# Constraints in Abstract Model Checking

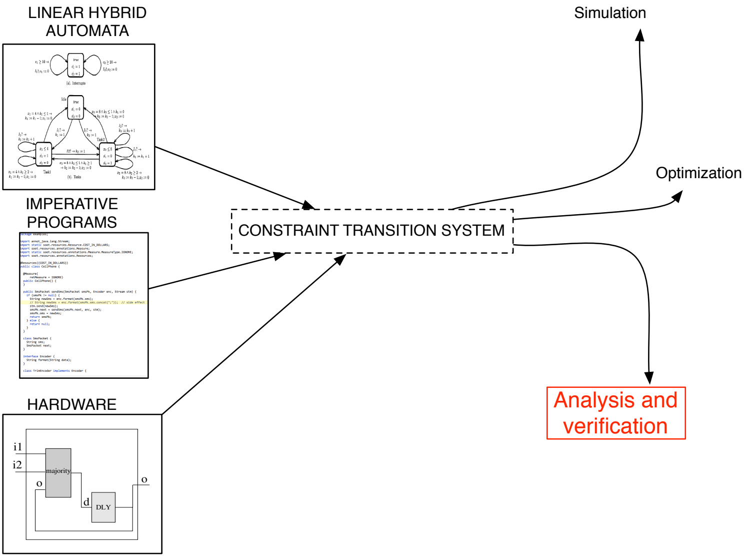
## Direct implementation of an abstract interpretation

John Gallagher[1,2]

[1]Roskilde University [2]IMDEA Software Institute, Madrid

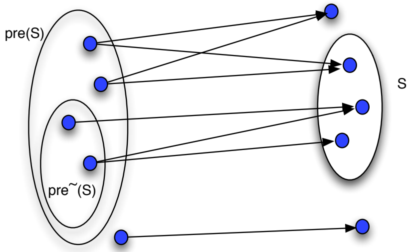
CP meets CAV
Turunç, Turkey

# Encoding operational semantics



# $pre$ and $\widetilde{pre}$ functions

From a transition relation, compute functions $pre : 2^S \rightarrow 2^S$, $\widetilde{pre} : 2^S \rightarrow 2^S$.




- $pre(Z)$: the set of possible predecessors of set of states $Z$.
- $\widetilde{pre}(Z)$: the set of definite predecessors of set of states $Z$.

# $pre$ and $\widetilde{pre}$ expressed using constraint operations

A constraint $c(\bar{X})$ stands for the set of states satisfying $c(\bar{X})$.

$$pre(c'(\bar{y})) = \bigvee\{\exists\bar{y}(c'(\bar{y}) \wedge c(\bar{x},\bar{y})) \mid \bar{x} \xrightarrow{c(\bar{x},\bar{y})} \bar{y} \text{ is a transition}\}$$
$$\widetilde{pre}(c'(\bar{y})) = \neg(pre(\neg c'(\bar{y})))$$

We assume that the constraint solver has a projection ($\exists$-elimination) operation and is closed under boolean operations.

# Checking CTL properties

Define a function $[\![\phi]\!]$ returning the set of states where $\phi$ holds.
Compositional definition:

$$
\begin{array}{ll}
[\![p]\!] & = \text{states}(p) \\
[\![EF\phi]\!] & = \text{lfp}.\lambda Z.([\![\phi]\!] \cup pre(Z)) \\
[\![AG\phi]\!] & = \text{gfp}.\lambda Z.([\![\phi]\!] \cap \widetilde{pre}(Z)) \\
\ldots &
\end{array}
$$

where states($p$) is the set of states where proposition $p$ holds (i.e. a constraint).

Model checking $\phi$:

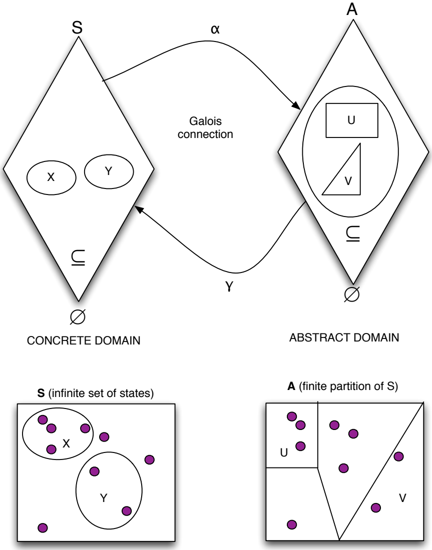
1. Evaluate $[\![\phi]\!]$.
2. Check that $I \subseteq [\![\phi]\!]$, where $I$ is the set of initial states.

Equivalently, check that $I \cap [\![\neg\phi]\!] = \emptyset$.

# Abstract model checking

- When the set of states is infinite, $[\![\phi]\!]$ cannot usually be evaluated
- Use abstract interpretation to define an abstract function $[\![\phi]\!]^a$ over some abstract domain.
- As an example, consider an abstract domain constructed from a *finite partition* of the set of states.

# Galois connection



# Galois connection implemented using constraint operations

Assume that the elements of the partition are given by constraints. Let $c_d$ be the constraint defining the partition element $d$.

$$\alpha(c) = \{d \in A \mid \mathrm{SAT}(c_d \wedge c)\}$$
$$\gamma(V) = \bigvee\{c_d \mid d \in V\}$$

- *SAT* can be implemented by an SMT solver. We used Yices (http://yices.csl.sri.com/) interfaced to Prolog.

# Abstraction of functions

Given a function

$$f : 2^S \to 2^S$$

on the concrete domain, the most precise approximation of $f$ in the abstract domain is

$$\alpha \circ f \circ \gamma : 2^A \to 2^A.$$

# Abstract checking of CTL properties

Applying this construction to the function $[\![.]\!]$, obtain a function $[\![\phi]\!]^a$.

$$\begin{array}{ll} [\![p]\!]^a & = (\alpha \circ \mathsf{states})(p) \\ [\![EF\phi]\!]^a & = \mathrm{lfp}.\lambda Z.([\![\phi]\!]^a \cup (\alpha \circ \mathit{pre} \circ \gamma)(Z)) \\ [\![AG\phi]\!]^a & = \mathrm{gfp}.\lambda Z.([\![\phi]\!]^a \cap (\alpha \circ \widetilde{\mathit{pre}} \circ \gamma)(Z)) \\ \ldots & \end{array}$$

Computation of $[\![\phi]\!]^a$ terminates. It can be shown that for all $\phi$,

$$[\![\phi]\!] \subseteq \gamma([\![\phi]\!]^a)$$

.

Abstract Model Checking of $\phi$

1. Compute $[\![\neg\phi]\!]^a$.
2. Check that $I \cap \gamma([\![\neg\phi]\!]^a) = \emptyset$.
3. This implies that $I \cap [\![\neg\phi]\!] = \emptyset$, since $\gamma([\![\neg\phi]\!]^a) \supseteq [\![\neg\phi]\!]$.

# Some Experiments on Linear Hybrid Automata

Arbitrary CTL formulas can be checked (not just A-formulas as in standard abstract model checking).

| *System* | *Property* | $A$ | $\Delta$ | *secs.* |
|---|---|---|---|---|
| Water Monitor | $AF(W \geq 10)$ | 5 | 4 | 0.02 |
| | $AG(0 \leq W \wedge W \leq 12)$ | 5 | 4 | 0.01 |
| | $AF(AG(1 \leq W \wedge W \leq 12))$ | 5 | 4 | 0.02 |
| | $AG(W = 10 \rightarrow AF(W < 10 \vee W > 10))$ | 10 | 4 | 0.05 |
| | $AG(AG(AG(AG(AG(0 \leq W \wedge W \leq 12)))))$ | 5 | 4 | 0.02 |
| | $EF(W = 10)$ | 10 | 4 | 0.01 |
| | $EU(W < 12, AU(W < 12, W \geq 12))$ | 7 | 4 | 0.04 |
| Task Sched. | $EF(K2 = 1)$ | 18 | 12 | 0.53 |
| | $AG(K2 > 0 \rightarrow AF(K2 = 0))$ | 18 | 12 | 0.30 |
| | $AG(K2 \leq 1)$ | 18 | 12 | 0.04 |

# Conclusions

- Direct abstraction framework, based on Galois connections
- Abstract semantics parameterised by Galois connection, not tied to any particular kind of abstraction
- No need for (dual) abstract transition systems
- Not limited to reachability properties
- For constraint-based domains, direct implementation using constraint solvers and satisfiability checkers.

- Future Research: mainly on refinement (e.g. CEGAR, or Ganty's scheme).
  - This is a huge search problem in itself!
- Other abstractions than partitions

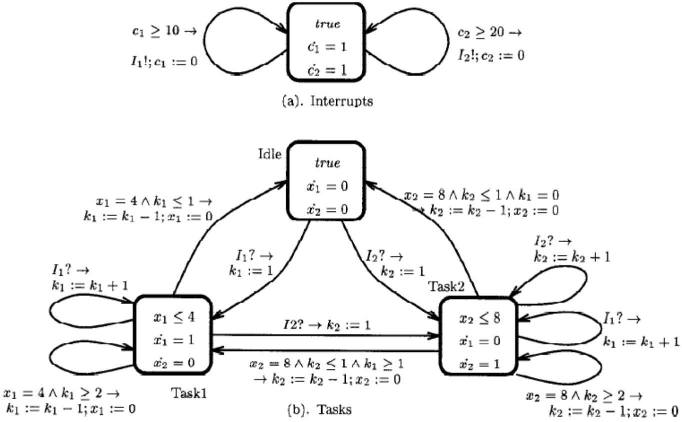
# CLP program encoding reachable states

| | | |
|---|---|---|
| transition(X,X') | $\leftarrow$ | $c_1$(X,X'). |
| transition(X,X') | $\leftarrow$ | $c_2$(X,X'). |
| $\ldots$ | $\leftarrow$ | $\ldots$ |
| initState(X) | $\leftarrow$ | $c_{init}$(X). |
| reach(X) | $\leftarrow$ | initState(X). |
| reach(X') | $\leftarrow$ | reach(X), transition(X,X'). |

# Sample Scheduler CTL Properties

- Liveness property (nested CTL property): $AG(K2 > 0 \rightarrow AF(K2 = 0))$. (A waiting high priority task is eventually scheduled).
- Existential liveness property: $EF(K2 = 1)$. (A high priority task can arise).
- Safety property: $AG(K2 \leq 1)$. (No more than one high priority task can be waiting).

# Example: A task scheduler [Halbwachs et al. 94]



# Transition System for Scheduler

*Sample transition of Scheduler.*

```
transition((J, L, N, P, R, S, G),(A, B, C, D, E, F, 0)) :-
    G<H,
    1*I=1*J+1*(H-G),
    1*K=1*L+1*(H-G),
    1*M=1*N+0*(H-G),
    1*O=1*P+0*(H-G),
    1*Q=1*R+0*(H-G),
    1*_=1*S+0*(H-G),
    K>=20, A=I, B=0,
    C=M, D=O, E=Q,
    F=1.
```