Stanford Little Engines Course (Fall 2003; Lecture 20)

Little Engines of Proof

N. Shankar, L. de Moura, H. Ruess, A. Tiwari

shankar@csl.sri.com

URL: http://www.csl.sri.com/~shankar/LEP.html

Computer Science Laboratory

SRI International

Menlo Park, CA

Application II

We have studied decision procedures for

- various classes of formulas
- over different logical theories

There are two classes of applications

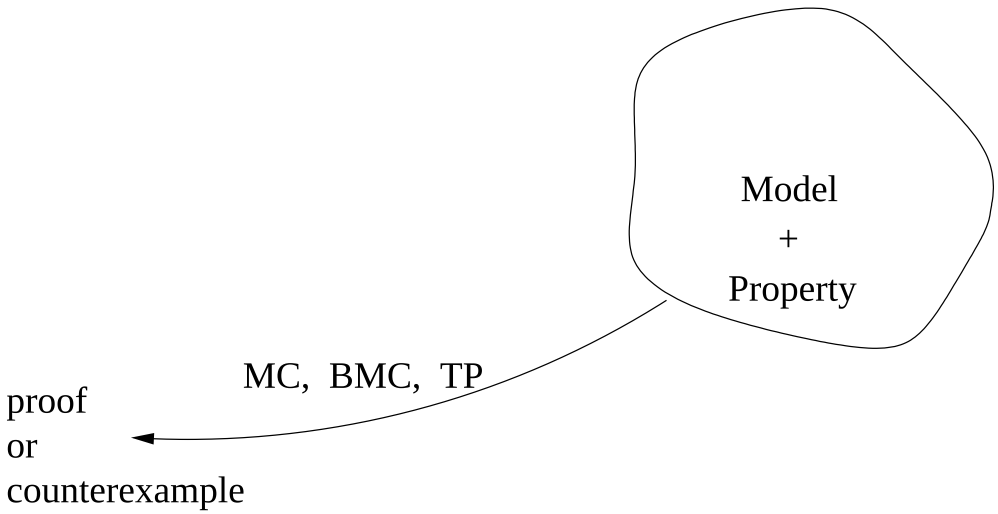
- Direct – theorem provers, constraint solvers, optimizers
- Embedded – compilers, type checkers, model checkers, test generation, parameter computation, diagnosis, model construction

We discuss application to verification

Verification

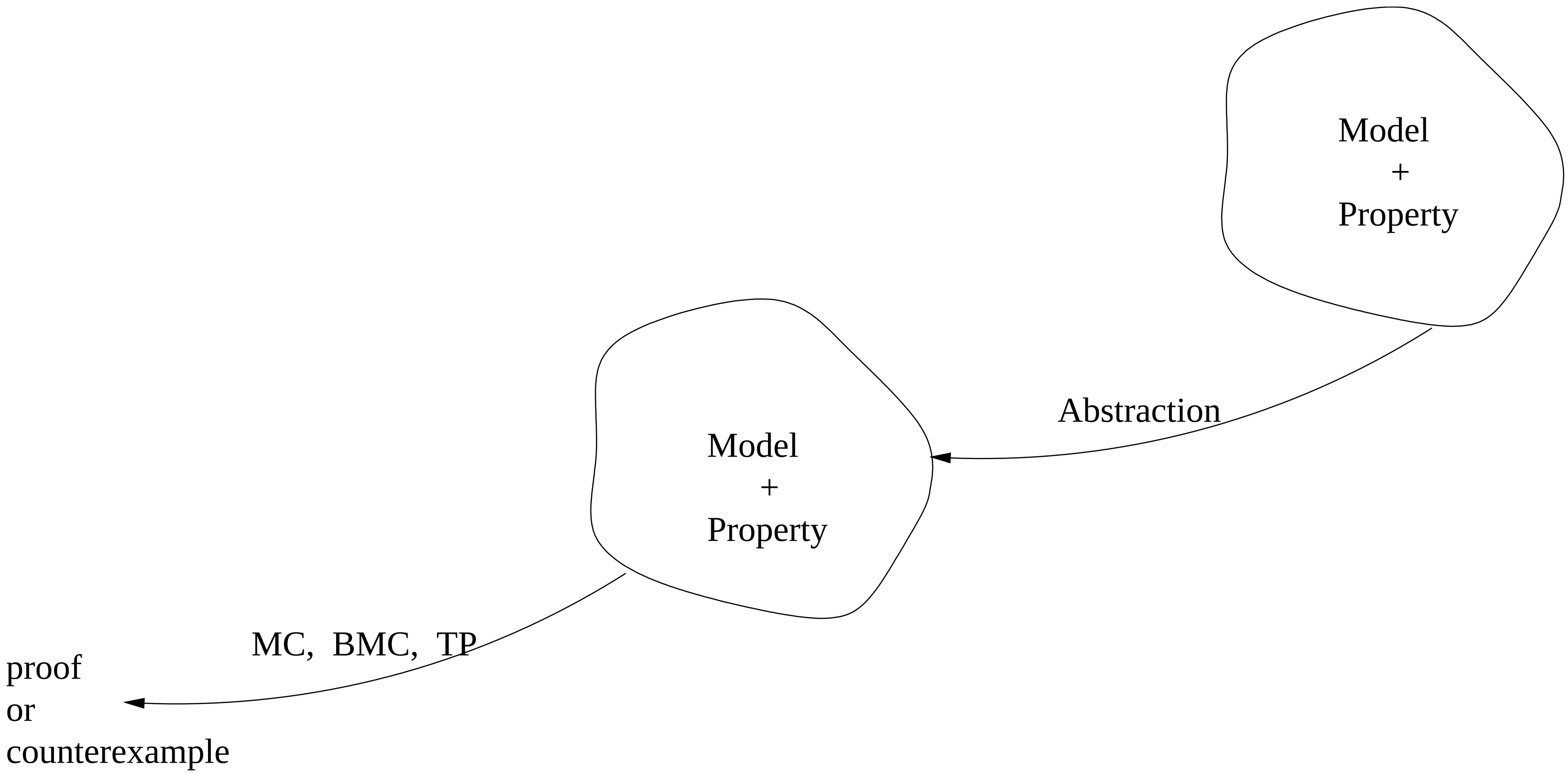
- Bounded model checking:
  - finite state systems: SAT
  - infinite state systems or systems with datatypes: lazy/eager theorem proving
- Abstraction
  - discrete transition systems
  - hybrid dynamical systems

Verification



We saw how theorem proving can be used in the process of model checking/ bounded model checking

Abstraction



Theorem proving and decision procedures also play a central role in creating simpler abstractions of complex initial models

Transition Systems

Transition system $M = (S, I, T)$

S: set of states.

valuation of state variables

$I \subseteq S$: set of initial states.

$T \subseteq S \times S$: transition relation.

Semantics $[\![M]\!]$. Collection of valid traces/paths

Trace. A sequence of states

$$s_0 \longrightarrow s_1 \longrightarrow s_2 \longrightarrow s_3 \longrightarrow s_4 \longrightarrow s_5 \longrightarrow s_6 \longrightarrow \ldots$$

s.t. $s_0 \in I$ and $(s_i, s_{i+1}) \in T$

Abstractions and Refinements

$M = (S, I, T)$, $\hat{M} = (\hat{S}, \hat{I}, \hat{T})$: Two transition systems

α: $S \mapsto \hat{S}$, α surjective

α defines an equivalence relation $\equiv$ on S: $s \equiv s'$ iff $\alpha(s) = \alpha(s')$

$\hat{M}$ is an abstraction of M (w.r.t the mapping α) if

- $\hat{s} \in \hat{I}$ if $\exists s \in S.\alpha(s) = \hat{s} \ \wedge \ s \in I$
- $(\hat{s}, \hat{s}') \in \hat{T}$ if $\exists s, s' \in S.\alpha(s) = \hat{s} \ \wedge \ \alpha(s') = \hat{s}' \ \wedge \ (s, s') \in T$

There are other notions of abstractions depending on the property.

View $\hat{M}$ as $M/\equiv$

Abstractions

If $\hat{M}$ is an abstraction of M (w.r.t α) then $[\![\hat{M}]\!] \supseteq \alpha([\![M]\!])$

Ex. Prove the above theorem.

If there is no path in $\hat{M}$ to a $\widehat{\text{bad}}$ state, then there is no path to a bad state in M

Approach to verifying safety properties of a transition system M:

- pick an abstract domain $\hat{S}$
- choose an abstraction mapping α
- construct an abstract system $\hat{M}$ (w.r.t α)
- verify the (mapped) property on $\hat{M}$
- if previous step fails, refine the mapping α

Constructing Abstractions

Elimination method: Requires theorem proving support

- $\hat{s} \in \hat{I}$ if $\exists s \in S . \alpha(s) = \hat{s} \ \wedge \ s \in I$
- $(\hat{s}, \hat{s'}) \in \hat{T}$ if $\exists s, s' \in S . \alpha(s) = \hat{s} \ \wedge \ \alpha(s') = \hat{s'} \ \wedge \ (s, s') \in T$

Theory: depends on the language used to specify I, T, $\hat{S}$, and α

Class of formulas: if I, T, α are specified using QF formulas, then we only need satisfiability of QF formulas

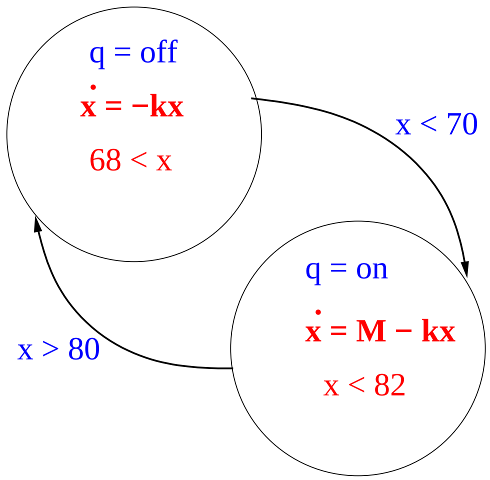
Works even when the prover is incomplete

Hybrid Systems

Several real-world systems are best modeled as a combination of

- discrete transition systems and
- continuous dynamical systems (differential equations)

Example. A thermostat.



Hybrid Automata

Formal model of a hybrid system is a hybrid automaton:

A tuple $(Q, X, \mathbf{S}_0, F, Inv, R)$:

- Q: finite set of discrete variables
- X: finite set of continuous variables
- $\mathbf{X} = \mathbb{R}^{|X|}$, $\mathbf{Q}$ = set of all valuations for Q
- $\mathbf{S} = \mathbf{Q} \times \mathbf{X}$
- $\mathbf{S}_0 \subseteq \mathbf{S}$ is the set of initial states
- $F : \mathbf{Q} \mapsto (\mathbf{X} \mapsto \mathbb{R}^{|X|})$ specifies the rate of flow, $\dot{x} = F(q)(x)$
- $Inv : \mathbf{Q} \mapsto 2^{\mathbb{R}^{|X|}}$ gives the invariant set
- $R \subseteq \mathbf{Q} \times 2^{\mathbf{X}} \mapsto \mathbf{Q} \times 2^{\mathbf{X}}$ captures discontinuous state changes

Semantics of Hybrid Systems

s1	s2	s3	s4	s5	s6	s6
s1	s2	s3	s4	s5	s6	s7

- $s1 \in \mathbf{S}_0$ is an initial state
- Discrete Evolution: $s_i \longrightarrow s_{i+1}$ iff $R(s_i, s_{i+1})$
- Continuous Evolution: $s_i = (l, x_i) \longrightarrow s_{i+1} = (l, x_{i+1})$ iff there exists a $f : \mathbb{R}^{|X|} \mapsto \mathbb{R}^{|X|}$ and $\delta > 0$ such that

$$\begin{array}{rclcrcl} x_{i+1} & = & f(\delta) & & x_i & = & f(0) \\ \dot{f} & = & F(l) & & f(t) & \in & Inv(l) \; for \; 0 \leq t \leq \delta \end{array}$$

Semantics Example

A possible trace for the thermostat

$$(q = \mathit{off}, x = 75) \rightarrow (q = \mathit{off}, x = 70) \rightarrow (q = \mathit{off}, x = 69) \rightarrow$$
$$(q = on, x = 69) \rightarrow (q = on, x = 75) \rightarrow (q = on, x = 81) \rightarrow$$
$$(q = \mathit{off}, x = 81) \rightarrow \ldots$$

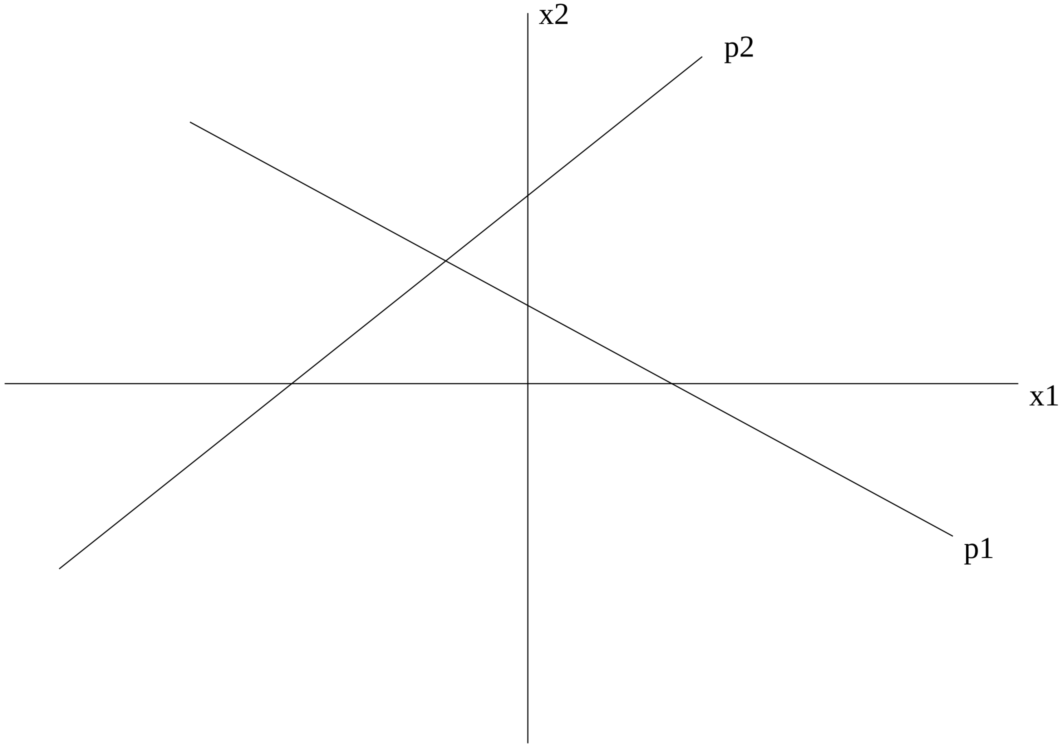
Qualitative Abstraction

Abstracting the hybrid automaton:

- Abstract domain: $\mathbf{Q} \times \{pos, neg, zero\}^m$
- Mapping: α is given by choosing m polynomials from $\mathbb{Q}[x_1, \ldots, x_n]$
  s.t. $\alpha((q, v_1, \ldots, v_n)) = (q, sign(p_1(\vec{v})), \ldots, sign(p_m(\vec{v})))$
- Abstract the initialization states: $\surd$
- Abstract the discrete transitions: $\surd$
- Abstract the continuous flow: How?

Abstraction Algorithm: 1

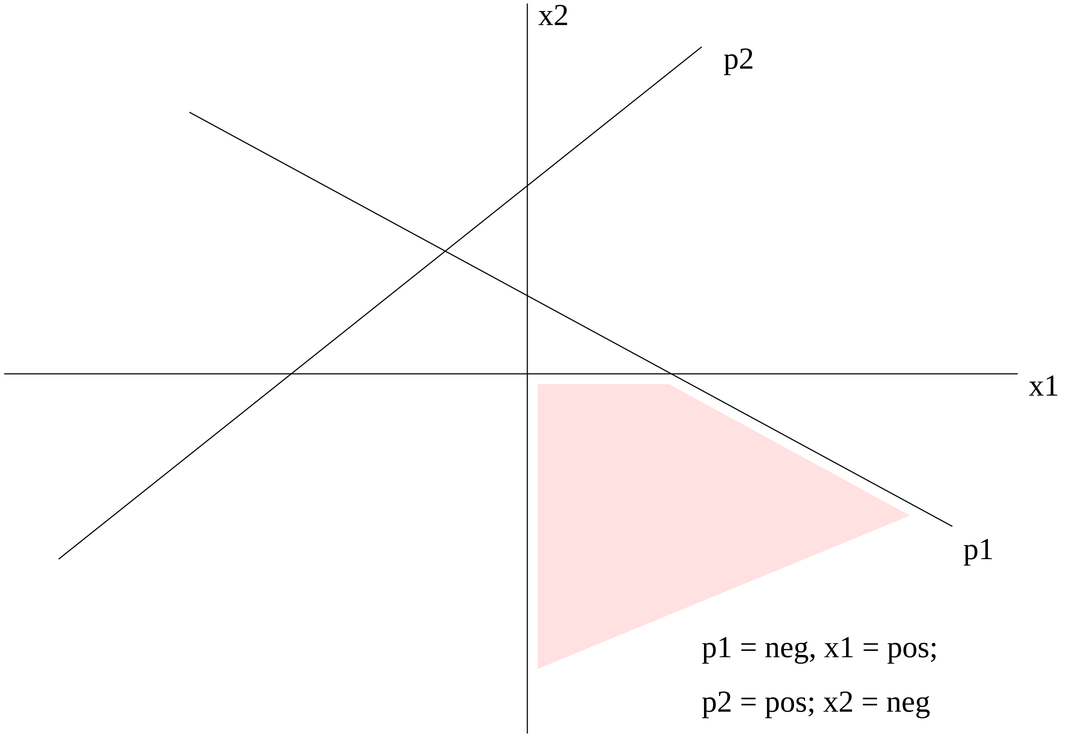
Consider a continuous dynamical system with two state variables. Concrete state space: $\mathbb{R}^2$



Partitioned w.r.t signs of four linear forms x_1, x_2, p_1, and p_2.

Abstraction Algorithm: 2

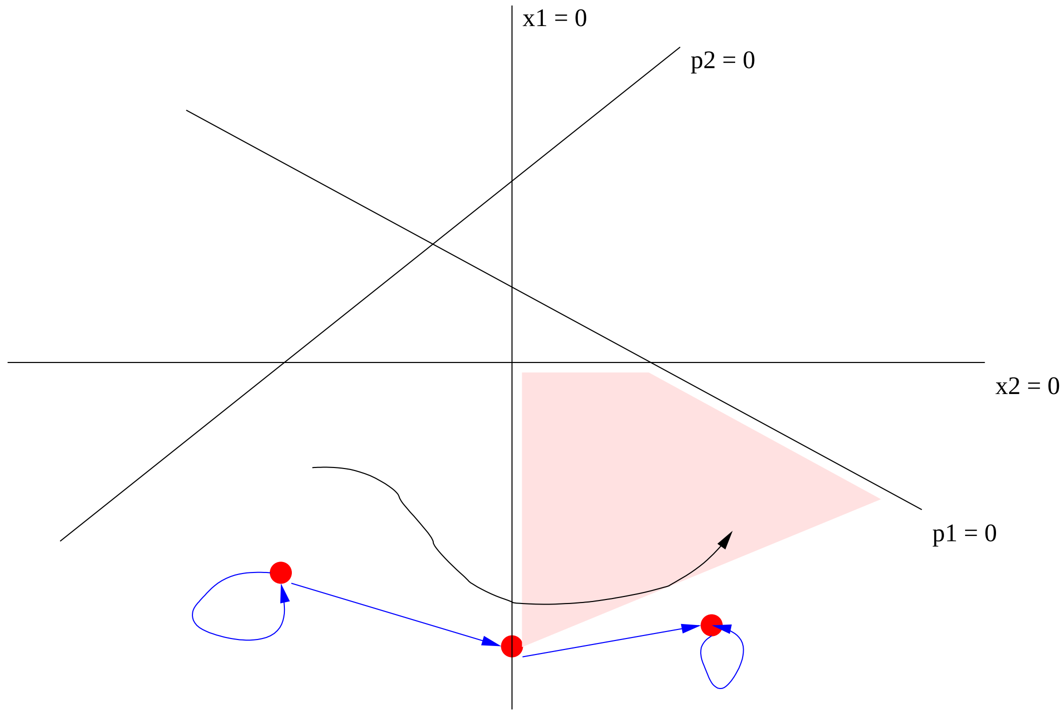
Abstract states correspond to sets of concrete states.



Total number of abstract states = $3^4 = 81$, but feasible abstract states = $11 + 16 + 6 = 33$

Abstraction Algorithm: 3

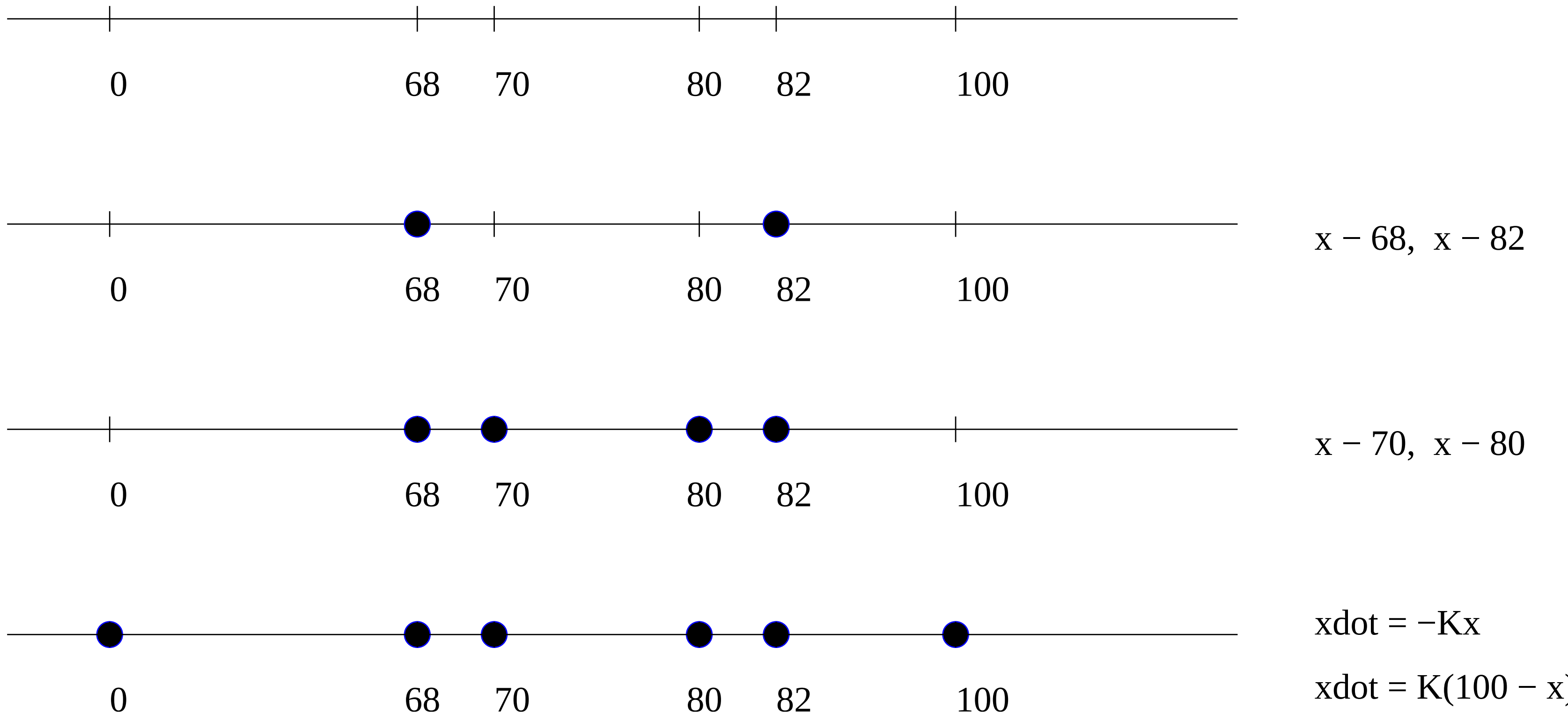
Abstract transitions overapproximate concrete transitions.



How to abstract the continuous transitions?.

Abstraction Mapping: Example

In the thermostat example:



Abstracting the Continuous Dynamics

Concrete state space : $\mathbf{Q} \times \mathbb{R}^n$

Abstract state space : $\mathbf{Q} \times 3^m$

Question: Fix a mode. Given

1. an abstract state $f_1 \begin{smallmatrix} > \\ < \\ = \end{smallmatrix} 0,\ f_2 \begin{smallmatrix} > \\ < \\ = \end{smallmatrix} 0,\ \ldots,\ f_m \begin{smallmatrix} > \\ < \\ = \end{smallmatrix} 0$

2. mode dynamics, $\dot{x_1} = g_1,\ \ldots,\ \dot{x_n} = g_n$

determine all new abstract states $f_1 \mathbin{?} 0,\ \ f_2 \mathbin{?} 0,\ \ \ldots,\ \ f_m \mathbin{?} 0$ reachable from the given abstract state.

Abstracting Cont. Dynamics: the dual question

What is the sign of f_i in the next state?

Our approach is based on qualitative reasoning. If

$$f_1 \begin{array}{c}>\\<\\=\end{array} 0 \wedge f_2 \begin{array}{c}>\\<\\=\end{array} 0 \wedge \cdots \wedge f_m \begin{array}{c}>\\<\\=\end{array} 0 \wedge \text{ state-invariant}$$

$$\Rightarrow \qquad \dot{f}_i > 0$$

then, sign of f_i in the next state is

- $\{pos\}$ if $f_i > 0$ now,
- $\{neg, zero\}$ if $f_i < 0$ now,
- $\{pos\}$ if $f_i = 0$ now.

Decision Procedure

We need a decision procedure to prove

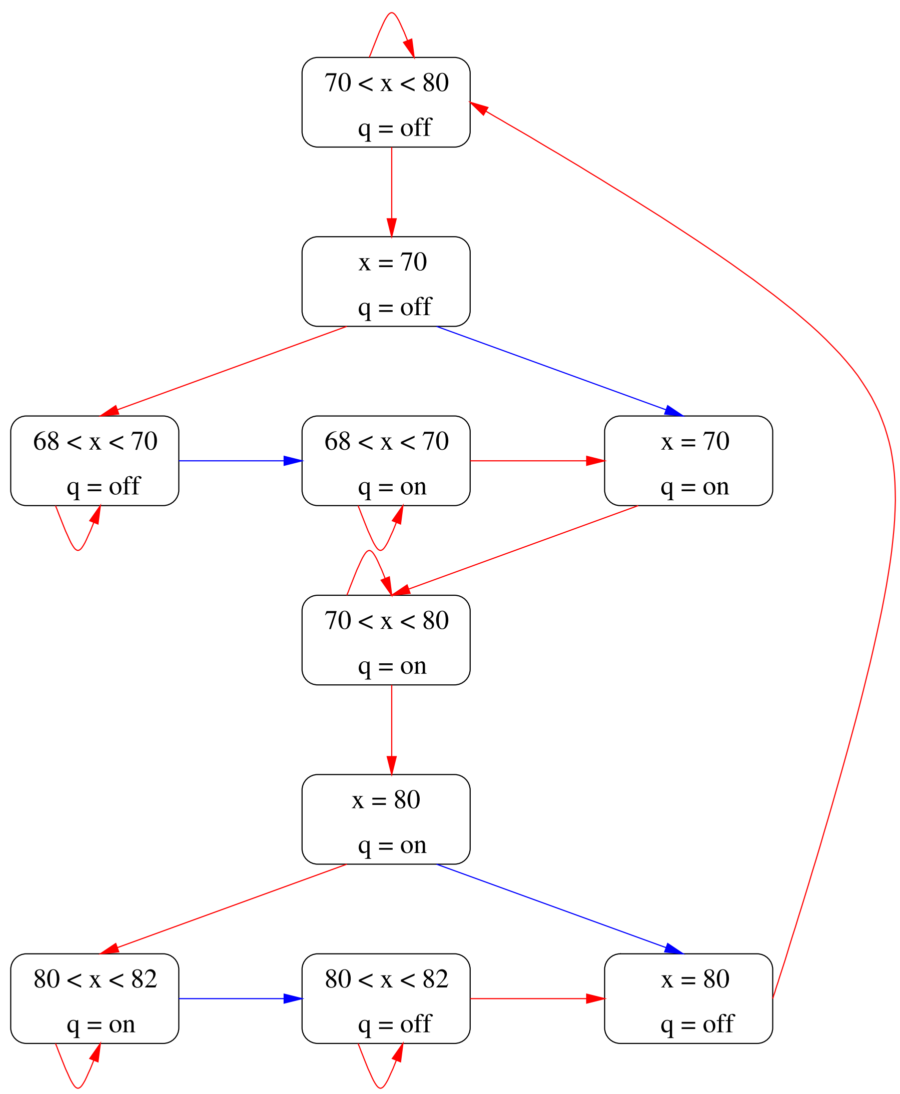
$$f_1 \ \substack{>\\<\\=} \ 0 \wedge f_2 \ \substack{>\\<\\=} \ 0 \wedge \cdots \wedge f_m \ \substack{>\\<\\=} \ 0 \wedge \text{state-invariant} \ \Rightarrow \ \dot{f}_i > 0$$

If f_i's, $\dot{f}_i$ are polynomials, and state-invariant also only consists of polynomials, then we can use a decision procedure for the QF-theory of reals.

Failure-tolerant Theorem Proving: sound, but incomplete, procedure suffices.

Implementation optimization: (i) Do a clever 3^m enumeration; (ii) Use witness generation capability of decision procedure.

Abstract Thermostat System



Hybrid Models

Hybrid automata is a powerful modeling formalism

Recently it has been used to create "simpler" models of processes traditionally viewed as continuous dynamical systems

Prominent example is genetic regulatory networks

Where discrete transitions model transcription regulation

And continuous transitions model metabolism processes

Qualitative abstraction, and hence decision procedures and theorem proving, can be used to analyze biological processes