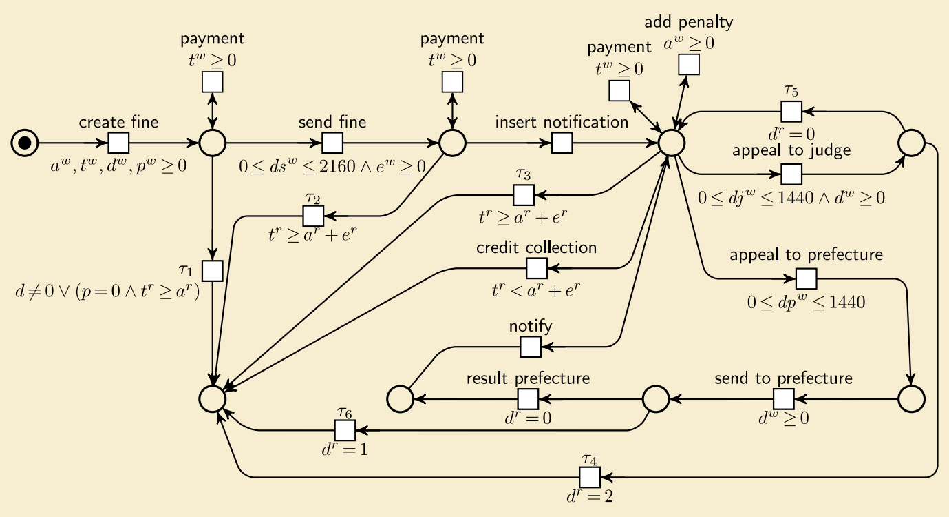
ada.

Automatic Verification of Data–Aware Processes with Arithmetic

Paolo Felli,[1] Marco Montali,[2] Sarah Winkler[2]

[1]Università di Bologna, Italy
[2]Free University of Bozen-Bolzano, Italy

Data Petri net
with arithmetic conditions
as expressive process models



questions

- **compliance:**
  is given LTLf property satisfiable?
  is given CTLf* property satisfiable?
- **soundness:**
  is the process data–aware sound?
- **anticipatory monitoring:**
  given LTLf property and trace,
  what is its monitoring state?
- strategy synthesis

undecidable

finite summary property

- abstract decidability criterion
- expresses that reachable states can be faithfully abstacted by finitely many state formulas

Concrete decidability criteria: instances of finite summary

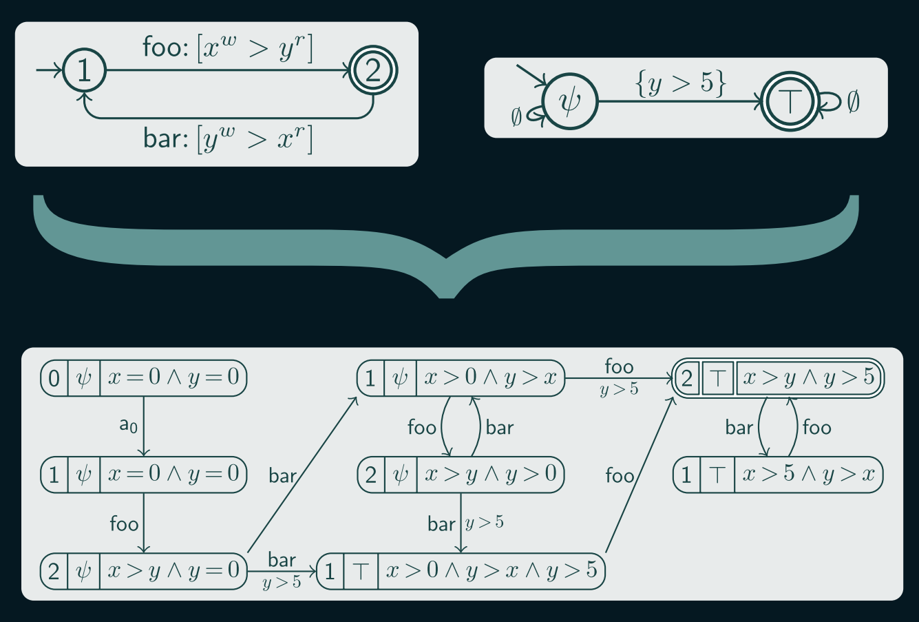
- **monotonicity constraints:**
  all constraints are variable–to–variable or variable–constant comparisons over ℝ
- **integer periodicity constraints:**
  restricted variable–to–variable/constant comparisons over ℝ (Demri 2007)
- **bounded lookback:**
  control–flow condition, generaliztion of feedback freedom (Vianu et al 2012)
- **combinations**
  sequential & variable–based decomposition

LTLf verification

Approach

- **data–aware dynamic systems (DDSs)**
  as simplified representation of process
- NFA for LTLf property ψ with constraints
- **product construction**
  – combine DDS and NFA states
  – represent verification states as SMT formulas
  $$update(\varphi, a) = \exists \overline{V}'.(\varphi(\overline{V}') \wedge guard_a(V', V) \wedge \bigwedge_{v \notin write(a)} v = v')$$
  – from final states can extract witness
- **decision procedure**
  product construction is finite for finite summary

product construction


foo: $[x^w > y^r]$
bar: $[y^w > x^r]$
1
2
ψ
$\{y > 5\}$
$\top$
$\emptyset$
$\emptyset$
$0 \mid \psi \mid x = 0 \wedge y = 0$
a_0
$1 \mid \psi \mid x = 0 \wedge y = 0$
foo
$2 \mid \psi \mid x > y \wedge y = 0$
bar
bar
$y > 5$
$1 \mid \psi \mid x > 0 \wedge y > x$
foo
bar
$2 \mid \psi \mid x > y \wedge y > 0$
bar
$y > 5$
$1 \mid \top \mid x > 0 \wedge y > x \wedge y > 5$
foo
$y > 5$
$2 \mid \top \mid x > y \wedge y > 5$
foo
bar
foo
$1 \mid \top \mid x > 5 \wedge y > x$

Implementation

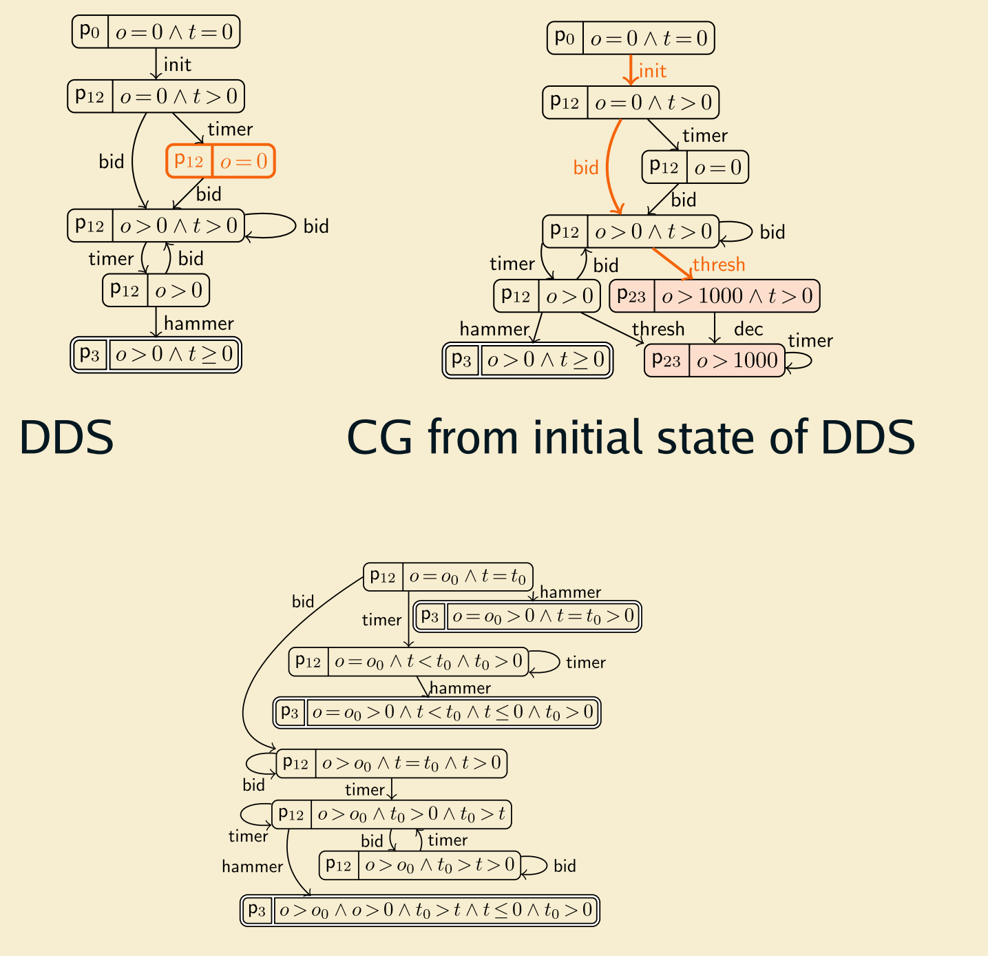
- Python tool available as online service
- supports LTLf and CTLf* model checking, soundness checking and monitoring
- performs verification and returns witesses/ counterexamples

https://ltl.adatool.dev

soundness checking

Approach

- **constraint graph (CG)**
  – faithful abstraction of state space
  – represent reachable states as SMT formulas
  $$update(\varphi, a) = \exists \overline{V}'.(\varphi(\overline{V}') \wedge guard_a(V', V) \wedge \bigwedge_{v \notin write(a)} v = v')$$
- check **data–aware soundness** by checking
  – presence of all transitions in CG
  – no "left over token" states in CG
  – for every state s in CG: build CG_s of all states reachable from s, check if final reachable
- can produce **counterexamples** to soundness
- **decision procedure** for finite summary DDS



DDS

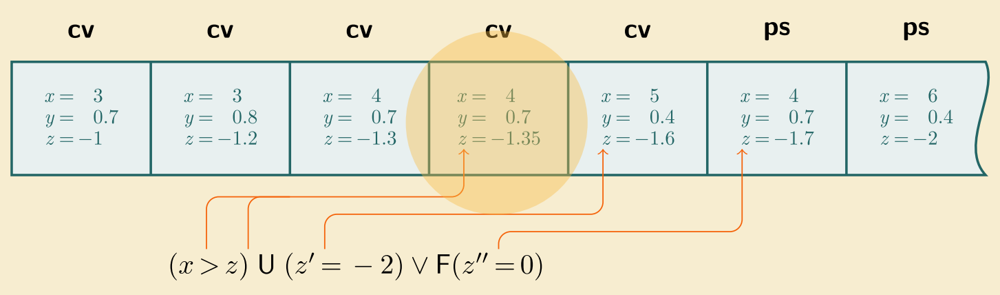
CG from initial state of DDS

CG from state p_{12}

anticipatory monitoring

Task

- consider LTLf with linear arithmetic constraints and allow **lookahead** on variables
- **monitoring task:** given trace and LTLf property, determine one of four monitoring states:
  currently satisfied   permanently satisfied
  currently violated   permanently violated


cv
cv
cv
cv
cv
ps
ps
$x = 3$ $y = 0.7$ $z = -1$
$x = 3$ $y = 0.8$ $z = -1.2$
$x = 4$ $y = 0.7$ $z = -1.3$
$x = 4$ $y = 0.7$ $z = -1.35$
$x = 5$ $y = 0.4$ $z = -1.6$
$x = 4$ $y = 0.7$ $z = -1.7$
$x = 6$ $y = 0.4$ $z = -2$
$(x > z)\ \mathsf{U}\ (z' = -2) \vee \mathsf{F}(z'' = 0)$

Results

- monitoring of properties without lookahead is decidable: DFA for property is monitor
- monitoring with lookahead is undecidable but decidable for finite summary properties
- properties with lookahead: monitor is given by
  – DFA plus
  – formulas obtained from CG that express whether final state is (still) reachable

P. Felli, M. Montali, S. Winkler: Linear–Time Verification of Data–Aware Dynamic Systems with Arithmetic. Proc. 36th AAAI, 2022.

P. Felli, M. Montali, S. Winkler: Soundness of Data–Aware Processes with Arithmetic Conditions. Proc. 34th CAiSE, LNCS 13295, 2022.

P. Felli, M. Montali, S. Winkler: CTL* model checking for data–aware dynamic systems with arithmetic. Proc. 11th IJCAR, LNCS 13385, 2022.

P. Felli, M. Montali, F. Patrizi, S. Winkler: Monitoring Arithmetic Temporal Properties on Finite Traces. Proc. 37th AAAI, 2023.