History-deterministic Timed Automata

Thomas A. Henzinger Karoliina Lehtinen Patrick Totzke

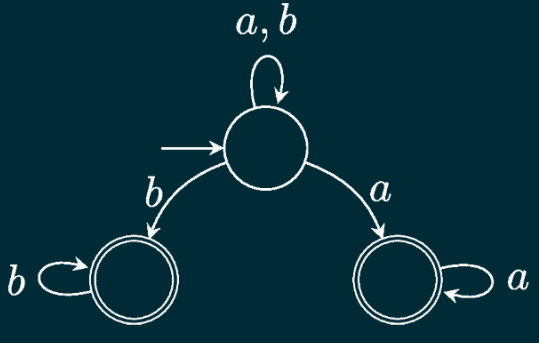
CONCUR 2022

PDF →

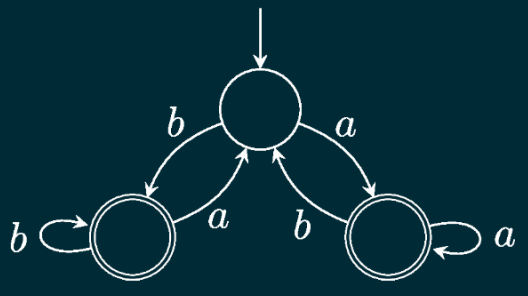
History-determinism (HD)

. . . is a restricted type of non-determinism where an automaton is HD if it can resolve choices only by looking at the past input..

A: Non-deterministic



B: Deterministic



$$L(A) = \{a,b\}^* b^\omega \cup \{a,b\}^* a^\omega = L(B)$$

aaabbaa$^\omega$

Definition

An automaton $\mathcal{A}$ is history-deterministic if there exist a resolver, a function

$r : T^* \times \Sigma \to T$ that maps every finite prefix run and letter to a next transition so that for every word w, if $w \in L(\mathcal{A})$ then the run r prescribes on it is accepting.

Equivalently, a resolver is a winning strategy for P2 in the Letter Game:
In every round i,

1. P1 picks a letter $a_i \in \Sigma$
2. P2 extends the run by an a-labelled edge.

P1 wins iff the word $w = a_1 a_2 a_3 \ldots \in L(A)$ but the run chosen by P2 is not accepting.

P

Equivalent Characterizations

TFAE. A finite ω-automaton is

- Good-for-Trees [Kupferman, Safra & Vardi '96]
- Good-for-Games [Henzinger & Piterman '06]
- Adaquate to Letter Games [Henzinger & Piterman '06]
- history-deterministic [Colcombert'09]
- Good for Composition with Alternating Automata [Colcombet '13]

Beyond ω-regular Properties

Quantitative automata: HD $\neq$ GfG [Boker & Lehtinen '21]
(For mean-payoff values, HD $\implies$ GfG but not vv.)

Pushdown automata: [Guha, Jecker, Lehtinen, and Zimmermann LICS,MFCS'21]

- DPDA $<$ HD-PDA $<$ PDA
- Solving Games with HD-PDA specs remains EXP-complete
- checking HDness is undecidable
- P1 always has pushdown strategies, P2 need not.

What can HD-X-Automata do?

Questions to ask about HD-X-automata

1. Does the HD-variant induce a new class of languages or does it coincide either with the deterministic or non-deterministic variants?
2. Is the HD variant more succinct than the deterministic automata?
3. Does the class have interesting closure properties?
4. Can we verify if a given system is history-deterministic?
5. What is the necessary internal complexity of resolver strategies?
6. What is the decidability/complexity status of verification problems?

Timed Automata = Finite-State + clocks ($\mathbb{R}$-valued variables)

- Recognise Languages over $\Sigma \times \mathbb{R}$

Fundamental Study

A theory of timed automata*

Rajeev Alur** and David L. Dill***
Computer Science Department, Stanford University, Stanford, CA 94305-2095, USA

Communicated by M.S. Paterson
Received November 1991
Revised November 1992

Abstract

Alur, R. and D.L. Dill, A theory of timed automata, Theoretical Computer Science 126 (1994) 183–235.

We propose *timed (finite) automata* to model the behavior of real-time systems over time. Our definition provides a simple, and yet powerful, way to annotate state-transition graphs with timing constraints using finitely many real-valued *clocks*. A timed automaton accepts *timed words* – infinite sequences in which a real-valued time of occurrence is associated with each symbol. We study timed automata from the perspective of formal language theory: we consider closure properties, decision problems, and subclasses. We consider both nondeterministic and deterministic transition structures, and both Büchi and Muller acceptance conditions. We show that nondeterministic timed automata are closed under union and intersection, but not under complementation, whereas deterministic timed Muller automata are closed under all Boolean operations. The main construction of the paper is an (PSPACE) algorithm for checking the emptiness of the language of a (nondeterministic) timed automaton. We also prove that the universality problem and the language inclusion problem are solvable only for the deterministic automata: both problems are undecidable (Π_1^1-hard) in the nondeterministic case and PSPACE-complete in the deterministic case. Finally, we discuss the application of this theory to automatic verification of real-time requirements of finite-state systems.

* Preliminary versions of this paper appear in the *Proc. 17th Internat. Colloq. on Automata, Languages, and Programming* (1990), and in the *Proc. of the REX workshop "Real-Time: Theory in Practice"* (1991).

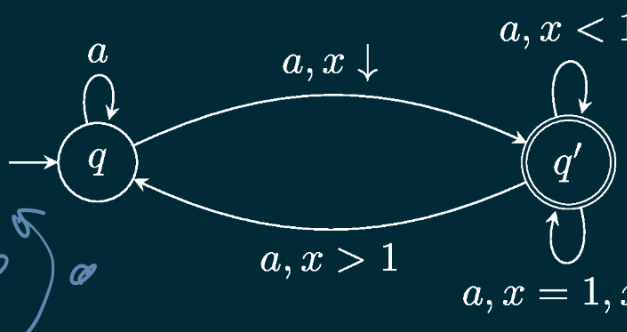
** Current address: AT&T Bell Laboratories, 600 Mountain Avenue, Room 2D-144, Murray Hill, NJ 07974.

*** Supported by the National Science Foundation under grant MIP-8858807, and by the United States Navy, Office of the Chief of Naval Research under grant N00014-91-J-1901. This publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement of this work should be inferred.

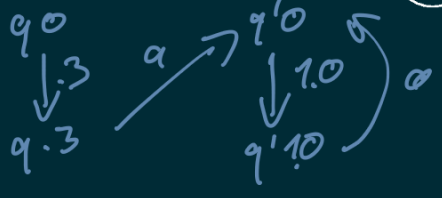
SSDI 0304-3975(93)E0097-N

- Region abstraction
  → Emptiness in PSPACE
- Universality is undecidable
- DTA ⊂ NTA
- Not closed under complement
- Lots more!

Example 1-clock TA, $\Sigma = \{a\}$

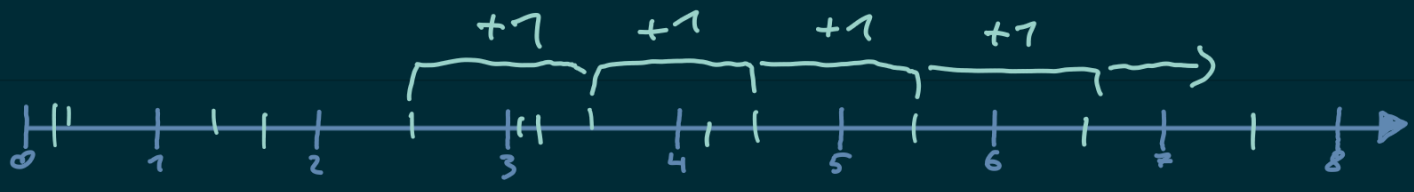


HD!
resolve: pick the oldest FV



Acceptance condition:
finitely often state q

L(q) = eventually a's appear with distance 1.



History-deterministic Timed Automata

Thomas A. Henzinger
IST Vienna, Austria

Karoliina Lehtinen
CNRS, Aix-Marseille University, University of Toulon, LIS

Patrick Totzke
University of Liverpool, UK

Abstract

We explore the notion of history-determinism in the context of timed automata (TA). History-deterministic automata are those in which nondeterminism can be resolved on the fly, based on the run constructed thus far. History-determinism is a robust property that admits different game-based characterisations, and history-deterministic specifications allow for game-based verification without an expensive determinization step.

We show yet another characterisation of history-determinism in terms of fair simulation, at the general level of labelled transition systems: a system is history-deterministic precisely if and only if it fairly simulates all language smaller systems.

For timed automata over infinite timed words it is known that universality is undecidable for Büchi TA. We show that for history-deterministic TA with arbitrary parity acceptance, timed universality, inclusion, and synthesis all remain decidable and are ExpTime-complete.

For the subclass of TA with safety or reachability acceptance, we show that checking whether such an automaton is history-deterministic is decidable (in ExpTime), and history-deterministic TA with safety acceptance are effectively determinizable without introducing new states.

1 ↓ 2

3

4 open questions

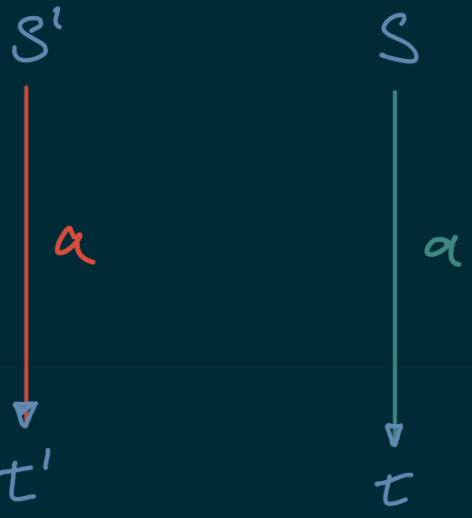
2012 ACM Subject Classification Theory of computation → Formal languages and automata theory

Keywords and phrases Timed Automata, History-determinism, Good-for-games, fair simulation, synthesis

Fair Simulation

(a game played on $S' \times S$)

In each round
- P1 moves on S'
- P2 moves on S using the same letter



P2 wins if
- run 1 is rejecting

OR

- run 2 is accepting

▶ **Theorem 4.** *For every fair LTS S and initial state q the following are equivalent:*

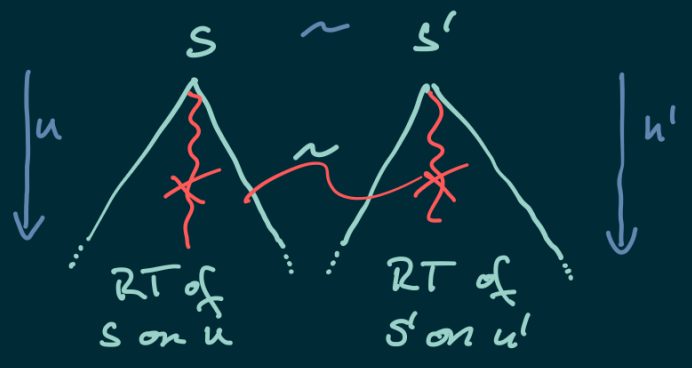
1. *S is history-deterministic.*
2. *For all complete fair LTS S' with initial state q', $q' \subseteq_L q$ if and only if $q' \preceq q$.*

Reducible to Timed Parity Games
(EXP-complete [D'souza & Madhusudan 2002])

Corollary For TA A and HD-TA B
Language inclusion $A \subseteq B$ is EXP-complete

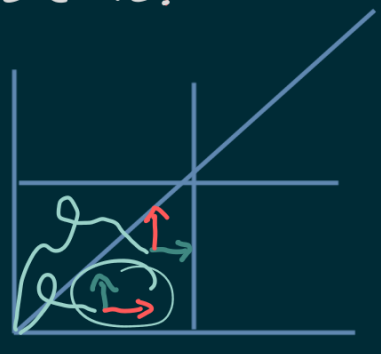
Towards Determinisation ...

Lemma Consider region eq. configs $sv \sim s'v'$.
For every word u exists a word u' with

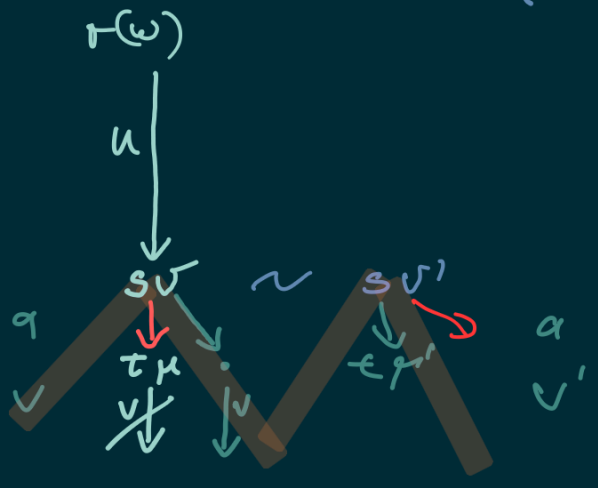


Theorem HD-Safety TA admit region-based resolvers.

Proof Pick any resolver r' and let r
resolve in R just like r' for some config. in R
Suppose r is no resolver.
$\Rightarrow$ $r(\omega)$ is not accepting for some $\omega \in L$.



$\Rightarrow$ at some position r makes a mistake
(remaining suffix v of $w = uav$ not recognisable)



$\Rightarrow$ For $sv' \sim sv$, τ picks sv'.

$\Rightarrow$ RT of sv' on av' $\sim$ RT of sv on av

Now, $av \in L(s\tau)$ assumption
$av' \in L(s\tau')$ RTs are eq.
$v' \in L(t\mu')$ τ' is resolver
$v \in L(t\mu)$ RTs are eq.

Corollary HD-Safety TA can be determinised

proof Hard-code regions into transition guards:

- no new states nor letters,
- |Regions| many new transitions

▶ **Definition 19** (Timed synthesis game). *Given a timed language $L \subseteq (\Sigma_I \times \Sigma_O)^\omega_T$, the synthesis game for L proceeds as follows. At turn i:*

- *Player I plays a delay d_i and a letter $a_i \in \Sigma_I$*
- *Player II plays a letter $b_i \in \Sigma_O$.*

Player II wins if $d_0\binom{a_0}{b_0}d_1\binom{a_1}{b_1}\ldots \in L$ or if time does not progress. If Player II has a winning strategy in the synthesis game, we say that L is realisable.

▶ **Theorem 20.** *Given a history-deterministic timed parity automaton $\mathcal{T}$, the synthesis game for $L(\mathcal{T})$ is decidable and* ExpTime*-complete.*

Proof Idea: Have output alphabet Σ_o encode moves in T.
(P2 proposes an accepting run)

This works for
- finite
- quantitative
- pushdown ...

$\Rightarrow$ Synthesis game for deterministic Parity TA which is EXP-complete [DM 2002]

Corollary Inclusion is EXP-c for HD-TA

On Checking HD-ness

▶ **Theorem 16.** *Given a safety or reachability TA, deciding whether it is history-deterministic is decidable in* ExpTime.

Can be shown using "token games" [xx]
(P1 builds several runs publicly)

Lemma
- G_K are in EXP for Parity TA (via timed Parity games [DM 2002])
- $G_K \equiv G_2$

For Safety, $LG_1 \equiv G_1$

For Reachability, $LG_1 \equiv G_2$ (requires finite branching for P2)

These rely on determinacy of LG_2, which is open for timed Parity.

Open Problem

- DTA $\subsetneq$ (safety) HDTA $\subsetneq_{\text{fin}}$ NTA
  - DTA $\subsetneq$ coBüchi HDTA

- Can resolvers be region-based for Reach/Büchi?
- are timed Parity languages Borel?
- Deciding HD-ness beyond Reach/Safety
- Complexities in dim. 1

Dzięki!