R(B)$.

■

7.2 Weak Simulations

Let $X \in \{\mathrm{R, F, iB, B, iFB, FB, iBF, BF, H, iP, P}\}$, i.e., any of the types of simulation discussed in this paper. Let A and B be automata. We define $A \leq_{\mathrm{wX}} B$ iff $R(A) \leq_{\mathrm{X}} R(B)$.

Proposition 7.4 *The relations $\leq_{\mathrm{wR}}$, $\leq_{\mathrm{wF}}$, $\leq_{\mathrm{wB}}$, $\leq_{\mathrm{wiB}}$, $\leq_{\mathrm{wFB}}$, $\leq_{\mathrm{wiFB}}$, $\leq_{\mathrm{wBF}}$, $\leq_{\mathrm{wH}}$, $\leq_{\mathrm{wP}}$ and $\leq_{\mathrm{wiP}}$ are all preorders. (However, $\leq_{\mathrm{wiBF}}$ is not a preorder.)*

Proof: Immediate from the definitions, since the corresponding strong[3] simulations are preorders. The counterexample of Figure 4, which we presented to demonstrate that $\leq_{\mathrm{iBF}}$ is not a preorder, only involves automata in which all states are reachable. Therefore the same example can also be used to show that $\leq_{\mathrm{wiBF}}$ is not a preorder. ■

Between the weak simulation relations we have exactly the same inclusion relations as between the corresponding strong simulations.

Proposition 7.5 *Let $X, Y \in \{R, F, iB, B, iFB, FB, iBF, BF, H, iP, P\}$. Then*

$$(\forall A, B : A \leq_{\mathrm{X}} B \Rightarrow A \leq_{\mathrm{Y}} B) \;\Leftrightarrow\; (\forall A, B : A \leq_{\mathrm{wX}} B \Rightarrow A \leq_{\mathrm{wY}} B).$$

Proof: "$\Rightarrow$" Immediate from the definitions.

"$\Leftarrow$" Follows from the observation that none of the counterexamples that we presented to prove the difference between the strong simulations involved unreachable states. ■

Each of the weak relations is at least as coarse as the strong relation from which it is derived.

Proposition 7.6 *Suppose $X \in \{R, F, iB, B, iFB, FB, iBF, BF, H, iP, P\}$. Then $A \leq_{\mathrm{X}} B \Rightarrow A \leq_{\mathrm{wX}} B$.*

Proof: Suppose $A \leq_{\mathrm{X}} B$. Then there exists an X-mapping m from A to B. By Lemma 7.3, the 'restriction' of m to the states of $R(A)$ and $R(B)$ is an X-mapping from $R(A)$ to $R(B)$. Thus $R(A) \leq_{\mathrm{X}} R(B)$, and hence $A \leq_{\mathrm{wX}} B$. ■

Some of the weak preorders are strictly coarser than their corresponding strong versions:

Proposition 7.7 *$\leq_{\mathrm{wR}}$, $\leq_{\mathrm{wiB}}$, $\leq_{\mathrm{wB}}$, $\leq_{\mathrm{wH}}$, $\leq_{\mathrm{wiP}}$ and $\leq_{\mathrm{wP}}$ are coarser than $\leq_{\mathrm{R}}$, $\leq_{\mathrm{iB}}$, $\leq_{\mathrm{B}}$, $\leq_{\mathrm{H}}$, $\leq_{\mathrm{iP}}$ and $\leq_{\mathrm{P}}$, respectively.*

Proof: The automata K and L of Figure 5 are related by all of the weak preorders but by none of the strong ones. In combination with Prop. 7.6, this gives the desired result. ■

On the other hand, several weak relations coincide with their originals.

Proposition 7.8 *Suppose $X \in \{F, iFB, FB, iBF, BF\}$. Then $A \leq_{\mathrm{X}} B \Leftrightarrow A \leq_{\mathrm{wX}} B$.*

[3] Note that our use of the words 'weak' and 'strong' in this subsection differs from that by Milner [25], who uses it to indicate whether or not internal steps are abstracted away.

Proof: By Prop. 7.6, we only have to worry about the implication "$\Leftarrow$".

- X=F.
$$\begin{array}{rcl} A \le_{\mathrm{wF}} B & \Rightarrow & \\ R(A) \le_{\mathrm{F}} R(B) & \Rightarrow & \text{(By Lemma 7.1)} \\ A \le_{\mathrm{H}} R(A) \le_{\mathrm{F}} R(B) \ge_{\mathrm{H}} B & \Rightarrow & \text{(By basic properties of } \le_{\mathrm{H}}) \\ A \le_{\mathrm{F}} R(A) \le_{\mathrm{F}} R(B) \le_{\mathrm{F}} B & \Rightarrow & \text{(By Prop. 4.9)} \\ A \le_{\mathrm{F}} B & & \end{array}$$

- X=iFB.
$$\begin{array}{rcl} A \le_{\mathrm{wiFB}} B & \Rightarrow & \text{(By Lemma 7.1)} \\ A \le_{\mathrm{H}} R(A) \le_{\mathrm{iFB}} R(B) \ge_{\mathrm{H}} B & \Rightarrow & \text{(By Theorem 5.1)} \\ \exists C : A \le_{\mathrm{H}} R(A) \le_{\mathrm{F}} C \le_{\mathrm{iB}} R(B) \ge_{\mathrm{H}} B & \Rightarrow & \text{(By basic properties of } \le_{\mathrm{H}}) \\ \exists C : A \le_{\mathrm{F}} R(A) \le_{\mathrm{F}} C \le_{\mathrm{iB}} R(B) \le_{\mathrm{iB}} B & \Rightarrow & \text{(By Prop. 4.9 and Prop. 4.16)} \\ \exists C : A \le_{\mathrm{F}} C \le_{\mathrm{iB}} B & \Rightarrow & \text{(By Theorem 5.1)} \\ A \le_{\mathrm{iFB}} B & & \end{array}$$

- X=FB. Analogous to case X=iFB.

- X=iBF. Suppose $A \le_{\mathrm{wiBF}} B$. Then $R(A) \le_{\mathrm{iBF}} R(B)$. Let g be an image-finite backward-forward simulation from $R(A)$ to $R(B)$. Define

  $$g' = g \cup \{(s, \emptyset) \mid s \in \mathit{states}(A) - \mathit{rstates}(A)\}.$$

  It is routine to check that g' is an image-finite backward-forward simulation from A to B. Thus $A \le_{\mathrm{iBF}} B$.

- X=BF. Analogous to case X=iBF.

■

The following proposition completes our classification of weak simulations.

Proposition 7.9

1. $A \le_{\mathrm{wR}} B \Rightarrow A \le_{\mathrm{F}} B$.

2. $A \le_{\mathrm{wH}} B \Rightarrow A \le_{\mathrm{F}} B$.

3. $A \le_{\mathrm{wiB}} B \Rightarrow A \le_{\mathrm{iBF}} B$.

4. $A \le_{\mathrm{wB}} B \Rightarrow A \le_{\mathrm{BF}} B$.

5. $A \le_{\mathrm{wB}} B$ *and* B *has fin* $\Rightarrow A \le_{\mathrm{wiB}} B$.

6. $A \le_{\mathrm{wP}} B$ *and* B *has fin* $\Rightarrow A \le_{\mathrm{wiP}} B$.

Proof: Easy. Cases (5) and (6) follow by Prop. 4.20 and Prop. 6.12, respectively. ■

The fact that in Prop. 7.9 the reverse implications do not hold in general follows from the observation that none of the counterexamples that we used to illustrate the difference between the relevant strong simulations involved unreachable states.

Combination of the above results gives that the weak simulations relations provide us with sound techniques for proving trace inclusion relations between automata.

Theorem 7.10 *(Soundness of weak simulations)*

1. *Suppose $X \in \{R, F, iB, iFB, iBF, H, iP\}$. Then $A \leq_{\mathrm{wX}} B \Rightarrow A \leq_{\mathrm{T}} B$.*
2. *Suppose $X \in \{B, FB, BF, P\}$. Then $A \leq_{\mathrm{wX}} B \Rightarrow A \leq_{*\mathrm{T}} B$.*

Proof: Propositions 7.5, 7.8 and 7.9 imply that each of the weak simulation relations is equal to or included in a (sound) strong simulation. ■

7.3 □-properties

In practice, it is often difficult to give an exact characterization of the set of reachable states of an automaton. However, in most cases it is possible to identify a proper subset of the state set that includes all reachable states.

Define a □-property of an automaton A to be any property that is always true, i.e., a subset of *states*(A) that includes all the reachable states of A. In practice, the fact that a □-property includes every reachable state is normally proved by induction on the length of a finite execution that leads to the state. Often, □-properties will be *invariants* of the automaton, i.e., properties that hold initially and are preserved by transitions.

Note that *rstates*(A) is the smallest □-property of A, and that *states*(A) is the largest □-property of A. We also have the following trivial technical lemma.

Lemma 7.11 *Let I be a □-property of A. Then $R(A/I) = R(A)$.*

If we want to establish that $A \leq_{\mathrm{wX}} B$, for given automata A and B, then the following theorem will often allow us to prove this even if we do not know exactly what the reachable states are.

Proposition 7.12 *Suppose $X \in \{R, F, iB, B, iFB, FB, iBF, BF, H, iP, P\}$. Then $A \leq_{\mathrm{wX}} B$ iff there exist □-properties I_A and I_B of A and B, respectively, such that $A/I_A \leq_{\mathrm{X}} B/I_B$.*

Proof: "⇒". If $A \leq_{\mathrm{wX}} B$, then choosing $I_A = rstates(A)$ and $I_B = rstates(B)$ yields $A/I_A = R(A) \leq_{\mathrm{X}} R(B) = B/I_B$.

"⇐". Here we discuss the case $X = R$. The other cases are completely analogous, and left to the reader.

Suppose that $A/I_A \leq_{\mathrm{R}} B/I_B$, for some I_A and I_B. Then there exists a a refinement r from A/I_A to B/I_B. By Lemma 7.3(1), $r' = r\lceil rstates(A/I_A)$ is a refinement from $R(A/I_A)$ to

$R(B/I_B)$. By Lemma 7.11, r' is also a refinement from $R(A)$ to $R(B)$. Thus $R(A) \leq_{\mathrm{R}} R(B)$, and hence $A \leq_{\mathrm{wX}} B$. ■

The significance of Prop. 7.12 is that in order to prove a trace inclusion relation between automata A and B, it suffices to establish a simulation relation between the subautomata A/I_A and B/I_B induced by any $\Box$-properties I_A and I_B. In practice, we will often have some freedom in the choice of the I_A and I_B, and there is a trade-off between the amount of work needed to find small $\Box$-properties, and the amount of work to establish a simulation.

7.4 Direct Definitions of Weak Simulations

For the convenience of those readers who would like to use our simulation techniques in actual verifications, we will now present direct definitions of the various weak simulation relations that do not refer to subautomata. These definitions are the same as the ones for ordinary simulations except that permission is explicitly given to use $\Box$-properties at appropriate places in the proof.

Let A and B be automata with $\Box$-properties I_A and I_B, respectively.

A *weak refinement* from A to B, with respect to I_A and I_B, is a function r from $states(A)$ to $states(B)$ that satisfies the following two conditions:

1. If $s \in start(A)$ then $r(s) \in start(B)$.
2. If $s' \xrightarrow{a}_A s$, $s', s \in I_A$, and $r(s') \in I_B$, then $r(s') \overset{\hat{a}}{\Longrightarrow}_B r(s)$.

A *weak forward simulation* from A to B, with respect to I_A and I_B, is a relation f over $states(A)$ and $states(B)$ that satisfies:

1. If $s \in start(A)$ then $f[s] \cap start(B) \neq \emptyset$.
2. If $s' \xrightarrow{a}_A s$, $s', s \in I_A$, and $u' \in f[s'] \cap I_B$, then there exists a state $u \in f[s]$ such that $u' \overset{\hat{a}}{\Longrightarrow}_B u$.

A *weak backward simulation* from A to B, with respect to I_A and I_B, is a relation b over $states(A)$ and $states(B)$ that satisfies:

1. If $s \in start(A)$ then $b[s] \cap I_B \subseteq start(B)$.
2. If $s' \xrightarrow{a}_A s$, $s', s \in I_A$, and $u \in b[s] \cap I_B$, then there exists a state $u' \in b[s'] \cap I_B$ such that $u' \overset{\hat{a}}{\Longrightarrow}_B u$.
3. If $s \in I_A$ then $b[s] \cap I_B \neq \emptyset$.

A *weak forward-backward simulation* from A to B, with respect to I_A and I_B, is a relation g over $states(A)$ and $\mathbf{P}(states(B))$ that satisfies:

1. If $s \in start(A)$ then there exists $S \in g[s]$ such that $S \cap I_B \subseteq start(B)$.
2. If $s' \xrightarrow{a}_A s$, $s', s \in I_A$ and $S' \in g[s']$, then there exists a set $S \in g[s]$ such that for every $u \in S \cap I_B$ there exists $u' \in S' \cap I_B$ with $u' \overset{\hat{a}}{\Longrightarrow}_B u$.

3. If $s \in I_A$ and $S \in g[s]$ then $S \cap I_B \neq \emptyset$.

A *weak backward-forward simulation* from A to B, with respect to I_A and I_B, is a relation g over $states(A)$ and $\mathbf{P}(states(B))$ that satisfies:

1. If $s \in start(A)$ then, for all $S \in g[s]$, $S \cap start(B) \neq \emptyset$.
2. If $s' \xrightarrow{a}_A s$, $s', s \in I_A$ and $S \in g[s]$, then there exists a set $S' \in g[s']$ such that for every $u' \in S' \cap I_B$ there exists a $u \in S \cap I_B$ with $u' \overset{\hat{a}}{\Longrightarrow}_B u$.
3. If $s \in I_A$ then $g[s] \neq \emptyset$.

A relation h over $states(A)$ and $states(B)$ is a *weak history relation* from A to B, with respect to I_A ad I_B, if h is a weak forward simulation from A to B, with respect to I_A and I_B, and h^{-1} is a weak refinement from B to A, with respect to I_B and I_A.

A relation p over $states(A)$ and $states(B)$ is a *weak prophecy relation* from A to B, with respect to I_A and I_B, if p is a weak backward simulation from A to B, with respect to I_A and I_B, and p^{-1} is a weak refinement from B to A, with respect to I_B and I_A.

Proposition 7.13 *Let $X \in \{R, F, iB, B, iFB, FB, iBF, BF, H, iP, P\}$. Then there exists a weak X-mapping from A to B, with respect to some $\Box$-properties I_A and I_B, iff $A \leq_{\mathrm{wX}} B$.*

Proof: "$\Leftarrow$" Suppose $A \leq_{\mathrm{wX}} B$. Then there exists an X-mapping m from $R(A)$ to $R(B)$. It is straightforward to check that m is a weak X-mapping from A to B, with respect to $rstates(A)$ and $rstates(B)$.

"$\Rightarrow$" Suppose that m is a weak X-mapping from A to B, with respect to I_A and I_B. By almost literally copying the proof of Lemma 7.3, one can show that the 'restriction' of m to the states in $rstates(A)$ and $rstates(B)$ is an X-mapping from $R(A)$ to $R(B)$.

The only case where we can not directly copy Lemma 7.3 is the one in which m is a weak forward-backward simulation. In that case define

$$m' = \{(s, S) \in rstates(A) \times \mathbf{P}(rstates(B)) \mid \exists S' \in m[s] : S = S' \cap I_B\}.$$

We show that m' is a forward-backward simulation from $R(A)$ to $R(B)$.

First note that $m' \subseteq rstates(A) \times \mathbf{N}(rstates(B))$ by the third clause in the definition of a weak forward-backward simulation.

For Condition 1, suppose $s \in start(R(A))$. Then $s \in start(A)$ and there exists $S \in m[s]$ with $S \cap I_B \subseteq start(B)$. Now let $S' = S \cap I_B$. Then $S' \in m'[s]$ and $S' \subseteq start(R(B))$.

For Condition 2, suppose $s' \xrightarrow{a}_{R(A)} s$ and $S' \in m'[s']$. Then $s' \xrightarrow{a}_A s$ and there exists $S'' \in m[s']$ such that $S' = S'' \cap I_B$. Using that m is a weak forward-backward simulation, we can find a set $S \in g[s]$ such that for every $u \in S \cap I_B$ there exists $u' \in S'$ with $u' \overset{\hat{a}}{\Longrightarrow}_B u$. Since all states in S' are reachable states of B, it follows by Lemma 7.2 that for every $u \in S \cap I_B$ there exists $u' \in S'$ with $u' \overset{\hat{a}}{\Longrightarrow}_{R(B)} u$. Consequently, all states in $S \cap I_B$ are reachable states of B as well and so $S \cap I_B \in m'[s]$. ■

8 Conclusions

In this paper, we have given a comprehensive presentation of simulation proof methods for untimed automata, including refinements, forward and backward simulations and combinations thereof, and history and prophecy relations. We have given basic results for all of these simulations, including soundness and completeness results.

We can summarize the basic implications between the various simulation techniques of this paper as follows. Suppose $X, Y \in \{\mathrm{T}, *\mathrm{T}, (\mathrm{w})\mathrm{R}, (\mathrm{w})\mathrm{F}, (\mathrm{w})(\mathrm{i})\mathrm{B}, (\mathrm{w})(\mathrm{i})\mathrm{FB}, (\mathrm{w})(\mathrm{i})\mathrm{BF}, (\mathrm{w})\mathrm{H}, (\mathrm{w})(\mathrm{i})\mathrm{P}\}$ (where (z)Z stands for either Z or zZ). Then $A \leq_{\mathrm{X}} B \Rightarrow A \leq_{\mathrm{Y}} B$ for all automata A and B if and only if there is a path from $\leq_{\mathrm{X}}$ to $\leq_{\mathrm{Y}}$ in Figure 6 consisting of thin lines only. If B has fin, then $A \leq_{\mathrm{X}} B \Rightarrow A \leq_{\mathrm{Y}} B$ for all automata A and B if and only if there is a path from $\leq_{\mathrm{X}}$ to $\leq_{\mathrm{Y}}$ consisting of thin lines and thick lines.

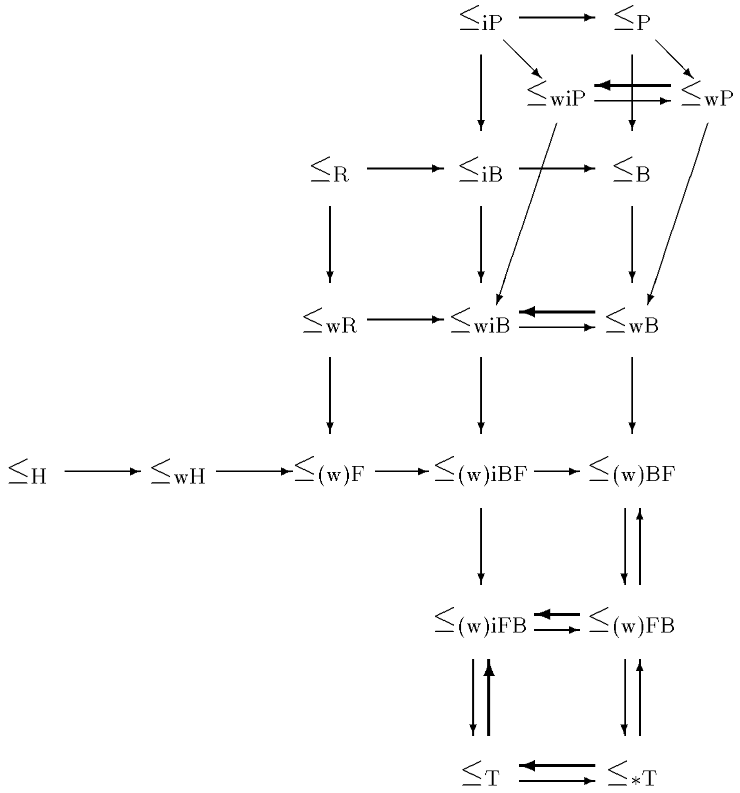

Figure 6: Classification of basic relations between automata.

Refinements and forward simulations have already been used extensively and successfully for verifying concurrent algorithms, and backward simulations (in the form of prophecy variables) have also been shown to be of practical value in a few cases. Additional work remains to determine the practical utility of backward simulations, the hybrid methods, and the history and prophecy relations of this paper. This will involve applying these techniques to a wide range of examples.

It remains to exploit these methods much further in producing formal proofs for algorithms of practical importance.

All the automata studied in this paper have been untimed. In Part II, we extend the simulation definitions and the results of this paper to timed systems.

References

[1] M. Abadi and L. Lamport. The existence of refinement mappings. *Theoretical Computer Science*, 2(82):253–284, 1991.

[2] J.W. de Bakker, W.P. de Roever, and G. Rozenberg, editors. *REX Workshop on Stepwise Refinement of Distributed Systems: Models, Formalism, Correctness,* Mook, The Netherlands 1989, volume 430 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990.

[3] R. Gerth. Foundations of compositional program refinement (first version). In de Bakker et al. [2], pages 777–808.

[4] A. Ginzburg. *Algebraic Theory of Automata*. Academic Press, New York – London, 1968.

[5] J. He. Process simulation and refinement. *Journal of Formal Aspects of Computing Science*, 1:229–241, 1989.

[6] C.A.R. Hoare. Proof of correctness of data representations. *Acta Informatica*, 1:271–281, 1972.

[7] C.A.R. Hoare, J. He, and J.W. Sanders. Prespecification in data refinement. *Information Processing Letters*, 25:71–76, 1987.

[8] B. Jonsson. *Compositional Verification of Distributed Systems*. PhD thesis, Department of Computer Systems, Uppsala University, 1987. DoCS 87/09.

[9] B. Jonsson. Modular verification of asynchronous networks. In PODC 87 [28], pages 152–166.

[10] B. Jonsson. On decomposing and refining specifications of distributed systems. In de Bakker et al. [2], pages 361–387.

[11] B. Jonsson. Simulations between specifications of distributed systems. In J.C.M. Baeten and J.F. Groote, editors, *Proceedings CONCUR 91,* Amsterdam, volume 527 of *Lecture Notes in Computer Science*, pages 346–360. Springer-Verlag, 1991.

[12] M.B. Josephs. A state-based approach to communicating processes. *Distributed Computing*, 3:9–18, 1988.

[13] N. Klarlund and F.B. Schneider. Verifying safety properties using infinite-state automata. Technical Report 89-1039, Department of Computer Science, Cornell University, Ithaca, New York, 1989.

[14] N. Klarlund and F.B. Schneider. Proving nondeterministically specified safety properties using progress measures, August 1991. To appear in *Information and Computation*.

[15] D.E. Knuth. *Fundamental Algorithms*, volume 1 of *The Art of Computer Programming*. Addison-Wesley, Reading, Massachusetts, 1973. Second edition.

[16] L. Lamport. Specifying concurrent program modules. *ACM Transactions on Programming Languages and Systems*, 5(2):190–222, 1983.

[17] B. Lampson, N. Lynch, and J. Søgaard-Andersen. Correctness of at-most-once message delivery protocols, 1993. Submitted for publication.

[18] B.L. Liskov and J.V. Guttag. *Abstraction and Specification in Program Development*. MIT Press and McGraw Hill, 1986.

[19] N.A. Lynch. Concurrency control for resilient nested transactions. Report TR-285, MIT, February 1983.

[20] N.A. Lynch. Multivalued possibilities mappings. In de Bakker et al. [2], pages 519–543.

[21] N.A. Lynch and M.R. Tuttle. Hierarchical correctness proofs for distributed algorithms. In PODC 87 [28], pages 137–151. A full version is available as MIT Technical Report MIT/LCS/TR-387.

[22] N.A. Lynch and F.W. Vaandrager. Forward and backward simulations for timing-based systems. In J.W. de Bakker, C. Huizing, W.P. de Roever, and G. Rozenberg, editors, *Proceedings of the REX Workshop "Real-Time: Theory in Practice"*, volume 600 of *Lecture Notes in Computer Science*, pages 397–446. Springer-Verlag, 1992.

[23] N.A. Lynch and F.W. Vaandrager. Forward and backward simulations – part II: Timing-based systems, 1993. In preparation.

[24] M. Merritt. Completeness theorems for automata. In de Bakker et al. [2], pages 544–560.

[25] R. Milner. *Communication and Concurrency*. Prentice-Hall International, Englewood Cliffs, 1989.

[26] S. Owicki and D. Gries. An axiomatic proof technique for parallel programs. *Acta Informatica*, 6(4):319–340, 1976.

[27] D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, 5^{th} *GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[28] *Proceedings of the* 6^{th} *Annual ACM Symposium on Principles of Distributed Computing*, August 1987.

[29] A.P. Sistla. Proving correctness with respect to nondeterministic safety specifications. *Information Processing Letters*, 39(1):45–49, July 1991.

[30] E.W. Stark. Proving entailment between conceptual state specifications. *Theoretical Computer Science*, 56:135–154, 1988.