Logic and Hybrid Systems

Manasvi Saxena

Formal Systems Lab, UIUC

- ▶ Dynamical Systems exhibiting both discrete (jump) and continuous (flow) behaviors.
- ▶ Serve as models of physical systems, from thermostats to trains.
- ▶ Continuous dynamics specified using Differential Equations.

Differential Dynamic Logic (dL)

- Main focus - Differential Dynamic Logic for Hybrid Systems (Andre Platzer).

Differential Dynamic Logic (dL)

- ▶ Main focus - Differential Dynamic Logic for Hybrid Systems (Andre Platzer).
- ▶ Practical deductive verification of hybrid systems.

Differential Dynamic Logic (dL)

- Main focus - Differential Dynamic Logic for Hybrid Systems (Andre Platzer).
- Practical deductive verification of hybrid systems.
- Introduces Hybrid Program - program notation for hybrid systems.

Differential Dynamic Logic (dL)

- Main focus - Differential Dynamic Logic for Hybrid Systems (Andre Platzer).
- Practical deductive verification of hybrid systems.
- Introduces Hybrid Program - program notation for hybrid systems.
- Dynamic Logic for Hybrid Programs, a generalization of Dynamic Logic.

- Main focus - Differential Dynamic Logic for Hybrid Systems (Andre Platzer).
- Practical deductive verification of hybrid systems.
- Introduces Hybrid Program - program notation for hybrid systems.
- Dynamic Logic for Hybrid Programs, a generalization of Dynamic Logic.
- Suited for automation.

Hybrid Automata

- Commonly used to model Hybrid Systems, via Graphs.
- Nodes specify continuous dynamics. Edges describe discrete transitions.
- Intuitive, but not suitable for deductive verification.

Hybrid Automata

- Commonly used to model Hybrid Systems, via Graphs.
- Nodes specify continuous dynamics. Edges describe discrete transitions.
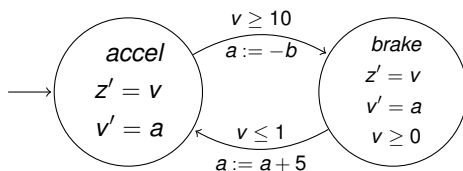- Intuitive, but not suitable for deductive verification.

Figure: Hybrid Automata (simplified) of a Train Control System

- **First Order Logic** - No builtin means for referring to state transitions.
- **Temporal Logics** - Modal operators allow referring to state transitions. But valid formulas only express generic facts.

- **First Order Logic** - No builtin means for referring to state transitions.
- **Temporal Logics** - Modal operators allow referring to state transitions. But valid formulas only express generic facts.
- **Dynamic Logic (DL)** - Combines operational system models with operators for reasoning.
    - Provides parameterized modal operators, $[\alpha]$, $\langle\alpha\rangle$ that refer to states reachable by system α.
    - $[\alpha]\phi$ expresses all states reachable by α satisfy ϕ, allowing reasoning about discrete systems.
    - Say $(b > 0) \rightarrow [a := -b](a < 0)$ expresses a discrete transition. Using DL's calculus, we get $(b > 0) \vdash (a < 0)[b/a]$. Convenient for reasoning about discrete behavior.

- **First Order Logic** - No builtin means for referring to state transitions.
- **Temporal Logics** - Modal operators allow referring to state transitions. But valid formulas only express generic facts.
- **Dynamic Logic (DL)** - Combines operational system models with operators for reasoning.
    - Provides parameterized modal operators, $[\alpha]$, $\langle\alpha\rangle$ that refer to states reachable by system α.
    - $[\alpha]\phi$ expresses all states reachable by α satisfy ϕ, allowing reasoning about discrete systems.
    - Say $(b > 0) \rightarrow [a := -b](a < 0)$ expresses a discrete transition. Using DL's calculus, we get $(b > 0) \vdash (a < 0)[b/a]$. Convenient for reasoning about discrete behavior.
    - No built in notion for describing or reasoning about continuous dynamics.

- ▶ Generalize DL so operational models α can be used in modal formulas like $[\alpha]\phi$. dL refers to generalized models as "Hybrid Programs".
- ▶ A calculus also allowing for reasoning about continuous dynamics.