ARTS/DExVal
Derivation of Meaningful Experiments for Validation

- Prof. A. Haeberer, PUC-Rio
- Prof. M. Wirsing, LMU Munich
- Dr. A. Ciarlini, PUC-Rio
- Dr. T. Fruehwirth, LMU Munich
ARTS

• Formal basis for software development, funded partially by Siemens, Brazil
Validation and Testing

- Critical
- Expensive
- Revealing maximum number of bugs
- Meaningful experiments
Model Checking

- Verification of properties
- Modal temporal logic
- Prop. holds or there is a counterexample
- Approximation
  - Infinite state machines $\rightarrow$ Finite state machines
  - Continuous variables $\rightarrow$ Discrete variables
  - State explosion
The Goal

- Verification and derivation of properties of concurrent transition systems
- Continuous variables and non-linear expressions
- Expressiveness: variables at different times
The Approach

• Symbolical execution
• Constraint Logic Programming
• User descriptions → all paths and corresponding derived properties
• E.g. Constraints on output → constraints on input
Hybrid Automata

- Continuous activities
- Discrete transitions
- Components
  - Variables
  - States: name, invariant and iteration
  - Transitions: source and target states, guarded actions, events
Hybrid System

• Timed hybrid automata
  – Synchronization: machine clock
  – Modifications according to last state
• Coordination: sharing of variables and events
• Simultaneous modifications
• Variable modified by only one automaton
Constraint Logic Programming

• Logic programming
  – Declarative rules defining relations
  – Search for all solutions using backtracking
  – Non-deterministic

• Constraint solving
  – Efficient algorithms
  – Solving sets of distinguished relations
  – Deterministic
Constraint Logic Programming

- LP + CS:
  - Expressiveness and efficiency
  - LP sends constraints to CS
  - Constraints solved in parallel
  - Inconsistency $\rightarrow$ cut branch
  - Ex:
    - $X+Y<5$ and $Y>0$
    - $X=6 \rightarrow$ fail
DExVal Tool

• Input:
  – Automata
  – Initial and final states (not mandatory)
  – Properties: Values or ranges (input, intermediate and output)

• Output: Paths and corresponding constraints relating selected variables

• Using output for testing
  \[
  \text{OUT} > 100 \rightarrow 10 < \text{IN} < 20 \\
  \text{OUT} \leq 100 \rightarrow (\text{IN} \leq 10) \lor (\text{IN} \geq 20)
  \]
  Better testing \( \text{IN} = 1, 10, 15, 20, 30 \) than \( \text{IN} = 12, 13, 14, 15, 16 \)
Examples of Properties

• Since $X > Y$, $Z = 1$

• For all states, $X$ has a higher value than its value in the previous state

• If, at some time, $X > Y$, then at most 5 clocks later $Z = 1$

• Obs: Existential and universal quantification
Implementation

- Preparation stage
  - Data structure → variables’ history
  - Translation of descriptions into constraints
- Symbolic execution
  - search for paths
  - addition of new constraints corresponding to invariants, iterations and transitions
Implementation

• Production of answers
  – Projection on selected variables
  – Printing
    • States at each clock
    • Remaining constraints resulting from execution and projection
Boiler Example

P - pump (automatic control - automaton Pump)
P=0 (off), P=1 (on)

MAX - maximum water volume (constant)

V - water volume (simulated by the automaton Boiler)

S - shower (external variable)
S=0 (off), S=1 (on)

G - granularity - controls the speed with which the water volume and temperature vary (constant)

HIGH - high temperature (constant)

LOW - low temperature (constant)

T - temperature (simulated by the automaton Boiler)

H - heater (automatic control - automaton Heater)
H=0 (off), H=1 (on)

WT - cold water temperature at the pump (constant)
Boiler Automaton

**normal_heater_on**
- **Inv:** $V \geq G \land H=1$
- **Iter:** $V := V + (P-S) \cdot G \land T := (V \cdot T + P \cdot G \cdot WT) / (V + P \cdot G) + G$

**normal_heater_off**
- **Inv:** $V \geq G \land H=0 \land T \geq WT$
- **Iter:** $V := V + (P-S) \cdot G \land T := (V \cdot T + P \cdot G \cdot WT) / (V + P \cdot G) - G \cdot 0.5$

**empty**
- **Inv:** $V < G \land T = WT$
- **Iter:** $V := V + (P-S) \cdot G$

Transitions:
- $V \geq G \land H=1 \rightarrow T := WT$
- $V \geq G \land H=1 \land T < WT \rightarrow T := WT$
- $V < G \rightarrow T := WT$
- $T < WT \land H=0 \land V \geq G \rightarrow T := WT$
Initial temperature for taking a shower without turning on the heater

INPUT:
CONSTRAINTS: heater:1=0, pump:1=1, water_volume:1=10.0,
              shower:1=1,
              all(X,shower:X=1), all(X,heater:X=0)
INITIAL STATES: pump_on, heater_maintain,
                boiler_normal_heater_off
FINAL STATES: (not specified)
CLOCKS: 5
PROJECT: temperature:1 (i.e. initial temperature)

OUTPUT:
<table>
<thead>
<tr>
<th>Clock</th>
<th>Pump</th>
<th>Heater</th>
<th>Boiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>on</td>
<td>maintain</td>
<td>normal_heater_off</td>
</tr>
<tr>
<td>2</td>
<td>on</td>
<td>maintain</td>
<td>normal_heater_off</td>
</tr>
<tr>
<td>3</td>
<td>on</td>
<td>maintain</td>
<td>normal_heater_off</td>
</tr>
<tr>
<td>4</td>
<td>on</td>
<td>maintain</td>
<td>normal_heater_off</td>
</tr>
<tr>
<td>5</td>
<td>on</td>
<td>maintain</td>
<td>normal_heater_off</td>
</tr>
</tbody>
</table>

temperature:1 > 47.18
Behaviour of the shower for the continuous increase of the water level

INPUT:
CONSTRAINTS: heater:1=0, pump:1=1, temperature:1=30.0, water_volume:1=6.0, all(X, water_volume:(X+1)>water_volume:X) (increase water)
INITIAL STATES: pump_on, heater_maintain, boiler_normal_heater_off
FINAL STATES: (not specified)
CLOCKS: 5
PROJECT: shower:X, water_volume:X (i.e. at all clocks)

OUTPUT:

<table>
<thead>
<tr>
<th>Clock</th>
<th>Pump</th>
<th>Heater</th>
<th>Boiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>on</td>
<td>maintain</td>
<td>normal_heater_off</td>
</tr>
<tr>
<td>2</td>
<td>on</td>
<td>turning_on</td>
<td>normal_heater_on</td>
</tr>
<tr>
<td>3</td>
<td>on</td>
<td>maintain</td>
<td>normal_heater_on</td>
</tr>
<tr>
<td>4</td>
<td>on</td>
<td>maintain</td>
<td>normal_heater_on</td>
</tr>
<tr>
<td>5</td>
<td>on</td>
<td>maintain</td>
<td>normal_heater_on</td>
</tr>
</tbody>
</table>

shower: [1..4] = 0, shower: 5 = Var,
water_volume: 1 = 6.0, water_volume: 2 = 8.0, water_volume: 3 = 10.0,
water_volume: 4 = 12.0, water_volume: 5 = 14.0
Summary

• We are concerned with validation and testing
• Meaningful experiments
• Derivation of properties
• Symbolic execution
• DExVal tool based on CLP
Future work

• Integration with ARTS’ graphical interface
• Tailoring the behaviour of the constraint solver:
  – Non-linear constraints
  – Non-determinism: disjunction and existential quantification
• Meaningful experiments:
  – Methodology
  – Real applications