Descriptions of Hybrid Systems

EE219C
Jia Zou
4/25/2007
Question:
- How do we describe hybrid systems?

One intuitive way to do describe HS
- Hybrid automata
- Is this a good idea?

Other approaches…
- Lazy linear hybrid automata
What is a Hybrid System

- **Discrete program with an analog environment**
  - How do we formally verify hybrid systems?
- **Modeled as a finite automaton with a set of variables.**
  - Vertices => continuous activities
  - Edges => discrete transitions

**H = (Loc, Var, Lab, Edg, Act, Inv)**
- State = (l,v), l ∈ Loc, v ∈ Valuations
- Stuttering label ∈ Lab
- (l,a,µ,l′) ∈ Edg
  - An edge is enabled in state (l, v) if for some v′ ∈ V, (v,v′) ∈ µ
  - (l′,v′) is the transition successor of (l,v)
Hybrid System Example

- Leaky gas burner
  - Loc: leak, no leak
  - Var: x, y, z.
  - Inv: x <= 1
  - Transition relation specified by guard
    - \( \mu = \{\text{NULL}, (x < 30, x \geq 30)\} \)

![Figure 4: Leaking gas burner](image)
Hybrid System Transitions

A run \([H]\) of a hybrid system:

- \(\rho : \sigma_0 \mapsto_{f_0}^{t_0} \sigma_1 \mapsto_{f_1}^{t_1} \sigma_2 \mapsto_{f_2}^{t_2} \ldots\)

- \(\sigma_i = (l_i, v_i)\)
- \(t_i \in \mathbb{R}^{\geq 0}\)
- \(f_i \in \text{Act}(l_i)\) \(f_i \in \text{Inv}(l_i)\)

Properties:
- If all Act are smooth functions, then all runs are piecewise smooth
- A run diverges if it’s infinite and \(\sum_{i \geq 0} t_i \to \infty\)
Run of Hybrid System

- Discrete and instantaneous transition of locations.
- Time delay that changes only the value of the variables, according to Act.
- Time-can-progress function to switch between transition-step and time-step
Transition System

- Hybrid system as a transition system:
  - \( T_H = (\Sigma, \text{Lab} \cup \mathbb{R}^{\geq 0}, \rightarrow) \)

- Two types of step relations \( \rightarrow \)
  - Transition-step relation \( \rightarrow^a \)
    \[
    (l, a, \mu, l') \in \text{Edg} \quad (v, v') \in \mu \quad v \in \text{Inv}(l), v' \in \text{Inv}(l')
    \]
    \[
    (l, v) \rightarrow^a (l', v')
    \]
  - Time-step relation \( \rightarrow^t \)
    \[
    f \in \text{Act}(l) \quad f(0) = v \quad \forall 0 \leq t' \leq t. f(t') \in \text{Inv}(l)
    \]
    \[
    (l, v) \rightarrow^t (l, v')
    \]

- Time can progress
  \[
  tcp_l[v](t) \iff \forall 0 \leq t' \leq t. \varphi_l[v](t') \in \text{Inv}(l)
  \]
Linear Hybrid Systems

- Act, Inv, Transition relations are linear.
- Special cases:
  - Act(l, x) = 0 for each location. x: discrete variable.
    - All variables discrete ⇔ discrete system
  - µ(e,x) ∈ {0,1} for each transition e ∈ Edg. x: proposition.
    - All variables are propositions ⇔ finite-state system
  - Act(l, x) = 1 for each location l and µ(e,x) ∈ {0,x} for each transition e. x: clock
More About Special Cases

- Act(l, x) = k for each location l and µ(e,x) ∈ {0,x} for each transition e. x: skewed clock
  - All variables are propositions are skewed clocks ⇔ Multirate timed system.
  - N-rate timed system: skewed clocks proceed at n different rates.
- Act(l, x) ∈ {0,1} for each l && µ(e,x) ∈ {0,x} for each e. x: integrator.
  - All variables are integrators: integrator system
- µ(e,x) = x for each e. x: parameter (symbolic constant)
Linear Hybrid System Example

Leaky gas burner

- Multirate timed system
  - X: clock that stores time in current location
  - Y: global clock
  - Z: integrator

![Diagram of Leaking gas burner]
Parallel Composition of HS

- $H_1 = (\text{Loc}_1, \text{Var}, \text{Lab}_1, \text{Edg}_1, \text{Act}_1, \text{Inv}_1)$
- $H_2 = (\text{Loc}_2, \text{Var}, \text{Lab}_2, \text{Edg}_2, \text{Act}_2, \text{Inv}_2)$
  - Common set of $\text{Var}$
  - Two hybrid systems synchronized by $\text{Lab}_1 \cap \text{Lab}_2$
- $H_1 \times H_2 = (\text{Loc}_1 \times \text{Loc}_2, \text{Var}, \text{Lab}_1 \cup \text{Lab}_2, \text{Edg}, \text{Act}, \text{Inv})$
  - $((l_1, l_2), a, \mu, (l'_1, l'_2)) \in \text{Edg}$
  - $(l_1,a_1,\mu_1,l'_1) \in \text{Edg}_1$ and $(l_2,a_2,\mu_2,l'_2) \in \text{Edg}_2$
  - Either $a_1=a_2=a$, or $a_1 \notin \text{Lab}_2$ and $a_2 = \tau$, or $a_2 \notin \text{Lab}_1$ and $a_1 = \tau$
  - $\mu = \mu_1 \cap \mu_2$
- $\text{Act}(l_1,l_2) = \text{Act}_1(l_1) \cap \text{Act}_2(l_2)$
- $\text{Inv}(l_1,l_2) = \text{Inv}_1(l_1) \cap \text{Inv}_2(l_2)$
- $[H_1 \times H_2]_{\text{Loc}_1} \subseteq [H_1]$  $[H_1 \times H_2]_{\text{Loc}_2} \subseteq [H_2]$
Reachability Problem for Linear Hybrid Systems (LHS)

- A LHS is **simple** if all local invariants and transition guards are in the form $x \leq k$ or $k \leq x$.

Reachability problem is
- decidable for simple multirate timed systems.
  - Our previous example
- Undecidable for 2-rate timed system
- Undecidable for simple integrator systems

![Diagram of Leaking Gas Burner](image.png)
Forward Analysis Graphical Representation

Universe for V

All I in Loc All I in Loc All I in Loc
Verification of LHS

- **Forward Analysis** – P is set of valuation
  - **Forward time closure of P at l:**
    \[ \nu' \in \langle P \rangle_l \Leftrightarrow \exists \nu \in V, t \in \mathbb{R}^{\geq 0}. \nu \in P \land tcp[\nu](t) \land \nu' = \varphi_l[\nu](t) \]

  - **Postcondition of P with respect to e:**
    \[ \nu' \in \text{post}_e[P] \Leftrightarrow \exists \nu \in V, \nu \in P \land (\nu, \nu') \in \mu \]

  - **A set of states is called a region:**
    \[ \langle R \rangle = \bigcup_{l \in \text{Loc}} (l, \langle R_l \rangle_l) \]
    \[ \text{post}[R] = \bigcup_{e = (l, l') \in \text{Edg}} (l', \text{post}_e[R_l]) \]
More Forward Analysis

- Symbolic run of linear hybrid system H:
  \[ \rho = (l_0, P_0)(l_1, P_1)\ldots(l_i, P_i) \ldots \]
  \[ P_{i+1} = \text{post}_e[\langle P_i \rangle_{l_i}] \]

- The region \((l_{i+1}, P_{i+1})\) is reachable from \((l_0, P_0)\)

- Reachable region \( I \mapsto * \)
  \[ \sigma \in (I \mapsto *) \iff \exists \sigma' \in I. \sigma' \mapsto *\sigma. \]

- Reachable region of I is the least fixpoint of:
  \[ X = \langle I \bigcup \text{post}[X] \rangle \]
  \[ X_i = \langle I_i \bigcup \bigcup_{e=(l',I) \in Edg} \text{post}_e[X_{l'}] \rangle \]

- Lemma:
  - If \( P \) is a linear set of valuations, then for all \( I \) and \( e \), both \( \langle P \rangle_I \) and \( \text{post}_e[P] \) are linear sets of valuations – makes sure the system is verifiable
Forward Reachability Example

\[ \varphi_{1,0} = \langle x = y = z = 0 \rangle_1 = (x \leq 1 \land y = x = z) \]
\[ \varphi_{2,0} = false \]

\[ \varphi_1 = \langle x = y = z = 0 \lor post_{(2,1)}[\varphi_2] \rangle_1 \]
\[ \varphi_2 = \langle false \lor post_{(1,2)}[\varphi_1] \rangle_2 \]
\[ \varphi_{1,i} = \varphi_{1,i-1} \lor post_{(2,1)}[\varphi_{2,i-1}] \]
\[ \varphi_{2,i} = \varphi_{2,i-1} \lor post_{(1,2)}[\varphi_{1,i-1}] \]
\[ \varphi_{1,1} = \varphi_{1,0} \lor post_{(2,1)}[\varphi_{2,0}]_1 = \varphi_{1,0} \]
\[ \varphi_{2,1} = \varphi_{2,0} \lor post_{(1,2)}[\varphi_{1,1}]_2 = post_{(1,2)}[x \leq 1 \land y = x = z = 0]_2 \]
\[ = \langle x = 0 \land y \leq 1 \land z = y \rangle_2 = (z \leq 1 \land y = z + x) \]

\[ \vdots \]

Prove: \( y \geq 60 \rightarrow 20z \leq y \)
Backward Analysis

- Backward time closure of $P$ at $l$:
  \[ v' \in \langle P \rangle_l \iff \exists v \in V, t \in \mathbb{R}^0. v = \phi_l[v](t) \land v \in P \land tcp[v'](t) \]

- Precondition of $P$ with respect to $e$:
  \[ v' \in \text{pre}_e[P] \iff \exists v \in V, v \in P \land (v', v) \in \mu \]

- Extension to a region:
  \[ \langle R \rangle = \bigcup_{l \in \text{Loc}} (l, \langle R_l \rangle_l) \]
  \[ \text{pre}[R] = \bigcup_{e=(l', l) \in \text{Edg}} (l', \text{pre}_e[R_l]) \]

- Initial region $I$ is the least fixpoint of:
  \[ X = \langle R \cup \text{pre}[X] \rangle \]
  \[ X_I = \langle R_I \cup \bigcup_{e=(l, l) \in \text{Edg}} \text{pre}_e[X_I] \rangle \]

- Lemma:
  - If $P$ is a linear set of valuations, then for all $l$ and $e$, both $\langle P \rangle_l$ and $\text{pre}_e[P]$ are linear sets of valuations – makes sure the system is verifiable
Timed Automata = simple multirate
- Nondeterministic
- Does not make transition as long as the Inv are satisfied.
- PSPACE complexity
Communicating Timed Automata

- Cooperations among processes to construct a state transition
- Channel concept introduced
  - Improve modularity of model description
  - Communicating real-time state machines.
- Monitor + Controller
- No distinction between sender and receiver
  - Model Bus Collisions

The model of gate-monitor-controller.
Hybrid Automata

- Generalization of timed automata
- N-rate timed system
- Undecidable => not subject to algorithmic verification
Logics

- Logic formulas used to describe system behavior
- System description and specifications put into the same language
  - Descriptions as axioms
  - Specification as theorems
- Soundness + completeness check

Pro:
- Small models that can prove/disprove theorems quickly
- Semi-decision procedures that prove first-order logics

Con:
- Becomes impossible for large scale systems
- We can’t build a theorem proving machine in general
Models Dealing With Real-Time Systems

- **Case: Train approaching, poles come down**
  - Linear-time Propositional Temporal Logic
    - \( G(\text{approach} \Rightarrow F \text{ down}) \)
  - LTL with with clock time
    - \( \forall x \exists y G((T = x \land \text{approach}) \Rightarrow F(T = y \land (y - x \leq 300) \land \text{down})) \)
  - Timed Propositional Temporal Logic
    - \( G_x.(\text{approach} \Rightarrow F_y.(y - x \leq 300) \land \text{down}) \)
      - Different from LTL with clock
  - Metric Temporal Logic
    - \( G(\text{approach} \Rightarrow F_{\leq 300} \text{down}) \)
  - Asynchronous PTL
    - \( G[x,y]((x+2)\leq(y+1)) \)
  - CTL
    - \( \forall G(\text{approach} \Rightarrow \forall F(\text{down})) \)
  - TCTL (most used)
    - \( \forall G(\text{approach} \Rightarrow \forall F_{\leq 300}(\text{down})) \)
Timed Process Algebra

- Three grammar rules
  - Wait $t$: wait for $t$ time units
  - $P_1 \triangleright t > P_2$: $P_1$, until time $t$, when no synchronization has happened, then $P_2$
  - $P_1 \triangleright t \downarrow P_2$: $P_1$ until time $t$, no matter what, $P_2$. 
Others

- **Timed Petri Nets**
  - Places, Tokens, Transitions
  - Many extension to tackle its inexpressiveness

- **Statecharts**
  - Describe behavioral hierarchies of untimed concurrent systems
Lazy Linear Hybrid Automata

- **Definition:**
  - A class of LHA where discrete time behavior can be computed and represented as finite state automata.

- **Simplifying by sampling.**

- **Why does this abstraction makes sense?**
- **Undecidable => decidable?**
Lazy Linear Hybrid Automata

- **Requirements:**
  - Periodic sampling
  - Finite precision – bound on the value

- **Formulation:**
  - On the control side:
    - \( A = (Q, \text{Act}, q_{\text{in}}, V_{\text{in}}, D, \varepsilon, \{p_q \}_{q \in Q}, B, \Rightarrow) \)
      - \( \Rightarrow : (Q \times \text{Act} \times \text{Grd} \times Q) \)
      - ...D closely related to \( \varepsilon \)?
  - On the system side:
    - Value
    - Guard
    - No states?
Transition Relations

- Configurations:
  - \((q, V, q'), q, q'\) are current and previous control states, \(V\) is set of actual values for Var
    - \(\text{Init: } (q_{\text{in}}, V_{\text{in}}, q_{\text{in}}')\)
    - \(a: \text{action}\)
    - \(\tau: \text{silent action}\)
    - \((q, V, q') = (a) > (q_1, V_1, q_1') \iff q_1' = q, q=(a, g)>q_1\)
      - \(t_1, t_2\) are delays. 2 delays to separate two rates
      - Let \(v_i = V(i) + \rho_{q'}(i) * t_1(i) + \rho_q(i) * (t_2(i) - t_1(i))\) for each \(i\)
        - \(v_i\)'s satisfies the guards (different from \(V\))
      - \(V(i) = V(i) + \rho_{q'}(i) * t_1(i) + \rho_q(i) * (1 - t_1(i))\) for each \(i\)
    - \((q, V, q') = (\tau) > (q_1, V_1, q_1') \iff q_1 = q_1' = q, \text{only t1 delay}\)
Transition Relation Graphic representation
More Transition Relations

- With transition relation:
  - Runs can be constructed.
    - \( \sigma = (q_0, V_0, q'_0) \alpha_0(q_1, V_1, q'_1) \alpha_1 \ldots (q_k, V_k, q'_k) \)
  - Initial condition: \( (q_0, V_0, q'_0) \)
  - \( \sigma = (q_m, V_m, q'_m) \xrightarrow{\alpha_m} (q_{m+1}, V_{m+1}, q'_{m+1}), \quad 0 \leq m < k \)
  - State and act sequences:
    - \( st(\sigma) = q_0 q_1 \ldots q_m \ldots q_k \)
    - \( act(\sigma) = \alpha_0 \alpha_1 \ldots \alpha_m \ldots \alpha_k \)
  - Languages (set of runs):
    - \( L_{st}(A) = \{ st(\sigma) \} \)
    - \( L_{act}(A) = \{ act(\sigma) \} \)

- Claim: The languages are REGULAR subsets of all possible state and act sequences.
Generalizations

- Guards
  - Do not have to be rectangular (not simple)

- Rates of Evolution
  - Does not have to be unique in each control location. Instead can be rectangular.
Conclusion

- Many different approaches (models, languages, etc.) available to solve hybrid systems.
- However, most hybrid systems are undecidable, except for some special cases.
- Abstractions may be able to reduce this problem.