The algorithmic analysis of hybrid system

Authors: R.Alur, C. Courcoubetis etc.
Course teacher: Prof. Ugo Buy
Xin Li, Huiyong Xiao
Nov. 13, 2002

Summary

- What's a hybrid system?
- Definition of Hybrid Automaton
- Subclasses
- Examples
- Reachability problems of Linear Hybrid Automata

What’s a hybrid system?

- A hybrid system consists of a discrete system with an analog component.
- For example:
  - An automobile engine whose fuel injection (continuous) is regulated by a microprocessor (discrete).
  - A digital controller of an analog plant.
  - Medical equipments, manufacturing controllers, and robots etc.

What’s a hybrid system? (cont’d)

- A run of a hybrid system is a sequence of steps.
- Within each step the system state evolves continuously according to a dynamical law until a transition occurs.
- With time elapsing, when the variable changes to break the invariant condition, state transitions will take place instantaneously.

Hybrid Automaton

- Intuitively – the plant example:
  - The discrete state of the controller → vertices of a graph (*locations*)
  - The discrete dynamics of the controller → edges of the graph (*transitions*)
  - The continuous state of the plant → points in R^n
  - The continuous dynamics of the plant → differential equations (*activities*)
  - Each transition may cause a discrete change in the state of the plant, as determined by a *synchronization label*.
  - The behavior of the controller depends on the state of the plant: when violating the *invariant condition*, a transition happens.

Formal definition for Hybrid Automaton

$\mathcal{H} = (Loc, Var, Lab, Edg, Act, Inv)$

Initial conditions

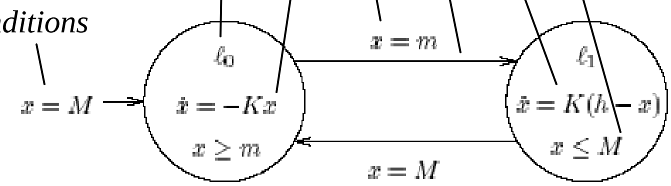

Figure 1: Thermostat

- ℓ_0: $x(t) = ?e^{-Kt}$, so $\dot{x} = dx/dt = -K?e^{-Kt} = -Kx$
- ℓ_1: $x(t) = ?e^{-Kt} + h(1-?e^{-Kt})$, $\dot{x} = K(h-x)$

Locations

- A unique *name* identifying each location.
- *State invariants:*
  - While the control stays in a location, the variables must satisfy the invariant conditions.
  - The state invariants decide how long the automaton can stay in the location.
- *Flow relations:*
  - How continuous variables evolve.

Arcs

- Each arc represents a state transition from a source location to a target location.
- *Synchronization labels:*
  - Two hybrid automaton synchronize on the common set of Synchronization labels.
- *Guarded assignments:*
  - Represent jump conditions using guards and update the state variables by assignments.
  - Assuming two variables x_1, x_2, and x'_i refers to the value of x_i after the transition: "$x_1 = x_2$, $x_1 := x_2$" stands for "$x_1 = x_2 \wedge x'_1 = 2x_2 \wedge x'_2 = x_2$".
  - "$x = m$" stands for "$x = m \wedge x' = x$".

Linear Hybrid Automaton

- Two concepts:
  - A ***linear term***: a linear combination of the variables in *Var* with integer coefficients.
  - A ***linear formula***: a boolean combination of inequalities between *linear terms* over *Var*.
- ***Linear Hybrid Automaton***: a time-deterministic hybrid system whose activities, invariants, and transition relations can be defined by linear expressions over the set *Var* of variables.

Special cases of Linear Hybrid Automaton

- ***Discrete system***: All variables are *discrete*.
  - x is a *discrete variable*, if $Act(l, x)=0$ for each $l \in Loc$.
- ***Finite-state system***: All variables are *propositions*.
  - x is a *proposition variable*, if $\mu(e, x) \in \{0,1\}$ for each $e \in Edg$.
- ***Timed Automaton***:
  - 1) All variables are *propositions* or *clocks*,
  - 2) the linear expressions are boolean combinations of inequalities of the form $x \# c$ or $x-y \# c$, where c is a nonnegative integer and $\# \in \{<, =, =, >, =\}$.
  - x is a *clock*, if $Act(l, x)=1$ for each l, and $\mu(e, x) \in \{0,x\}$ for each e.

Special cases of Linear Hybrid Automaton

- ***Multirate timed system***: All variables are *propositions* or *skewed clocks*.
  - x is a *skewed clock*, if $Act(l, x)=k$ for each l, where $k \in Z$; and $\mu(e, x) \in \{0,x\}$ for each e.
  - *N-rate timed system*: a multirate timed system whose skewed clocks proceed at n different rates.
- ***Integrator system***: All variables are *propositions* or *integrators*.
  - x is an *integrator*, if $Act(l, x)=\{0, 1\}$ for each l and $\mu(e, x) \in \{0,x\}$ for each e.
- **Parameter**:
  - x is an *parameter*, if $\mu(e, x) = x$ for each e.
  - We obtain *parameterized* versions of above system by admitting parameters

Example: A mutual-exclusion protocol

- The asynchronous shared-memory system that consists of two processes P_1 and P_2 with atomic read and write operations.
- Each process has a critical section and at each time instant, at most one of the two processes is allowed to be in its critical section.

Example: A mutual-exclusion protocol

repeat
  repeat
    await k=0
    k := i
    delay b
  until k = i
    Critical section
    k:=0
forever

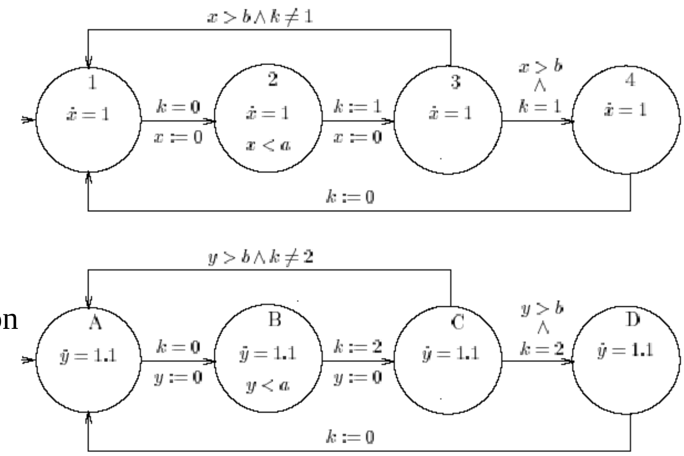

Figure 3: Mutual-exclusion protocol

Reachability problems for Linear Hybrid Automaton

- If there is a run of system $\mathcal{H}$ that starts in state s and ends in state s', then the state s' is *reachable* from the state s, written $\sigma \mapsto^* \sigma'$.
- *reachability question*: if $\sigma \mapsto^* \sigma'$ for two given states s and s' of a hybrid system.
- **Theorem 3.1.** *The reachability problem is decidable for simple multirate timed systems.*
- **Theorem 3.2.** *The reachability problem is undecidable for 2-rate timed systems.*
- **Theorem 3.3.** *The reachability problem is undecidable for simple integrator systems.*

The runs of a hybrid system

- A finite or infinite sequence: ([$\mathcal{H}$] is the set of runs of $\mathcal{H}$)

$$\rho: \quad \sigma_0 \mapsto^{t_0}_{f_0} \sigma_1 \mapsto^{t_1}_{f_1} \sigma_2 \mapsto^{t_2}_{f_2} \cdots$$

- where states $s_i = (\ell_i, v_i) \in S$, nonnegative reals $t_i \in R^{=0}$, and activities $f_i \in Act(\ell_i)$, such that for all $i = 0$:
  - 1. $f_i(0) = v_i$,
  - 2. for all $0 = t = t_i, f_i(t) \in Int(\ell_i)$,
  - 3. the state s_{i+1} is a <u>*transition successor*</u> of the state $s_i' = (\ell_i, f_i(t_i))$.

- For time-deterministic systems, we can omit the subscripts f_i from the *next relation.*
- The run ? diverges if ? is infinite and the infinite sum $S_{i=0}\, t_i$ diverges.

The following slides are presented by Xin Li

The algorithmic analysis of hybrid system

- Research motivation
- Background
- Forward analysis
- Backward analysis
- Discussion

Research Motivation

- Purpose of automatic verification: Given a system and a correctness property, does the system satisfy the property?

Research Motivation

- Modeling of hybrid systems:

  The runs of a hybrid system: the state can change in two ways:

	Nature	Location	Valuation	
Jump	Instant & discrete	Change	Transition Relation	Followed by new **flow**
Flow	Continuous	No Change	Activities	Until invariant becomes false

Research Motivation

- Reachability issue: Now that a run of a hybrid system is a finite/infinite sequence of "**flow**s" and "jumps", can we guarantee a system is safe?

"The reachability problem is central to the verification of hybrid systems… a set $R \subseteq \Sigma$ of states is an invariant of the hybrid system H iff no state in Σ-R is reachable from an initial state of H."

Research Motivation

- Decidability issue: Are we always able to know if a hybrid system is safe or unsafe?

Reachability analysis is a search over an infinite state space. For linear hybrid system, the termination of this procedure is not guaranteed. Additional techniques (approximation analysis) may help the convergence of this process.

Background

- Sets

  $\in$ membership $\subseteq$ subset $\cap$ set intersection $\cup$Set union – set difference

- Quantifiers

  Notation:($\forall$ x $P(x)$) “for all x $P(x)$ is true.”

  Notation:($\exists$ x $P(x)$) “there exists an x such that $P(x)$ is true.”

- Proposition Logic:

  A disjunction $\vee$ is true if either of its parameters are true.

  A conjunction $\wedge$ is true only when both parameters (called conjuncts) are true.

Forward Analysis

- General procedure of verification process:

  Start from the initial state, then trace the state change as system runs, finally check if this process converge.

- State change during **flow** process:

The *forward time closure* $\langle P \rangle_l'$ of P at l is the set of valua-tions that are reachable from some valuation $v \in P$ $l \in Loc$, valuation $P \subseteq V$, $\in P$ by letting time progress.

$v' \in \langle P \rangle_l'$ iff $\exists v \in V$, $t \in R^{\geq 0}. v \in P \wedge \underline{tcp_l[v](t)} \wedge \underline{v' = \varphi[v](t)}$

Forward Analysis

What does it mean?

Invariant factor: $tcp_l[v](t)$: *time can progress*: iff $\forall\, 0 \leq t' \leq t$, $\varphi[v](t) \in Inv(l)$. $\varphi[v](t)$: activity at time t.

- State change during **jump** process:

$v' \in post_e[P]$ iff $\exists v \in V. v \in P \cap Inv(l) \wedge (v, v') \in \mu \wedge v' \in Inv(l)$

μ: transition relation. For a linear hybrid system:

$(v, v') \in \mu$ iff $v(y) \wedge \forall\, x \in Var.\ v(\alpha_x) \leq v'(x) \leq v(\beta_x)$

y Þ { x:= $[\alpha_x, \beta_x]$| x ∈ Var}

Forward Analysis

- Extension to “region” — a set of state:

  flow: $\langle R\rangle' = {}_{l \in loc}\cup(l, \langle R_l\rangle_l)'$

  jump: $post[R] = {}_{e=(l,l')\in edge}\cup(l', post_e[R_l])$

  Combine them together, for the *i* step:

  $P_{i+1} = post_e[\langle P_i\rangle'_{li}]$

Proposition 4.1: least fixpoint.

Proposition 4.2: linearity of sets.

Forward Analysis

- Example:

Prove $y \geq 60 \Rightarrow 20z \leq y$.

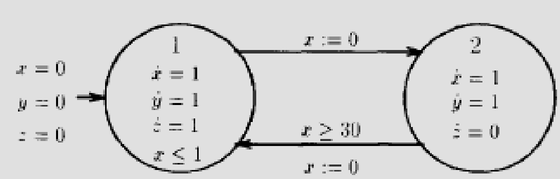

Fig. 4. Leaking gas burner.

Forward Analysis

- Analysis:

  Initial state defined by linear formula:

  $\psi_I = (pc = 1 \wedge x = y = z = 0)$ pc: control variable

  At location 1: $\psi_1 = \langle x = y = z = 0 \vee post_{(2,1)}\psi_2] \rangle_1'$

  At location 2: $\psi_2 = \langle false \vee post_{(1,2)[}\psi_1] \rangle_2'$

  For step i: $\psi_{1,i} = \psi_{1,i-1} \vee \langle post_{(2,1)}[\psi_{2,i-1}] \rangle_1'$

  $\psi_{2,i} = \psi_{2,i-1} \vee \langle post_{(1,2)}[\psi_{1,i-1}] \rangle_1'$

Forward Analysis

- Result:

$\psi_R = (pc = 1 \wedge \psi_1)\ (pc = 2 \wedge \psi_2)$

$\psi_1 = (x \leq 1 \wedge x=y=z) \vee$ **(x £ 1 Ù x £ z Ù y + 30x ³31z)**

$\psi_2 = (z \leq 1 \wedge y=x+z \wedge x \geq 0) \vee$ **y ³ x + 31z - 30**

Therefore, $y \geq 60 \Rightarrow 20z \leq y$.

Backward Analysis

- An “mirror” approach of forward analysis.

The differences:

  - The initial state is the “unsafe condition”.
  - “Propagation” is done “backward”
  - It takes six iterations to converge.
  - Converge conditions do not contain that initial state, so the original statement proven.

Discussion

- Other approaches:
  - Approximation analysis.
  - Minimization.
- Questions…