99

LA-UR- 01-1687 c.1

Approved for public release; distribution is unlimited.

Title: COMPLEXITY OF ANALYSIS & VERIFICATION PROBLEMS FOR COMMUNICATING AUTOMATA & DISCRETE DYNAMICAL SYSTEMS

Author(s):

Harry B. Hunt III, D-2
Daniel J. Rosendrantz, SUNY, Albany, NY
Christopher L. Barrett, D-2
Madhav V. Marathe, D-2
S. S. Ravi, SUNY, Albany, NY

Submitted to:

Concur "01 - U. Aalborg, Department of Computer Science
Denmark
August 21, 2001

Los Alamos
NATIONAL LABORATORY

Complexity of Analysis and Verification Problems for Communicating Automata and Discrete Dynamical Systems

HARRY B. HUNT III [2,3] DANIEL J. ROSENKRANTZ [2] CHRIS BARRETT [1]
MADHAV V. MARATHE [1] S. S. RAVI [2,3]

March 20, 2001

Abstract

We identify several *simple* but powerful concepts, techniques, and results; and we use them to characterize the complexities of a number of basic problems Π, that arise in the analysis and verification of of the following models $\mathcal{M}$ of communicating automata and discrete dynamical systems:

> *systems of communicating automata* including both finite and infinite cellular automata, *transition systems*, *discrete dynamical systems*, and *succinctly-specified finite automata*.

These concepts, techniques, and results are centered on the following: (i) reductions of STATE-REACHABILITY problems, especially for very simple systems of communicating copies of a single simple finite automaton, (ii) reductions of generalized CNF satisfiability problems [Sc78], especially to very simple communicating systems of copies of a few basic *acyclic* finite sequential machines, and (iii) reductions of the EMPTINESS and EMPTINESS-OF-INTERSECTION problems, for several kinds of regular set descriptors.

For systems of communicating automata and transition systems, the problems studied include: *all equivalence relations and simulation preorders in the Linear-time/ Branching-time hierarchies of equivalence relations and simulation preorders of [vG90, vG93]*, both without and with the *hiding* abstraction. For discrete dynamical systems, the problems studied include *the* INITIAL *and* BOUNDARY VALUE PROBLEMS*(denoted* IVPS *and* BVPS, *respectively), for nonlinear difference equations* over many different algebraic structures, e.g. all unitary rings, all finite unitary semirings, and all lattices. For succinctly-specified finite automata, the problems studied also include *the several problems studied in [AY98]*, e.g. the EMPTINESS, EMPTINESS-OF-INTERSECTION, EQUIVALENCE and CONTAINMENT problems.

The concepts, techniques, and results presented *unify* and *significantly extend* many of the known results in the literature, e.g. [Wo86, Gu89, BPT91, GM92, Ra92, HT94, SH+96, AY98, AKY99, RH93, SM73, Hu73, HRS76, HR78], for communicating automata including both finite and infinite cellular automata and for finite automata specified by special kinds of context-free grammars, by regular operations augmented with *squaring* and *intersection*, and specified succinctly as in [AY98, AKY99].

Moreover, our development of these concepts, techniques, and results shows how several ideas, techniques, and results, for the individual models $\mathcal{M}$ above can be extended to apply to all or to most of these models. As one example of this and paraphrasing [BPT91] , we show:

> Most of these models $\mathcal{M}$ exhibit *computationally-intractable sensitive dependence on initial conditions*, for the *same* reason. These *computationally-intractable sensitivities* range from **PSPACE**-*hard* to *undecidable*.

[1] Los Alamos National Laboratory, MS M997, P.O. Box 1663, Los Alamos, NM 87545. Email: {barrett, marathe}@lanl.gov. The work is supported by the Department of Energy under Contract W-7405-ENG-36.

[2] Department of Computer Science, University at Albany - SUNY, Albany, NY 12222. Email addresses of authors: {hunt, ravi, djr, res}@cs.albany.edu. Supported by a grant from Los Alamos National Laboratory and by NSF Grant CCR-97-34936.

[3] Part of the work was done while the authors were visiting the Basic and Applied Simulation Sciences Group (TSA-2) of the Los Alamos National Laboratory.

1 Introduction, motivation, and overview of results

A number of researchers, e.g. [Wo86, Gu89, BPT91, GM92, Ra92, HT94, SH+96, AY98, AKY99, RH93, SM73, Hu73, HRS76, HR78], have studied *separately* the computational complexity of various problems, for finite networks of communicating automata, infinite cellular automata, both finite and infinite transition systems, discrete dynamical systems, sequential digital circuits, regular sets specified by regular operations augmented with *squaring* [4] and *intersection*, and succinctly-specified finite automata. Here in contrast, we study *simultaneously* these different models $\mathcal{M}$ with the following four goals guiding this study:

1. We want to identify ideas, concepts, techniques, etc., that apply *naturally* to all or most of these models.

2. Moreover when possible, we want to identify general techniques, etc., that apply to all or most of these models, when instances are specified succinctly, especially hierarchically [Ga82, BOW83, GW83, Le86, LW88, LW92], periodically/dynamically [KMW67, Or84, CM93, KO91, MH+98], and/or by parallel composition [Ho84, GM92, Ra92, SH+96, AY98].

3. We want to develop proof techniques, that extend *naturally* in the limit to apply to infinite cellular automata of [Wo86, Gu89], discrete dynamical systems over continuous algebraic structures such as the reals, and if possible continuous dynamical systems [Ro99]. Our reasons here are as follows: There has been extensive mathematical research on both infinite cellular automata and on continuous dynamical systems. We want to determine which concepts or techniques from this research can be ported so as to apply to finite networks of communicating finite automata, transitions systems, and finite discrete dynamical systems.

4. We want to identify ideas, concepts, techniques, etc., from the literature of one of these models, that can be extended (preferably mechanically) to apply to all or most of the other models.

In this paper, we emphasize those concepts that can be used to characterize the *complexities* of the analysis and verification of these models, as discussed in the Abstract above. Additionally following [BPT91], we emphasize concepts, techniques, etc., that can be used to characterize the *computationally- tractable* or *computationally-intractable sensitivity to initial values* of these models.

The actual concepts, techniques, and general results identified and/or developed include the following:

1. general efficient reductions of STATE-REACHABILITY problems to *all* equivalence relations and simulation preorders between the COMPUTATIONAL-IDENTITY [5] and the TRACE-EQUIVALENCE and TRACE-CONTAINMENT problems. These relations include *all* relations in the Linear-time/Branching-time hierarchies of [vG90, vG93]; and thus, they include BISIMULATION-EQUIVALENCE, 2-NESTED-SIMULATION-EQUIVALENCE, READY-SIMULATION-EQUIVALENCE, SIMULATION-EQUIVALENCE, FAILURES-EQUIVALENCE, COMPLETED-TRACE- EQUIVALENCE, TRACE-EQUIVALENCE, etc.;

2. results from [RH93] that the STATE-REACHABILITY problems are already **DSPACE**(n)- and **EXSPACE**-*hard*, respectively, for systems of linearly inter-connected and for hierarchically-specified systems of linearly inter-connected copies of *one* particularly simple deterministic finite automaton;

[4] Intuitively, *squaring* means that a language R^n can be represented by $(D_R)^\nu$, where D_R is a language descriptor specifying the language R and ν is the binary numeral without leading zeros denoting the nonnegative integer n.

[5] Intuitively, two systems are *computationally-identical* if, given common input(s), they execute exactly the *same* sequences of computational actions and state transitions.

3. general efficient reductions of GENERALIZED CNF SATISFIABILITY PROBLEMS [Sc78], especially of the problem EXACTLY1-EX3MONOTONESAT [6], to the STATE-REACHABILITY problems for several kinds of communicating *acyclic* finite sequential machines;

4. direct highly efficient translations of finite systems of linearly- interconnected copies of the above simple finite automaton into intuitively equivalent systems of nonlinear difference equations and into intuitively equivalent nonlinear difference equations, over any algebraic structure with *monotone-logic expressibility*[7]

   (These algebraic structures include all *unitary* rings, all finite *unitary* semi-rings, all lattices, and all fixed-precision discretizations of the integers, rationals, reals, complex numbers, etc.); and

5. general efficient reductions of the EMPTINESS and EMPTINESS-OF- INTERSECTION problems, for several types of regular set descriptors as developed in [SM73, Hu73, HRS76, HR77] into a number of basic computational problems, for the succinctly-specified finite automata **HSMs** and **CHSMs** of [AY98, AKY99].

Intuitively, these reductions are usually by *local-replacement* [GJ79]. Formally, these reductions can be shown (see [MH+98, HSM00, HMS01]) *both* (i) to be ultra-efficient in both sequential and parallel computing resources and (ii) to extend directly to efficient reductions when problem instances are specified succinctly using the hierarchical and/or dynamic/periodic specifications referenced above.

1.1 Models and problems considered

The models $\mathcal{M}$ considered here include the following:

1. finite cellular automata (**FCA**), finite graph automata (**FGA**), finite networks of finite-state machines communicating by explicit channels (**CFSMs**), and finite networks of sequential machines communicating using parallel composition (**CSMs**) [Ho84, Ra92, SH+96],

2. systems of nonlinear difference equations with constant coefficients over any abstract algebraic structures with *monotone-logic expressibility*,

3. 1- or 2-dimensional finite or infinite systems of communicating finite automata inter-connected linearly or in simple regular bounded-grid patterns, including 1- and 2-dimensional **CA** defined as in [Wo86, Gu89]; and

4. (non)deterministic finite and infinite state automata represented by hierarchically- or dynamically-/periodically-specified state-transition diagrams, possibly augmented with *parallel composition* [AY98, AKY99].

Depending upon the model $\mathcal{M}$, the problems Π considered here include

[6] That is the problem of determining if there is an assignment of truth-values to the variables of a 3CNF formula in which, all clauses consist of exactly 3 non-negated literals, that satisfies exactly one literal per clause.

[7] An algebraic structure **F** has *monotone-logic expressibility* if there exist distinct elements a,b of **F** and functions f_1,f_2 expressible by the operations of **F** such that

1. $f_1(a,a) = a$ and $f_1(a,b) = f_1(b,a) = f_1(b,b) = b$.
2. $f_2(a,a) = f_2(a,b) = f_2(b,a) = a$ and $f_2(b,b) = b$.

That is the functions f_1,f_2, when restricted to $\{a,b\}$, are isomorphic to *or* and *and* applied to $\{0,1\}$.

the STATE-REACHABILITY, FIXED-POINT-REACHABILITY, EQUIVALENCE, CONTAINMENT, EMPTINESS-OF-INTERSECTION, BISIMULATION EQUIVALENCE, WEAK-BISIMULATION EQUIVALENCE, INITIAL-VALUE and BOUNDARY-VALUE Problems (denoted by IVPS and BVPS), and all relations in the Linear-time/branching-time hierarchies of equivalence relations and preorders of [vG90, vG93].

For the models $\mathcal{M}$ of **1**, **3**, and **4**, we consider instances specified standardly, hierarchically or dynamically/periodically (by the various specifications referenced above); and for the models $\mathcal{M}$ of **2**, we consider both *narrow* and *wide* specifications of one or of two independent variables.

2 Summary of particular results obtained and their significance

The following particular results obtained here are direct corollaries of the general ideas, concepts, and general results outlined in Section 1:

1. the **NSPACE(n)**-*hardness* results, for problems for **CSMs** with or without *hiding* in [Ra92, SH+96] and the *new* results that each of these problems is **EXSPACE**-*hard*, for *both* hierarchically-specified networks of linearly-interconnected networks of finite automata communicating over explicit channels and hierarchically-specified **CSMs** with compatible succinct specification of action symbols;

2. the *new* results that all equivalence relations and simulation preorders in the Linear-time/Branching-time hierarchy are **coNDEXPTIME**-*hard*, even for succinctly-specified *acyclic* 2-dimensional periodically specified **FCAs**[8], and all such equivalence relations and simulation preorders are **PSPACE**-*hard*, even for hierarchically- specified *acyclic* **CSMs**, with compatible succinct specification of action symbols

   (To our knowledge, these are the first such *hardness* results, for *acyclic* succinctly-specified 2-dimensional communicating finite automata or for *acyclic* hierarchically-specified **CSMs**.);

3. the *new* results that various analysis questions are **DSPACE**(n)- and **EXSPACE**-*hard*, for *narrow* and for *wide* nonlinear difference equations with constant-coefficients, respectively, on any algebraic structure with *monotone-logic expressibility*

   (These *hard* analysis questions include the IVPS and BVPS, as well as the discrete analogues of a number of the qualitative questions about the *phase spaces* of continuous dynamical systems studied in the literature of dynamical systems and chaos [Ro99].);

4. the *new* results that, when extended to apply to the **CHSMs** of [AY98, AKY99], all equivalence relations and simulation preorders in the Linear-time/ Branching-time Hierarchies of [vG90, vG93] are **EXSPACE**-*hard*

   (Of these relations and preorders, only TRACE-EQUIVALENCE and TRACE-CONTAINMENT were considered in [AY98].);

5. a number of *new* results, for very simple classes of the **HSMs** and **CHSMs** of [AY98, AKY99] including the following:

   (a) the EMPTINESS-OF-INTERSECTION problem is already **PSPACE**-*hard*, for pairs of *acyclic* **HSMs**,

[8] see [MH+98] for definition of 2-dimensional periodic specifications

(b) the EMPTINESS problems are already PSPACE-*hard*, for **CHSMs** consisting of the parallel composition of a pair of *acyclic* **HSMs** and for **CHSMs** consisting of the parallel composition of a finite number of deterministic finite automata,

(c) the EQUIVALENCE and CONTAINMENT problems are already **coNDEXPTIME**-*complete*, for *acyclic* nondeterministic **HSMs**,

(d) *all* equivalence relations in the Linear-time/Branching-time Hierarchy are **PSPACE**-*hard* for *acyclic* **CHSMs**, and

(e) for all fixed *acyclic* **HSMs** M_0, testing TRACE-EQUIVALENCE to M_0 is polynomial time solvable; but in contrast for all fixed *acyclic* **CHSMs** M_0, testing TRACE-EQUIVALENCE to M_0 is **PSPACE**-*hard*

(These appear to be the first *hardness* results, for acyclic **HSMs** and for acyclic **CHSMs**. All of these results follow directly from *known* results for regular set descriptors from [SM73, Hu73, HRS76, HR77, HR78].);

6. for all integers $k \geq 1$, when restricted to hierarchical specifications of depth $\leq k$, all of the problems of Item 2, for many of these models, become **DSPACE**(n^k)-*hard* and/or *-complete*

(• The **EXSPACE**-*hardness* results, for hierarchical specifications in [RH93, AY98] and those in this paper, *require* problems instances with *unboundedly* large depth. The depths of hierarchical specifications, that occur in practice, are usually bounded by fixed constants depending upon the application area. Consequently, the potential practical implications of these **EXSPACE**-*hard* results are questionable. In contrast, this last result provides the first complexity results, for hierarchically-specified problem instances of any *fixed* depth of hierarchical specification. We can prove similar indexed complexity results, for problems for *narrow* difference equations with $\leq k$ independent variables, and for **CHSMs** with fixed bounds on the numbers of applications of *hierarchy* and *parallel composition* in their specifications. All of these indexed families of complexity results are *new*.); and

7. paraphrasing [BPT91], a family of general results showing *computationally- intractable dependence on initial conditions*, for all the models $\mathcal{M}$ above (except for the **HSMs** and **CHSMs**) ranging from **DSPACE**(n)-*hard* to *undecidability* inclusive

(Moreover, *both* the *same* proof implies *simultaneously* all of these *computationally-intractable dependence* results, and we get indexed families of complexity results identical to those of the previous item.).

All of the above results are, for problem instances *without* the *hiding* abstraction. We can also extend our results in [SH+96] to show several additional general complexity-theoretical implications, for instances *with* the *hiding* abstraction, for most of the models $\mathcal{M}$ above. These results include results exactly analogous to those of Items 1 and 6 above.

2.1 Nuances of various models considered and additional implications

Several issues involving details of model specifications turn out to play important roles in the development of the results outlined here. First, the notions of *finiteness* and *infiniteness* occur in essentially two different ways in the models considered here as follows:

1. *Finiteness/infiniteness* can refer to the cardinality of the number of equations, automata, cells, or states in a discrete dynamical system, system of communicating automata, or nondeterministic automaton, even

when the individual variables of the equations can only take on values from a fixed finite collection of finite sets or the state sets of the individual automata are contained within a fixed finite collection of finite sets of states. For example, a 1-**FCA** has only a finite number of cells; but the classical cellular automata (**CAs** of [Wo86, Gu89] have countably infinite numbers of cells.
2. The domains of the algebraic structures in which computations are carried out or the sets of states of the individual automata can be either finite or infinite.

Henceforth, we use the term **FDDS** to mean *both* a finite number of equations, state variables, cells, etc., and a finite set of finite algebraic structures or a finite set of finite sets of possible states. [9] Second, there are several variant models of transitions systems that occur in the references cited above. For many references on process algebras, transition systems have *no* accepting states (or equivalently *all* states can be viewed as accepting.) Often these systems synchronize on ACTION symbols. We consider synchronization using both explicit channels and synchronization using ACTION symbols; and we consider succinct specifications of both types of distributed systems. For the latter type, we consider succinct specifications with natural mechanisms, for succinctly specifying ACTIONs. In contrast, the references [AY98, AKY99, Al00] discuss nondeterministic finite automata; and they implement both word acceptance and parallel composition using *explicit final* or *accepting* states. For these succinctly-specified finite automata, we present the following three types of results:
3. Simple proofs showing how STATE-REACHABILITY problems for such systems are still reducible, using reductions by *local-replacement* to the various equivalence relations and simulation preorders of the Linear-Time/Branching-Time hierarchy. Again, these reductions are ultra-efficient in terms of both sequential and parallel complexity.
4. Proofs of **hardness** for the STATE-REACHABILITY problem for such very simple such systems using ideas from [Hu73] on the use of intersection.
5. Direct applications of efficient reductions by *local-replacement* of the EMPTINESS and EMPTINESS-OF-INTERSECTION into various problems for these machines.

2.2 Sensitivity to initial conditions

Additional important properties of the concepts, techniques, results, and proofs presented here include their generality and uniformity, e.g. they apply directly *both* to finite and infinite discrete dynamical systems **DDSs** and to the various kinds of communicating automata considered here. One very general result is that

- All the models **M** of **DDSs** or *communicating automata except* for the **HSMs** and the **CHSMs** of [AY98, AKY99] exhibit *computationally- intractable sensitive dependence on initial conditions*, whenever **1.** their STATE-EXECUTABILITY Problem is *computationally- intractable*, and **2.** they are efficiently-closed under certain simple *local-replacements*.

In the limit case of a countably infinite number of cells, this *computationally- intractable sensitive dependence on initial conditions* becomes **undecidable sensitive dependence**, without requiring such unnatural properties of problem instances as unbounded local memories, unbounded fan-in, fully-centralized control, or complicated inter-connections (as occur in the infinite limits of the systems of automata in [BPT91, Ra92]). Moreover when sequential circuits or systems of communicating automata are hierarchically-specified as in [RH93, Ga82, BOW83, LW88, LW92, RH93], we get **PSPACE**-, **DEXPTIME**-, **NDEXPTIME**-, and **EXSPACE**-*hard sensitive dependence on initial conditions*, depending upon the actual kind of specification and whether individual automata are cyclic or are acyclic.

[9] For us a *finite algebraic structure* consists of a nonempty finite domain D, together with a finite set of (possibly partial) finite-arity functions or relations on D.

Finally, the remainder of this extended abstract consists of the following: Section 3 selected preliminaries and definitions; Section 4 selected proof sketches; and the Appendix in which for the convenience of the reader, we recall the basic definitions from the literature of *transition systems*, *parallel composition*, the *hiding* abstraction, and several basic equivalence relations and simulation pre-orders including *bisimulation-equivalence* and *weak-bisimulation equivalence*.

3 Preliminaries

The proofs of our *hardness* results involve only structurally very simple instances of discrete dynamical systems or of networks of inter-connected communicating finite automata. For this reason and to simplify and shorten the statements of the definitions used here, our definitions are not as general as they could be. Thus for example, we define only *k,l-discrete* and *k,l-finite discrete dynamical systems*, rather than arbitrary *discrete* and finite discrete dynamical systems. Also we restrict our definition of difference equations to difference equations with constant coefficients *only*. However, no real loss of generality occurs, since the restricted instances of *discrete dynamical systems* and *nonlinear difference equations* defined here are sufficiently general to include *all* instances actually needed in our *hardness* proofs and they obviously satisfy natural formulations of the corresponding more general definitions.

Throughout this paper, **F** denotes an algebraic structure consisting of a nonempty domain D, together with a finite set of finite-arity operators on this domain. We say that an algebraic structure is *non-degenerate* if its domain has at least 2 elements. Throughout this paper all algebraic structures are assumed to be *non-degenerate*. Most of the algebraic structures considered have 2 binary operators called *addition*, denoted $+$, and *multiplication* , denoted $\cdot$. We denote the *identity* of the operation $+$ by 0. An algebraic structure with a multiplication operator $\cdot$ is said to be *unitary* if it has a *multiplicative identity*, denoted here by 1. Definitions of the kinds of algebraic structures considered here, namely *rings*, *semirings*, and *lattices*, can be found in [MB67, Ei74]. Unlike [MB67, Ei74], however, we do *not* assume that all rings or semirings are *unitary*. For us, a *formula* on an algebraic structure is a finite string built-up from operators symbols of the structure[10], variable symbols, and parentheses. Because of computational complexity considerations, we assume that all such variable symbols are of the forms x_r, y_r, etc., where the subscripts r are binary numerals without leading zeros.

We denote the sets of languages accepted by nondeterministic (deterministic) linearly space-bounded, polynomially space-bounded, exponentially time-bounded, and exponentially space-bounded multiple-tape Turing machines (abbreviated **Tms**) by **NSPACE(n) DSPACE(n)**, **PSPACE**, **NEXPTIME**, **DEXPTIME**, and **EXSPACE**, respectively. We denote the set of languages that are the complements of languages accepted by nondeterministic multiple-tape **Tms** by **coNDEXPTIME**. Finally, we abbreviate (*deterministic*) *finite automata* by **(d)fa**, *linearly-bounded automaton* by **lba**, *context-free grammar* by **cfg**, and *pushdown automaton* by **pda**.

3.1 Particular models considered

The following models **M** are considered here:

1. discrete and finite discrete dynamical systems (**DDSs** and **FDDSs**), i.e. systems consisting of several and systems consisting of a single nonlinear difference equations, presented with designated initial values as needed,

2. both finite and infinite cellular automata (**FCA** and **CA**) [Wo86, CY88, CPY89, Gu89],

[10]We assume the operators and operator symbols of the structure are in one-to-one correspondence.

3. finite graph automata (**FGA**) [Ma98, NR98],

4. communicating sequential machines, hierarchical sequential machines, and communicating hierarchical sequential machines (**CSMs, HSMs,** and **CHSMs**) as defined in [Ho84, VW86, GM92, Ra92, SH+96, AY98, AKY99], etc.

Due to lack of space, here, we only formally define the following models:

(a) structurally restricted versions of the **DDSs** and **FDDSs,** that are strict generalization of CAs and graph automata

(b) the classes of *narrow* and of *wide* nonlinear difference equations,

(c) the parallel compositions of transitions of transitions systems, denoted **CSMs,** of [Ho84, Ra92, SH+96], and

(d) the **HSMs** and **CHSMs** of [AY98, AKY99].

3.1.1 DDSs, FDDSs, and difference equations

We define formally the *k,l-restricted-* and the *k,l-finite restricted- dynamical systems*, where k and l are positive integers such that, for each variable $x_j()$,

1. the value $v_j(t+1)$ of x_j at time $t+1 \geq 1$ is a function of the values of $\leq k$ different variables x_i at time t, and

2. the algebraic formulas on **F** giving the equations to compute the values $v_j(t+1)$ have no more than l occurrences of operator symbols of **F**.

Throughout for reasons of simplicity, we assume that the parameter(s) of discrete dynamical systems take values from **N**, the set of *natural numbers*. Also for reasons of simplicity, we only define one parameter difference equations.

Definition 3.1 *Let* $n \geq 1$ *and* **F** *be an algebraic structure. Let* $x_1(), \ldots, x_n()$ *be distinct (parameterized) variable symbols.*

*1. A k,l-***(restricted) discrete dynamical system** *on* **F** *(denoted k,l-***DDS(F)***)* $\mathcal{F}$ *consists of a finite sequence of equations* $(eq_1, \ldots, eq_n)$*, together with an n-tuple* $(c_1, \ldots, c_n)$ *of elements of D, where each equation* eq_j *is of the form*

$$x_j(t+1) := f_j(x_{j_1}(t), \ldots, x_{j_{k'}}(t)),$$

such that $1 \leq k' \leq k$, $1 \leq j_1 \leq \ldots \leq j_{k'} \leq n$*, and* f_j *is a formula on* **F** *with* $\leq l$ *occurrences of operator symbols of* **F**. *When the structure* **F** *is finite, we say that* $\mathcal{F}$ *is a k,l-(restricted) finite discrete dynamical system on* **F***) (denoted k,l-***FDDS(F)***).*

2. The **sequence specified by** $\mathcal{F}$*, denoted* $\sigma(\mathcal{F})$*, is the countably infinite sequence of n-tuples of elements of D* $(\sigma(0), \ldots, \sigma(s), \ldots)$*, where*

$$\sigma(0) = (c_1, \ldots, c_n), \text{ and for } s \geq 0 \text{ and } 1 \leq j \leq n,\ \sigma(s+1)_j = f_j(\sigma(s)_{j_1}, \ldots, \sigma(s)_{j_{k'}}).$$

Here $\sigma(s+1)_j$ *denotes the* j^{th} *element of* $\sigma(s+1)$.

Definition 3.2 *Let $k \geq 1$. Here, $\{x_i \mid i \geq 0\}$ is a set of distinct variable symbols.* **1.** *A* **narrow** *(***nonlinear***)* **difference equation** $\mathcal{F}$ **with initial values** *(on* **F***) consists of a single equation of the form $x_n := f(x_{n-1}, \ldots, x_{n-k})$ defined in terms of the indicated variable symbols and operators of* **F***, together with a k-tuple $v_0, \ldots, v_{k-1}$ of values in D such that*

i the subscript offsets $-1, \ldots, -k$ are written in unary, and

ii each of the values v_i where $0 \leq i \leq k-1$ is separately specified.

2. *Let $m \geq 1$ such that $k < 2^m$. A* **wide** *(***nonlinear***)* **difference equation** $\mathcal{F}$ **with initial values** *(on* **F***) consists of a single equation of the form $x_n := f(x_{n-2^m+k}, \ldots, x_{n-2^m-k})$ defined in terms of the indicated variable symbols and operators of* **F***, together with a $2^m + k$-tuple $v_{-k}, \ldots, v_{2^m-1}$ of values in D such that*

iii the subscripts offsets $-2^m + k, \ldots, -2^m - k$ are written in binary, and

iv each of the values v_i where $-k \leq i \leq m-1$ are separately specified and the remaining values of the v_i where $m \leq i \leq 2^m - 1$ are specified by statements of the form "for $m \leq i \leq 2^m - 1$ $v_i = b$", where b is an element of D. (Again, the integer $2^m - 1$ is written in binary without leading zeros.) [11]

We define the *sequence $\sigma(\mathcal{F})$ defined by* a difference equation with initial values $\mathcal{F}$ on an algebraic structure **F** in the obvious ways, directly analogous to the corresponding definitions given in Definition 3.1.

A number of different models of communicating finite automata studied in the literature, restricted to linearly inter-connected automata or to automata inter-connected in simple bounded grid patterns, can be viewed directly as *k,l*-**FDDSs(F)**, for appropriately chosen algebraic structures **F**. For example, consider 1- or 2- dimensional **FCAs** can be so viewed, since all variable values at time $(t+1)$ depend only on variable values at time t. As a 2nd example, consider the variant of **FCAs** in which the states are updated in a specified sequential order. Such systems can also be modeled directly by sets of equations as above. This can be used to see that **FDDSs** model finite systems of communicating finite automata with both *synchronous* and *sequential* state update. Moreover by allowing infinite algebraic structures **F**, our definitions apply to arbitrary *discrete dynamical systems* in the sense of the mathematical literature on dynamical systems [Ro99]. Finally, the infinite systems of inter-connected automata considered here include both 1- and 2-dimensional **CA** as defined and studied in [Wo86, CPY89, CY88, Gu89, Du94, Su95, Wo86].

3.1.2 HSMs and CHSMs

In order to define the classes of **CSMs, HSMs**, and **CHSMs** of [Ho84, Ra92, SH+96, AY98, AKY99], the reader should first recall the basic definitions of *transition systems*, of the *parallel composition* of transitions systems, and of various *simulation equivalences* and *simulation pre-orders* for transitions systems (also see the above references plus [vG90]). For the convenience of the reader, selections of these last definitions (including the definitions of *transition systems* and the *parallel composition* of such systems) appear in the Appendix. The definitions below are essentially from Alur, Kannan and Yannakakis [AKY99].

Definition 3.3 *Letting $\|$ denote the parallel composition operator of [Ho84]. A* **CSM** *is a finite nonempty sequence of transitions systems $(M_1, \ldots, M_n)$ denoting the transition system $((\ldots(M_1 \| M_2)\ldots) \| M_n)$.*

Definition 3.4 *Formally, a* **communicating hierarchical state machine** *(***CHSM***) is one of the following three forms:*

[11] The actual wording of this definition was chosen for reasons of simplicity. More generally, we say a difference equation with initial values $\mathcal{E}$ is **wide**, if there exists integer $c \geq 1$, such that E can be obtained, from a **wide** difference equation with initial values satisfying the definition above by replacing each variable x_i and each initial value a_j by c distinct variables $x_i^1, \ldots, x_i^c$ and initial values $a_j^1, \ldots, a_j^c$, respectively.

1. **Base Case:** *A FSM* $\mathcal{T} = (S, A, D, s, f)$[12] *is a* **CHSM**

2. **Concurrency:** *If* $\mathcal{T}_1, \mathcal{T}_2, \ldots \mathcal{T}_k$ *are* **CHSMs** *then* $\mathcal{T}_1 \| \mathcal{T}_2 \cdots \| \mathcal{T}_k$ *is also a* **CHSM** *where* $\|$ *is a parallel composition operator as defined in the Appendix.*

3. **Hierarchy:** *If* $\mathcal{M}$ *i a set of* **CHSMs** *and* $\mathcal{T} = (S, A, D, s, f)$ *is a* **FSM** *with* S *as the state set and* μ *is a labeling function* $\mu : S \to \mathcal{M}$ *that associates with each state in* S *a machine in* $\mathcal{M}$ *with appropriate mapping of incoming and outgoing edges, then* $(\mathcal{T}, \mathcal{M}, \mu)$ *is also a* **CHMS**.

A **hierarchical state machine** *(***HSM***) is a* **CHSM** *having* no *occurrences of the parallel composition operator.*

The semantics of a **CHSM** $\mathcal{M}$ are defined by mapping it to a finite sequential machine (**FSM**) $[[\mathcal{M}]]$ as follows:

1. **Base Case:** If $\mathcal{M}$ is an FSM then $[[\mathcal{M}]]$ equals $\mathcal{M}$.

2. **Concurrency:** If $\mathcal{M}$ is a product of expressions $\mathcal{M}_1 \parallel \mathcal{M}_2 \cdots \parallel \mathcal{M}_k$ then $[[\mathcal{M}]]$ is automata defined by the parallel composition rule above.

3. **Hierarchy:** If $\mathcal{M} = (\mathcal{T}, \mathcal{S}, \mu)$ is a **CHSM** with the top level **FSM** being $\mathcal{T} = (S, A, D, s, f)$, then
   (a) A state of $[[\mathcal{M}]]$ is of the form (q, w), where $q \in S$ and w is the state of **FSM** $[[\mu(q)]]$ associated with q.
   (b) A symbol σ belongs to the symbol set of $[[\mathcal{M}]]$, iff either σ is in the actions set of $\mathcal{T}$ or it is in the action set of one $[[\mu(q)]],\ q \in S$.
   (c) The initial of $[[\mathcal{M}]]$ is the initial state of $[[\mu(s)]]$
   (d) The final of $[[\mathcal{M}]]$ is the final state of $[[\mu(f)]]$
   (e) $[[\mathcal{M}]]$ has two types of transitions:
      - For a transition (q, σ, q') of the top level **FSM** $\mathcal{T}$, there is a transition from the final state of $[[\mu(q)]]$ to the initial state of $[[\mu(q')]]$
      - For $q \in S$ if (w, σ, w') is a transition of $[[\mu(q)]]$, then $((q, w)\sigma, (q, w'))$ is a transition of $[[\mathcal{M}]]$

4 Selected proof sketches

Most of the *new* results of this paper are for **DDSs**, networks of communicating automata, systems of nonlinear difference equations, etc., whose specifications can be much smaller than the specifications of the object as usually considered in the literature. The need for, and the consequent use of, such *succinctly*-specified **DDSs**, networks of communicating automata, etc., occurs naturally in the design, analysis, and verification of large to very large practical problems. This is because when humanly-designed, such large to very large objects are usually defined/specified in terms of regular combinations of smaller objects. Two well known kinds of such *succinct specifications* are the *hierarchical* and the *dynamic/periodic* specifications. Several different variants of *hierarchical* and *dynamic/periodic* specifications of graphs, circuits and automata have been considered over the last twenty years, e.g. [Le86, LW88, LW92, Ga82, GW83, BOW83, MH+97, RH93,

[12]Recall that S denotes the set of states D is the transition relation, A is the set of actions, s is the initial state and f is the final state.

Or84, KO91, CM93, MH+98, AY98, AKY99, Al00] and the many references therein. These specifications have been studied in the context of circuit design, analysis, and specification and in the context of the verification of software systems and structured programs. The formalism of Alur et. al. [AY98, AKY99, Al00] is one possible way to succinctly specify nondeterministic finite automata and can be viewed as a direct extension of the hierarchical specification of graphs proposed by Lenguaer et. al. [Le86, LW88, LW92]. Also, the result obtained by Lengauer et. al. [LW92] can be used to obtain the result in [AY98] that STATE REACHABILITY for **HSMs** is **P**-complete.

4.1 Overall techniques and their properties

As stated in the introduction, our proofs rely on the following four general ideas and their properties.

1. The proofs of **DSPACE**(n)- and **NDSPACE**(n)-*hardness* results, for STATE- REACHABILITY problems, intuitively only require that a model $\mathcal{M}$ of communicating automata be able to specify a system of N linearly-connected copies of one particular *fixed* deterministic and of one particular *fixed* nondeterministic finite automaton m by a specification of size $O(N)$. This suffices because in [RH93] two of the authors have already shown that the STATE-REACHABILITY problem, for such linearly-connected systems of copies of m is **DSPACE**(n)- and **NDSPACE**(n)-*complete*. Two additional relevant details of the construction in [RH93] are:

   i For the STATE-REACHABILITY problem to be **PSPACE**-*hard*, for the model $\mathcal{M}$, all that is necessary is that there exist a *fixed* rational number $r > 0$ such that the specification be of size $O(N^r)$.

   ii The state s of the system of linearly-connected copies of m in [RH93], determining whose reachability is **DSPACE**(n)- or **NDSPACE**(n)-*hard* is a state of the *right-most* copy of m in the linearly-connected copies of m. This means that we can use very *simple* instances of **Local**-STATE-REACHABILITY problems as the sources of the efficient reductions used to prove our *hardness* results.

2. The proofs of our **EXSPACE**-*hardness* results, for STATE-EXECUTABILITY problems, intuitively only require that a variant *hierarchical* specification $\mathcal{S}$ be able to specify a system of 2^N linearly-connected copies of one particular *fixed* deterministic finite automaton m by a specification of size $O(N)$. This suffices because in [RH93] two of the authors have already shown that the STATE-EXECUTABILITY problem, for such linearly-connected systems of copies of the deterministic automaton m is **EXSPACE**- *complete*. The *exact* analogues of **i** and **ii** of the previous item hold here as well.

3. The proofs of our **coNDEXPTIME**-*hardness* results for STATE-EXECUTABILITY problems, intuitively only require that a variant *periodic* specification $\mathcal{S}$ be able to specify a system of $2^N \times 2^M$ copies of a few basic *acyclic* finite automata connected together in a 2-dimensional grid-pattern by a specification of size $O(N + M)$. This suffices because:
   iii.In [MH+98] we show that the problem EXACTLY1-EX3MONOTONESAT is already **NDEXPTIME**-*complete*, when instances are so specified.
   iv.In [SH+96, Sh97], we showed how to reduce the problem EXACTLY1-EX3MONOTONESAT to the STATE-REACHABILITY problem for *acyclic* communicating automata by means of a reduction by *local-replacement*.

4. Most of the *hardness* proofs in this paper are by reductions by *local- replacement.* We can show *both* that these reductions can be carried out in deterministic $O(n \cdot logn)$ time on deterministic multi-tape **Tms** and that these reductions can be extended to efficient reductions, when instances are specified-succinctly as discussed above [MH+98, HSM00, HMS01].

Because of the simplicity of the instances used to prove these *hardness* results, it seems intuitively clear that analogues of the **EXSPACE**-and **coNDEXPTIME**-*hardness* results discussed in **1** and **2 hold** generally, when systems of communicating automata are specified by any of the variant succinct specifications referenced above. In particular, these *hardness* results hold, for hierarchically-specified **FCA** and **FGA**. They also hold for hierarchically- and dynamically-/periodically-specified **CSMs**, provided we allow appropriately defined succinct specifications of ACTION symbols, e.g. distinct expansions of a module have distinct copies of those ACTION symbols specified to be *local* in the module.

In the remainder of the paper, we provide select proof sketches that illustrate the above ideas.

4.2 A general reduction of state-executability to simulation equivalence relations and preorders

We recall the following definition from [HS76, SH+96]:

Definition 4.1 *Let ρ, σ, τ be binary relations on a nonempty set D. We sat that σ is* **between** *ρ and τ if, for all $x, y \in D$, $x\rho y$ implies $x\sigma y$ and $x\sigma y$ implies $x\tau y$.*

By direct inspection of the systems S_1,S_2 in Figure 1, if the state labeled A is *not* reachable from the initial state of S_1, then S_1 and S_2 are COMPUTATIONALLY-IDENTICAL, and otherwise, *both* the set of traces of S_1 is *not* a subset of the set of traces of S_2 and the set of traces of S_2 is *not* a subset of the set of traces of S_1. Consequently, for all equivalence relations or simulation pre-orders σ *between* COMPUTATIONAL-IDENTITY and TRACE- CONTAINMENT, $S_1 \ \sigma \ S_2$ if and only if the state A is *not* reachable from the initial state of S_1. Recalling the definitions of COMPUTATIONAL-IDENTITY and BISIMULATION-EQUIVALENCE given above and in the Appendix, respectively, every relation *between bisimulation-equivalence* and *trace-equivalence/trace-containment* is also *between* COMPUTATIONAL-IDENTITY and *trace- equivalence/trace-containment.* Consequently since every equivalence relation or simulation pre-order of the Linear-time/Branching-time hierarchy of [vG90] is *between bisimulation-equivalence* and *trace-equivalence/trace-containment,* the argument illustrated in Figure 1 implies that various STATE-REACHABILITY problems are efficiently reducible to *each* of the equivalence relations and simulation pre-orders of the Linear-time/ Branching-time hierarchy using intuitively reductions by *very simple local replacement* [13].

• In particular this *very* simple but general meta-argument, together with the above discussion on the "four important properties of our hardness proofs", yield directly the **PSPACE**- and **EXSPACE**-*hardness* of determining all equivalence relations and simulation pre-orders of the Linear-time/Branching-time hierarchy, for various of the models $\mathcal{M}$ above.

4.3 Direct implications to DDSs

We restrict our discussion to systems of simultaneous difference equations with constant coefficients on *any* algebraic structure with *monotone-logic expressibility.* First, we show that *all* unitary rings, *all* finite unitary semirings, and *all* lattices have *monotone-logic expressibility.* This gives some idea of the wide generality of our results.

Case 1:Let **F** be a unitary ring; let D be the domain of **F**; and let f_1, f_2, f_3 be the functions from $D \times D$

[13]General discussion along these lines already appears in [HS76, Hu82].

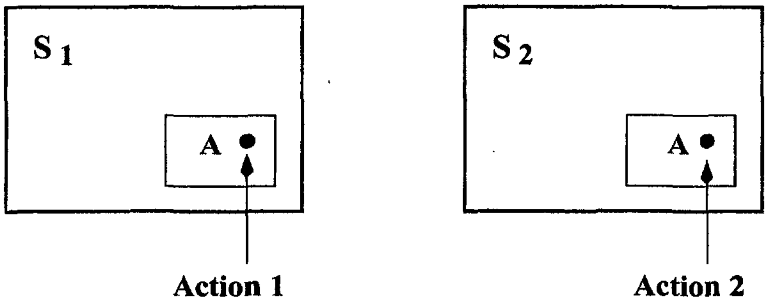

Figure 1: Figure illustrating the reduction of state reachability problem to all relations between computational identity and trace equivalence and trace containment. S_1 and S_2 are identical except for the inner boxes. Note that if the state A is not reachable from the initial state of S_1 then S_1 and S_2 are computationally-identical; otherwise the sets of traces of S_1 are not a subset of traces of S_2 and conversely. We assume that actions 1 and 2 do not occur anywhere else.

to D or D to D defined by– $f_1(x,y) = x + y + -(x \cdot y)$, $f_2(x,y) = x \cdot y$, and $f_3(x) = 1 - x$. Let g_1, g_2, g_3 be the restrictions of f_1, f_2, f_3, respectively, to $\{0,1\} \times \{0,1\}$ or to $\{0,1\}$. Then, the algebraic structure $\mathcal{A} = (\{0,1\}, g_1, g_2, g_3)$ is isomorphic to the 2-element Boolean algebra; and hence a fortiori, **F** has *monotone-logic expressibility.*

Case 2:Let **F** be a lattice. Since we assume all algebraic structures are non-degenerate, there exist a, b in **F** such that $a \neq b$. Let $\alpha = a \wedge b$ and $\beta = a \vee b$. Under the operations $\vee$ and $\wedge$ of **F**, α, β are isomorphic to 0,1 under *or* and *and.*

Case 3:Let **F** be a finite unitary semiring. If 1 is invertible under $+$, then **F** is actually a *unitary* ring and **Case 1** applies. Otherwise, there exists $n \geq m \geq 1$ such that $m * 1 = n * 1 \neq 0$. (Here, $m * 1$, $n * 1$ equal the sum in **F** of m and n 1s, respectively.) In which case, it can be shown that there exists $m' \geq 1$ such that $m' * 1 = m' * 1 \cdot m' * 1 = m' * 1 + m' * 1 \neq 0$. But this implies that **F** has a non-degenerate *finite* sub-lattice, the operations of which are definable in terms of the operations of **F**. Consequently, **Case 2** applies.
Additionally, the proof of **Case 1** shows that *all* fixed-precision discretizations of the integers, rationals, reals, and complex numbers also have *monotone-logic expressibility.*

Next, we show how to reduce the *State-Reachability* problem for linearly inter- connected systems of copies of a single **dfa** to the **IVP** for a system of simultaneous non-linear difference equations over **F** with *monotone-logic expressibility.* Let a,b f_1,f_2 be as in the definition of *monotone- logic expressibility.* Henceforth we view a, b, f_1, f_2 as 0, 1, *or*, *and*, respectively. Let the system $\mathcal{M}$ consist of $n \geq 1$ copies of the p-state **dfa** m_0. Each state of each cell c_i $(1 \leq i \leq n)$ of the system $\mathcal{M}$ at time $t \geq 0$ is represented by a $2k$-tuple of distinct Boolean-valued variables $x_i^1(t)$, $y_i^1(t)$, $\ldots$, $x_i^k(t)$, $y_i^k(t)$. Here, the k-tuple $(x_i^1(t), x_i^2(t), \ldots, x_i^k(t))$ is a binary code of the state of cell c_i at time $t \geq 0$ and for each $1 \leq i \leq n$, for each $1 \leq j \leq k$, and each $t \geq 0$, $x_i^j(t)$ *or* $y_i^j(t) = b$ and $x_i^j(t) and y_i^j(t) = a$. By this *twinning* of Boolean-valued variables, we can eliminate the need for the Boolean operator *not* in the Boolean equations defining the values of the variables $x_i^1(t+1), y_i^1(t+1), \ldots, x_i^k(t+1), y_i^k(t+i)$ in terms of the values of that subset of the variables $x_{i-1}^1(t), y_{i-1}^1(t), \ldots, x_{i+1}^k(t), y_{i+1}^k$ needed to compute their values. Given this, the remainder of the proof uses standard arguments.

This reduction, together with the results in [RH93], immediately implies the **PSPACE**-*hardness* of the IVP, for such systems of difference equations on *any* algebraic structure with *monotone-logic expressibility,* when the system is specified *non*-succinctly, i.e. as usually assumed in the literature. Because all the cells in the sources of this reduction are the *same* and are linearly inter- connected, it is not hard to see that the systems of difference equations, that are the targets of this reduction, can be specified hierarchically so that a system of $O(2^N)$ such equations can be specified by a hierarchical specification of size $O(N)$. Given this the results in [RH93] also imply that the **IVP** is **EXSPACE**-*hard*, when systems of difference equations are hierarchically-specified.

4.4 Selected applications of known hardness results for regular set descriptors to acyclic HSMs and CHSMs

PSPACE-*hardness* proofs by the direct encoding of the computations of **lbas** M on fixed inputs x into strings consisting of concatenations of appropriate *instantaneous-descriptions* (**ids**) for M on the the fixed inputs x are well-known and go back to [SM73, Hu73]. Let $N = |x|$. The properties of such encodings include: (1)Each such **id** is of length N. (2)The **initial-id** of M on x, denoted $\mathbf{Init}_0(x)$, is the length N string $(q_0, x_1)\cdots x_N$, where q_0 is the *start-state* of M and $x = x_1 \cdots x_N$. (3)Without-loss-of-generality, we may assume that there exists a positive integer c such that if M accepts, then after $\leq 2^{cN}$ steps M's **id** is $(q_f, 0)\cdots 0$, denoted by **Fin**(x), where q_f is M's unique *accepting-state*. Given these properties, let M be a *fixed* deterministic **lba**. Let x of length $N \geq 1$ be an input to M. Then two **cfgs** G_1,G_2 can be constructed deterministically in polynomial time in N such that:

$$L(G_1) = \mathbf{Init}_0(x) \cdot \{w^{rev}w \mid w \text{ is an ID of } M \text{ of length } N\}^{2^{cN}} \text{ and}$$

$$L(G_2) = \{wu \mid wu^{rev} \text{ are length } N \text{ IDs of } M \text{ and } u^{rev} \text{ results from } w \text{ by a move of } M\}^{2^{cN}} \cdot \mathbf{Fin}(x).$$

By inspection, $x \in L(G_1) \cap L(G_2)$ if and only if the **lba** M accepts x. Thus, the EMPTINESS-OF-INTERSECTION problem for pairs of such **cfgs** is **PSPACE**- *hard.* Noting that the languages $L(G_1)$,$L(G_2)$ are *both* finite and that the lengths of all the substrings **Init**(x), w, u, and **Fin**(x) equal N, it is not hard to see that the grammars G_1,G_2 can be translated into equivalent deterministic **PDA** with bounded pushdown stores and into equivalent *acyclic* **HSMs**. (Recall that the two languages $L(G_1)$,$L(G_2)$ are finite.) These two *acyclic* **HSMs** are actually *acyclic incompletely-specified* **deterministic HSMs**. Consequently by adding single *trap-states*, one for each **HSM**, the resulting **HSMs** are **deterministic HSMs** that accept *finite* languages.

Immediate direct corollaries of the above argument and results from the literature on the complexity of problems, for regular set descriptors in [SM73, Hu73, HRS76, HR77], for *acyclic* **HSMs**, *acyclic* **CHSMs**, and for **CHSMs** specified *without* use of the *hierarchy* constructor, are given in the following theorem. **No** results, for any of these very simple restricted **HSMs** or **CHSMs**, are claimed in [AY98, AKY99].

Theorem 4.1 *1. The* EMPTINESS-OF-INTERSECTION *problems are* **PSPACE**-*hard, for* acyclic **HSMs**, *for* acyclic *incompletely-specified* **deterministic HSMs**, *and* **deterministic HSMs** *that accept* finite *languages.*

2. The EMPTINESS *and* STATE-REACHABILITY *problems are* **PSPACE**-*hard for* **CHSMs**, *that are the parallel composition of two* acyclic **HSMs**, *are the parallel composition of two* acyclic *incompletely-specified* **deterministic HSMs**, *or are the parallel composition of two* **deterministic HSMs** *that accept* finite *languages.*

3. The CONTAINMENT *problem is* **PSPACE**-*hard, for pairs of* **deterministic HSMs**, *even when one of the* **HSMs** *is known to accept a finite set and the other a co-finite set.*

4. For all regular sets R_0, the problems of determining if the language accepted by a deterministic **HSM** *equals R_0 or is contained in R_0 are* **PSPACE**- *hard.*

5. The EQUIVALENCE *and* CONTAINMENT *problems are* **coNDEXPTIME**-*hard, for* acyclic **HSMs**.

6. *The* EMPTINESS *and* STATE-REACHABILITY *problems are* **PSPACE**-*hard, for* **CHSMs**, *that are the parallel composition of finite sequences of* **dfa**. *(Thus, these problems are already* **PSPACE**-*hard, for* **CHSMs** *specified without any use of the hierarchy constructor.)*

Proof: The claims of Items 1 and 2 follow directly from the argument above and the fact that, as defined by [AY98, AKY99], the language accepted by the parallel composition of two **HSMs** with *identical* tape alphabets equals the *intersection* of the languages accepted by the two **HSMs**. The claim of Item 3 follows directly the claim of Item 1 since

$L_1 \cap L_2 = \emptyset$ if and only if $L_1 \subset \overline{L_2}$ and the two **HSMs** of Item 1 are deterministic.

The claim of Item 4 is implied directly by the following argument from [Hu73, HR77]: For simplicity let c be a letter *not* appearing in any word in the language R_0. Let M be any **HSM**. Let $L_M = L(M) \cdot \{c\} \cup R_0$. Then the following are equivalent: L_M *equals* R_0; L_M is *contained in* R_0; and $L(M) = \emptyset$. Finally, a **HSM** recognizing the language L_M can be constructed from the **HSM** M in deterministic polynomial time.

The claim of Item 5 follows immediately from the **coNEXPTIME**-*hardness* of the EQUIVALENCE and CONTAINMENT problems for $(\cup, \cdot, ^2)$-regular expressions and for context-free grammars generating finite languages [SM73, HRS76]. The claim of Item 6 follows directly from the results showed in [Hu73], that the problem of determining, given a finite sequence of deterministic finite automata $(M_1, \ldots, M_n)$, if $L(M_1) \cap \ldots \cap L(M_n) = \emptyset$ is **PSPACE**-*complete.* (Recall again, that as defined by [AY98, AKY99], the language accepted by the parallel composition of a sequence of **CHSMs** all with the *same* alphabet equals the *intersection* of all of the languages accepted by the **CHSMs**.) ∎

Acknowledgements: This research was also funded by the LDRD-DR project *Foundations of Simulation Science* and the LDRD-ER project *Extremal Optimization* at the Los Alamos National Laboratory. We thank Sandeep Shukla (Intel Corp.) and Professor Richard Stearns (SUNY, Albany) for several discussions on topics related to this paper.

References

[AKY99] R. Alur, S. Kannan, M. Yannakakis, Communicating Hierarchical State Machines. *International Colloquium on Automata programming and Languages (ICALP)* 1999. 169-178.

[AY98] R. Alur, M. Yannakakis. Model Checking of Hierarchical State Machines. *ACM Symposium on Foundations of Software Engineering (SIGSOFT FSE)*. 1998, pp. 175-188.

[Al00] R. Alur. Exploiting Hierarchical Structure for Efficient Formal Verification. *CONCUR* 2000, pp. 66-68.

[ARV94] S. Arora, Y. Rabani and U. Vazirani. Simulating quadratic dynamical systems is **PSPACE**-complete," *Proc 26th. Annual ACM Symposium on the Theory of Computing (STOC)*, pp. 459-467, Montreal, Canada, May 1994.

[BHM+00] C. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz and R. E. Stearns. Dichotomy Results for Sequential Dynamical Systems. Submitted for publication, Dec. 2000.

[BMR00] C. Barrett, H. Mortveit and C. Reidys. Elements of a theory of computer simulation III: equivalence of SDS. to appear in *Applied Mathematics and Computation*, 2000.

[BOW83] J.L. Bentley, T. Ottmann and P. Widmayer, "The Complexity of Manipulating Hierarchically Defined set of Rectangles," *Advances in Computing Research, ed. F.P. Preparata* 1, pp. 127-158, 1983.

[BPT91] S. Buss, C. Papadimitriou and J. Tsitsiklis. On the predictability of coupled automata: An allegory about Chaos. *Complex Systems*, 1(5), pp. 525-539, 1991. Preliminary version appeared in *Proc. 31st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, Oct 1990.

[CM93] E. Cohen and N. Megiddo. "Strongly polynomial-time and NC algorithms for detecting cycles in dynamic graphs," *Journal of the ACM (JACM)* 40(4), pp. 791-830, 1993.

[CPY89] K. Cullik, J. Pachl and S. Yu. On the limit sets of cellular automata. *SIAM J. Computing*, 18(4), pp. 831-842, 1989.

[CY88] K. Cullik and S. Yu. Undecidability of CA classification schemes. *Complex Systems*, 2(2), pp. 177-190, 1988.

[Du94] B. Durand. Inversion of 2D cellular automata: some complexity results. *Theoretical Computer Science*, 134(2), pp. 387-401, November 1994.

[Ei74] S. Eilenberg. *Automata, Languages, and Machines*, Vol. A, Academic Press, NY, 1974.

[Ga97] P. Gacs. Deterministic computations whose history is independent of the order of asynchronous updating. Tech. Report, Computer Science Dept, Boston University, 1997.

[Ga82] H. Galperin. Succinct representation of graphs. Ph.D. Thesis, Princeton University, 1982.

[GW83] H. Galperin and A. Wigderson. Succinct representations of graphs, *Information and Control* 56, pp. 183-198, 1983.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman and Co., San Francisco, CA, 1979.

[GC86] M. Gouda and C. Chang. Proving Liveness for Networks of Communicating Finite State Machines. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 8(1): 154-182, pp. 1986.

[GM92] J.F. Groote and F. Moller. Verification of parallel systems via decomposition, *Proceedings of CONCUR'92*, LINC 630, pp. 62-76, 1992.

[Gu89] H. Gutowitz (Editor). *Cellular Automata: Theory and Experiment* North Holland, 1989.

[Ho84] C. Hoare. *Communicating Sequential Processes*. Prentice Hall International, 1984.

[Ho91] G. Holzmann. *Design and Validation of Computer Protocols*. Prentice Hall, 1991.

[HG99] B. Huberman and N. Glance. Evolutionary games and computer simulations. *Proc. National Academy of Sciences*, 1999.

[Hu73] H.B. Hunt III. On the Time and Tape Complexity of Languages. Ph.D. Thesis, Cornell University, Ithaca, NY, 1973. Some of the results of this thesis appear in On the Time and Tape Complexity of Languages I. were presented at the *Fifth Annual ACM Symposium on Theory of Computing (STOC)* 1973, pp. 10-19.

[Hu82] H.B. Hunt III. On the Complexity of Flowchart and Loop Program Schemes and Programming Languages. *Journal of the ACM (J ACM)* 29(1), pp. 228-249, 1982.

[HMS01] H.B. Hunt III, M.V. Marathe and R. Stearns. Strongly-local reductions and the complexity/efficient approximability of algebra and optimization on abstract algebraic structures, Technical Report Los Alamos National Laboratory, 2001.

[HR77] H.B. Hunt III and D.J. Rosenkrantz. On Equivalence and Containment Problems for Formal languages. *Journal of the ACM (J ACM)*. 24(3), pp. 387-396, 1977.

[HR78] H.B. Hunt III and D.J. Rosenkrantz. Computational Parallels Between the Regular and Context-Free Languages. *SIAM Journal on Computing (SICOMP)* 7(1), pp. 99-114, 1978.

[HRS76] H.B. Hunt III, D.J. Rosenkrantz, and T.G. Szymanski. On the Equivalence, Containment, and Covering Problems for the Regular and Context-Free Languages. *Journal of Computer and System Sciences (JCSS)* 12(2), pp. 222-268, 1976.

[HSM00] H.B. Hunt III, R.E. Stearns and M.V. Marathe Relational representability, local reductions, and the complexity of generalized satisfiability problems. Technical Report Los Alamos National Laboratory, 2000. Submitted for publication.

[HS76] H.B. Hunt III and T.G. Szymanski. Dichotomization, Reachability, and the Forbidden Subgraph Problem. *ACM Symposium on Theory of Computing (STOC)* 1976, pp. 126-134.

[HT94] D.T. Huynh and L. Tian. On deciding some equivalences for concurrent processes, *Theoretical Informatics and Applications* 28(1), pp. 51-71, 1994.

[KMW67] R.M. Karp, R.E. Miller, and S. Winograd. The organization of computations for universal recurrence equations, *Journal of the ACM (JACM)* 14(3), pp. 563-590, 1967.

[KMP95] Y. Kesten, Z. Manna and A. Pnueli. Verifying Clocked Transition Systems. *Hybrid Systems* 1995, pp. 13-40. Complete version in *Acta Informatica* 36(11), pp. 837-912, 2000.

[KO91] M. Kodialam and J.B. Orlin. Recognizing strong connectivity in periodic graphs and its relation to integer programming, *Proc. 2nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 131-135, 1991.

[KF75] A.N. Kolmogorov and S.V. Fomin. *Introductory Real Analysis*, Revised English edition translated and edited by R.A. Silverman. Dover Publications, Inc., NY, 1975.

[LP00] R. Laubenbacher and B. Pareigis. Finite Dynamical Systems. Technical report, Department of Mathematical Sciences, New Mexico State University, Las Cruces, 2000.

[Le86] T. Lengauer. Exploiting hierarchy in VLSI design, *Proc. Aegean Workshop on Computing*, F. Makedon et al., eds., *LNCS* 227, pp. 180-193, 1991.

[LW88] T. Lengauer and K. Wagner. Efficient solution of connectivity problems in hierarchically defined graphs, *SIAM Journal on Computing (SICOMP)* 17, pp. 1063-1089, 1988.

[LW92] T. Lengauer and K. Wagner. The correlation between the complexities of non-hierarchical and hierarchical versions of graph problems. *J. Computer and System Sciences (JCSS)*, Vol. 44, 1992, pp. 63-93.

[MB67] S. MacLane and G. Birkhoff. *Algebra*. Macmillan, NY, 1967.

[MH+98] M.V. Marathe, H.B. Hunt III, D.J. Rosenkrantz and R.E. Stearns. Theory of periodically specified problems:complexity and approximability. *Proc. 13th IEEE Conf. on Computational Complexity*, 1998.

[MH+97] M.V. Marathe, H.B. Hunt III, R.E. Stearns and V. Radhakrishnan. Complexity of hierarchically and 1-dimensional periodically specified problems. *AMS-DIMACS Volume Series on Discrete Mathematics and Theoretical Computer Science: Workshop on Satisfiability Problem: Theory and Application,* 35, November 1997.

[Ma98] B. Martin. A Geometrical Hierarchy of Graphs via Cellular Automata. Proc. MFCS'98 Satellite Workshop on Cellular Automata, Brno, Czech Republic, Aug. 1998.

[Mi99] R. Milner. *Communicating and Mobile systems: the π-calculus*. Cambridge University Press, 1999.

[Mo91] C. Moore. Generalized shifts: unpredictability and undecidability in Dynamical Systems. *Nonlinearity*, 4, pp. 199-230, 1991.

[Mo90] C. Moore. Unpredictability and undecidability in dynamical Systems. *Physical Review Letters*, 64(20), pp 2354-2357, 1990.

[MR99] H. Mortveit, and C. Reidys. Discrete sequential dynamical systems. *Discrete Mathematics,* 2000.

[NR98] C. Nichitiu and E. Remila. Simulations of Graph Automata. Proc. MFCS'98 Satellite Workshop on Cellular Automata, Brno, Czech Republic, Aug. 1998.

[Or84] J. Orlin. Some problems on dynamic/periodic graphs. *Progress in Combinatorial Optimization,* Academic Press, May 1984, pp. 273-293.

[Pa94] C. Papadimitriou. *Computational Complexity,* Addison-Wesley, Reading, Massachusetts, 1994.

[Pe97] W Peng. Deadlock Detection in Communicating Finite State Machines by Even Reachability Analysis. *Mobile Networks (MONET).* 2(3): 251-257, 1997.

[Ra92] A. Rabinovich. Checking equivalences between concurrent systems of finite state processes. *International Colloquium on Automata Programming and languages (ICALP),* LNCS 623, Springer, pp. 696-707, 1992.

[RSW92] Y. Rabinovich, A. Sinclair and A. Wigderson. Quadratic dynamical systems. *Proc. 33rd Annual Symposium on Foundations of Computer Science (FOCS),* pp. 304-313, Pittsburgh, October 1992.

[Ro99] C. Robinson. *Dynamical systems: stability, symbolic dynamics and chaos.* CRC Press, New York, 1999.

[Rk94] Z. Roka. One-way cellular automata on Cayley graphs. *Theoretical Computer Science,* 132(1-2), pp. 259-290, September 1994.

[RH93] D.J. Rosenkrantz and H.B. Hunt III. The complexity of processing hierarchical specifications. *SIAM J. Computing,* 22, 1993, pp. 323-350.

[Sc78] T. Schaefer. The Complexity of Satisfiability Problems. *Proc. 10th ACM Symposium on Theory of Computing* (STOC'78), 1978, pp. 216–226.

[SDB97] C. Schittenkopf, G. Deco and W. Brauer. Finite automata-models for the investigation of dynamical systems. *Information Processing Letters,* 63(3), pp. 137-141, August 1997.

[Sh97] S.K. Shukla. Uniform Approaches to the Verification of Finite State Systems. Ph.D. thesis, University at Albany-SUNY, Albany, NY, 1997.

[SHR96] S.K. Shukla, H.B. Hunt III and D.J. Rosenkrantz. HORNSAT, Model Checking, Verification and games (Extended Abstract). *Computer Aided verification Confenrence (CAV)* 1996, pp. 99-110.

[SH+96] S.K. Shukla, H.B. Hunt III, D.J. Rosenkrantz and R.E. Stearns. On the Complexity of Relational Problems for Finite State Processes. *International Colloquium on Automata Programming and Languages (ICALP),* pp. 466-477, 1996.

[Sm71] A. Smith. Simple computation-universal cellular spaces. *J. ACM,* 18(3), pp. 339-353, 1971.

[SM73] L.J. Stockmeyer and A.R. Meyer. Word problems requiring exponential time. *Proceedings 5th Annual ACM Symposium on Theory of Computing (STOC),* pp. 1-9, 1973.

[Su95] K. Sutner. On the computational complexity of finite cellular automata. *Journal of Computer and System Sciences,* 50(1), pp. 87-97, February 1995.

[vG90] R.J. van Glabbeek. The linear time-branching time spectrum. Technical Report CS-R9029, Computer Science Department, CWI, Centre for Mathematics and Computer Science, Netherlands, 1990.

[vG93] R.J. van Glabbeck. The linear time-branching time spectrum II (the semantics of sequential systems with silent moves). *LNCS* 715, 1993.

[VW86] M. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. *Proc. 1st IEEE Symposium on Logic in Computer Science,* pp.332-344, 1986.

[Wo86] S. Wolfram, Ed. *Theory and applications of cellular automata.* World Scientific, 1987.

5 Appendix:Transition Systems, Simulation Equivalences, and Simulation Pre-Orders

Let *Act* be a set of actions containing a special action τ called the **internal action** or *unobservable action.*

Definition 5.1 *A* **transition system** $\mathcal{T}$ *over Act is a triple* (S, D, s_1) *where S is a set of states,* $D \subset S \times Act \times S$ *is a set of transitions and* $s_1 \in S$ *is the starting state.* $\mathcal{T}$ *is said to be* **finite** *if both S and Act are finite.* $ext(\mathcal{T}) = Act - \{\tau\}$*is the set of* **external** *or* **visible** *actions. If* σ *is a sequence over Act then* $\hat{\sigma}$ *is the sequence over* $ext(\mathcal{T})$ *obtained by deleting all* τ *actions from* σ*. If* (p_1, a, p_2) *is in D then we write* $p_1 \xrightarrow{a} p_2$*. Also if* σ *is a sequence of actions such that there is a transition from state* p_1 *to state* p_2 *through some intermediate steps such that the sequence of actions is* σ*, then we write* $p_1 \stackrel{\sigma}{\Longrightarrow} p_2$ *and call this an* **extended step**. *Given* $\mathcal{T} = (S, D, s_1)$*, let* $\overline{D} = \{(p, a, p') \mid p \in S, a \in Act, p' \in S, \exists\sigma \in \tau^* a \tau^*$*, and* $p \stackrel{\sigma}{\Longrightarrow} p'\}$*. We call* $\overline{D}$ *the* **extended transition relation** *of* $\mathcal{T}$*.*[14]

Let $\mathcal{T}_1 = (S, D_1, s_1)$ and $\mathcal{T}_2 = (T, D_2, t_1)$ be two transition systems.

Definition 5.2 *Let* $R \subseteq (S \times T)$ *be a binary relation between S and T. R is a* **simulation** *if* $\forall(s, t) \in R$

$$\forall a \in Act,\ \forall s' \in S,\ (s, a, s') \in D_1 \Rightarrow \exists t' \in T,\ (t, a, t') \in D \text{ and } (s', t') \in R$$

In other words, for every labeled path in $\mathcal{T}_1$ *there is a corresponding labeled path in* $\mathcal{T}_2$ *with the same edge labels. We say R is a* bisimulation *if R and* R^{-1} *are both simulations. Here by* R^{-1}*, we simply mean inverting the pairs* $(s, t) \in R$ *to* (t, s)*. Two transition systems* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *are said to be* **bisimulation equivalent** *(denoted* $\mathcal{T}_1 \sim_{\text{bsim}} \mathcal{T}_2$*) iff there is a bisimulation relation R such that* $(s_1, t_1) \in R$*.* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *are said to be* **simulation equivalent** *(denoted* $\mathcal{T}_1 \preceq_{\text{sim}} \mathcal{T}_2$*) iff there is a simulation relation R such that* $(s_1, t_1) \in R$*.*

Definition 5.3 $B \subset S \times T$ *is a* **weak bisimulation** *relation from* $\mathcal{T}_1$ *to* $\mathcal{T}_2$ *if the following conditions are satisfied:*

1. $(s_1, t_1) \in B$*.*
2. $\forall(r, s) \in B$ *and* $a \in Act$*, if* $\exists\gamma \in \tau^* a \tau^*$ *such that* $r \stackrel{\gamma}{\Longrightarrow} r'$ *then*

   $\exists s' \exists\beta \in \tau^* a \tau^*$ *such that*
   $s \stackrel{\beta}{\Longrightarrow} s'$*,* $(r', s') \in B$*, and*
   if $\exists\beta \in \tau^* a \tau^*$ *with* $s \stackrel{\gamma}{\Longrightarrow} s'$*, then* $\exists r' \exists\gamma \in \tau^* a \tau^*$ *such that* $r \stackrel{\beta}{\Longrightarrow} r'$ *and* (r', s')*.*

If there exists a weak bisimulation from $\mathcal{T}_1$ *to* $\mathcal{T}_2$*, we sat that they are* **weak bisimulation equivalent***, denoted by* $\mathcal{T}_1 \sim_{wbsim} \mathcal{T}_2$*.*

Definition 5.4 *We say* γ *is a finite* **trace** *of a transition system* $\mathcal{T} = (S, D, s)$ *if there is a finite sequence* $\sigma \in Act^*$ *for which there is a state* $q \in S$ *such that* $s \stackrel{\sigma}{\Longrightarrow} q$ *and* $\gamma = \hat{\sigma}$*. Let* traces$(\mathcal{T})$ *denote the set of all finite traces of a transition system* $\mathcal{T}$*. We define* **trace preorder** *and* **trace equivalence** *as follows. If* traces$(\mathcal{T}_1) \subseteq$ traces$(\mathcal{T}_2)$*, then we say that* $(\mathcal{T}_1, \mathcal{T}_2)$ *are in* **trace preorder** *and denote this by* $(\mathcal{T}_1 \preceq_{trace} \mathcal{T}_2)$*. If* traces$(\mathcal{T}_1) =$ traces$(\mathcal{T}_2)$*, then we say that* $(\mathcal{T}_1, \mathcal{T}_2)$ *are* **trace equivalent** *and denote this by* $(\mathcal{T}_1 \sim_{trace} \mathcal{T}_2$*.*

[14]A transition system as defined here can be viewed as a directed edge-labeled graph: the edge-labels corresponds to the actions that take the system from one state to another. As defined here, D need not be a (partial) function; thus the system $\mathcal{T}$ can be *nondeterministic*.

In the context of parallel composition, to be defined below, a transition system is formally represented as a 4-tuple, rather than as a 3-tuple as in Definition 5.1. In this context, a transition system (S, D, s) over an action alphabet *Act* is represented as $(S, s, A, \rightarrow)$, where $A = Act - \{\tau\}$ and $\rightarrow = D$. Although the composition we define here is in the style of **CSP** [Ho84], the complexity bounds obtained also hold for other possible variants of parallel composition including composition of I/O automata and composition in **CCS** [Mi99]. Formally, the **parallel composition** of two transition systems $\mathcal{T}_1$ and $\mathcal{T}_2$,denoted $\mathcal{T}_1 \parallel \mathcal{T}_2$, is defined as follows.

Definition 5.5 *Let* $\mathcal{T}_1 = (S_1, D_1, A_1, s_1)$ *and* $\mathcal{T}_2 = (S_2, D_2, A_2, s_2)$. *Let* $\mathcal{T} = \mathcal{T}_1 \parallel \mathcal{T}_2 = (Q, q_0, A, \rightarrow)$. *Then* $Q = S_1 \times S_2$, $q_0 = (s_1, s_2)$ *and* $A = A_1 \cup A_2$. *The transition relation* $\rightarrow$ *is defined as follows: (i)If* $a \in A_1 \cap A_2$, $q_1 \xrightarrow{a} q_1'$, *and* $q_2 \xrightarrow{a} q_2'$, *then* $(q_1, q_2) \xrightarrow{a} (q_1', q_2')$.
(ii)If $a \in A_1$ *and* $a \notin A_2$ *and* $q_1 \xrightarrow{a} q_1'$, *then* $(q_1, q) \xrightarrow{a} (q_1', q)$.
(iii)If $a \notin A_1$ *and* $a \in A_2$ *and* $q_2 \xrightarrow{a} q_2'$, *then* $(q, q_2) \xrightarrow{a} (q, q_2')$.
(iv)If $q_1 \xrightarrow{\tau} q_1'$, *then* $(q_1, q) \xrightarrow{\tau} (q_1', q)$; *and if* $q_2 \xrightarrow{\tau} q_2'$ *then* $(q, q_2) \xrightarrow{\tau} (q, q_2')$.

Finally, we define the **hiding** operation on transition systems.

Definition 5.6 *Let* $\mathcal{T}_1 = (Q_1, q_0^1, A_1, \rightarrow_1)$ *be a transition system. Then* $\mathcal{T} =$ *hide a in* $\mathcal{T}_1$ *is the transition system* $(Q, q_0, A, \rightarrow)$ *where* $Q = Q_1$, $A = A_1 - \{a\}$, $q_0 = q_0^1$, *and the transition relation* $\rightarrow$ *is defined by–*

> *If* $a' \neq a$, *then* $q_1 \xrightarrow{a'}_1 q_2$ *implies* $q_1 \xrightarrow{a'} q_2$, *and*
> *if* $q_1 \xrightarrow{\tau}_1 q_2$ *or* $q_1 \xrightarrow{a}_1 q_2$, *then* $q_1 \xrightarrow{\tau} q_2$.

Let $A \subset Act$ *with* $|A| = n \geq 1$ *and* $A = \{a_1, \ldots, a_n\}$. *Then* **hide***A***in**$\mathcal{T}$ *means* **hide a_1 in***(***hide** a_2**in***(...* **in** *(***hide** a_n **in** $\mathcal{T}$*)...)).*