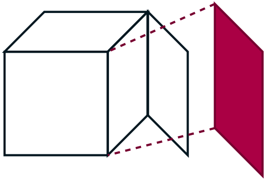
Monitoring Arithmetic Temporal Properties on Finite Traces

Sarah Winkler

Free University of Bozen-Bolzano, Italy

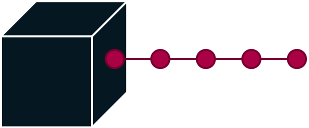
AAAI Spring Symposium
On the Effectiveness of Temporal Logics on Finite Traces in AI
27–29 March 2023, San Francisco

Checking properties of dynamic systems



- system **fully known**, **specification** available
- analyze **all** executions, or all execution trees

analysis task:
model checking

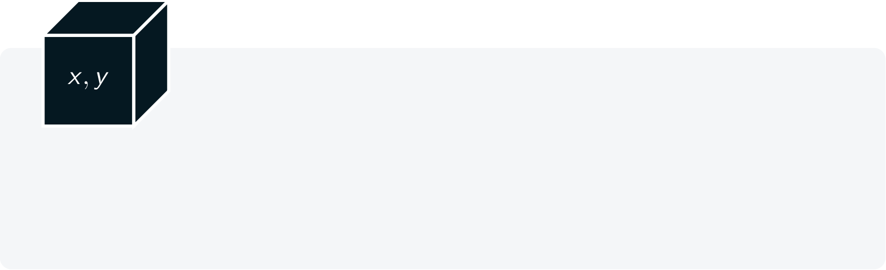


- system **unknown**, or properties **inaccessible**
- analyze **running execution** and its possible continuations

analysis task:
monitoring

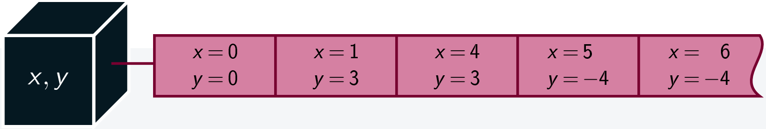
[FMPW23] P. Felli, M. Montali, F. Patrizi, S. Winkler. Monitoring Arithmetic Temporal Properties on Finite Traces. AAAI-37, 2023

Overview



- can access finite set of numeric **process variables** V

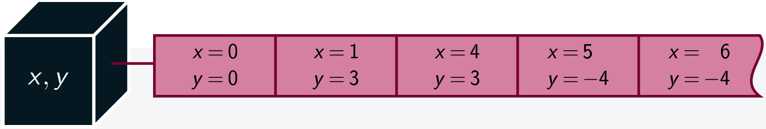
Overview



$x = 0$	$x = 1$	$x = 4$	$x = 5$	$x = 6$
$y = 0$	$y = 3$	$y = 3$	$y = -4$	$y = -4$

- can access finite set of numeric **process variables** V
- **trace** is finite sequence of assignments to V

Overview

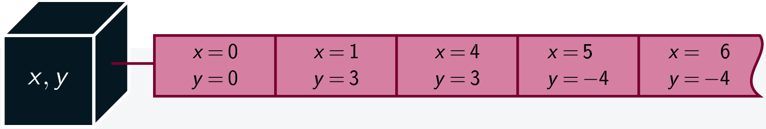


$x = 0$ $y = 0$	$x = 1$ $y = 3$	$x = 4$ $y = 3$	$x = 5$ $y = -4$	$x = 6$ $y = -4$

- $\psi_1 = (y \geqslant 0) \ \mathsf{U} \ (\mathsf{G}(x > y))$

- can access finite set of numeric **process variables** V
- **trace** is finite sequence of assignments to V
- **linear-time property** ψ with linear **arithmetic constraints** (**ALTL**$_f$)

Overview



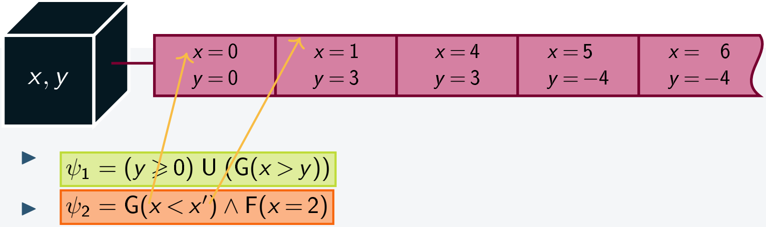
$x=0$	$x=1$	$x=4$	$x=5$	$x=\ 6$
$y=0$	$y=3$	$y=3$	$y=-4$	$y=-4$

- $\psi_1 = (y \geqslant 0)\ \mathsf{U}\ (\mathsf{G}(x > y))$
- $\psi_2 = \mathsf{G}(x < x') \wedge \mathsf{F}(x = 2)$

x' is value of x looking one trace instant ahead

- can access finite set of numeric **process variables** V
- **trace** is finite sequence of assignments to V
- **linear-time property** ψ with linear **arithmetic constraints** (**ALTL**$_f$)
  variables can have **lookahead** to refer to future values

Overview

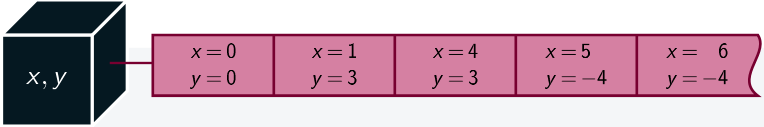


$x=0$	$x=1$	$x=4$	$x=5$	$x=6$
$y=0$	$y=3$	$y=3$	$y=-4$	$y=-4$

- $\psi_1 = (y \geqslant 0)\ \mathsf{U}\ (\mathsf{G}(x > y))$
- $\psi_2 = \mathsf{G}(x < x') \wedge \mathsf{F}(x = 2)$

- can access finite set of numeric **process variables** V
- **trace** is finite sequence of assignments to V
- **linear-time property** ψ with linear **arithmetic constraints** **(ALTL$_f$)**
  variables can have **lookahead** to refer to future values

Overview



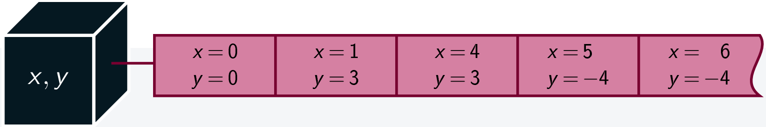
$x=0$	$x=1$	$x=4$	$x=5$	$x=6$
$y=0$	$y=3$	$y=3$	$y=-4$	$y=-4$

- $\psi_1 = (y \geqslant 0)\ \mathsf{U}\ (\mathsf{G}(x > y))$
- $\psi_2 = \mathsf{G}(x < x') \wedge \mathsf{F}(x = 2)$

"the current value of x is always less than the next one, and at some point x has value 2"

- can access finite set of numeric **process variables** V
- **trace** is finite sequence of assignments to V
- **linear-time property** ψ with linear **arithmetic constraints** (**ALTL**$_f$)
  variables can have **lookahead** to refer to future values

Overview



$x=0$	$x=1$	$x=4$	$x=5$	$x=6$
$y=0$	$y=3$	$y=3$	$y=-4$	$y=-4$

- $\psi_1 = (y \geqslant 0)\ \mathsf{U}\ (\mathsf{G}(x > y))$
- $\psi_2 = \mathsf{G}(x < x') \wedge \mathsf{F}(x = 2)$

- can access finite set of numeric **process variables** V
- **trace** is finite sequence of assignments to V
- **linear-time property** ψ with linear **arithmetic constraints** (**ALTL**$_f$)
  variables can have **lookahead** to refer to future values
- **anticipatory monitoring**: determine current and future satisfaction

Anticipatory monitoring task

given trace and ALTL$_f$ property, determine monitoring state:

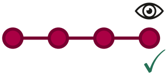
ps: permanent satisfaction

A. Bauer, M. Leucker, and C. Schallhart: Comparing LTL Semantics for Runtime Verification. J. Logic and Comput., 20(3): 651–674, 2010.

Anticipatory monitoring task

given trace and ALTL$_f$ property, determine monitoring state:

ps: permanent satisfaction

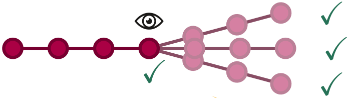


A. Bauer, M. Leucker, and C. Schallhart: Comparing LTL Semantics for Runtime Verification. J. Logic and Comput., 20(3): 651–674, 2010.

Anticipatory monitoring task

given trace and ALTL$_f$ property, determine monitoring state:

ps: permanent satisfaction



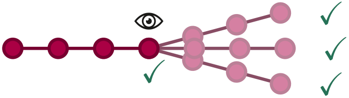
consider all finite continuations of unbounded length

A. Bauer, M. Leucker, and C. Schallhart: Comparing LTL Semantics for Runtime Verification. J. Logic and Comput., 20(3): 651–674, 2010.

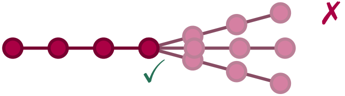
Anticipatory monitoring task

given trace and ALTL$_f$ property, determine monitoring state:

ps: permanent satisfaction



cs: current satisfaction



A. Bauer, M. Leucker, and C. Schallhart: Comparing LTL Semantics for Runtime Verification. J. Logic and Comput., 20(3): 651–674, 2010.

Anticipatory monitoring task

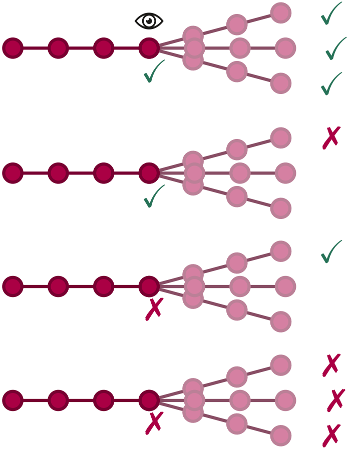
given trace and ALTL$_f$ property, determine monitoring state:

ps: permanent satisfaction

cs: current satisfaction

cv: current violation

pv: permanent violation



A. Bauer, M. Leucker, and C. Schallhart: Comparing LTL Semantics for Runtime Verification. J. Logic and Comput., 20(3): 651–674, 2010.

Anticipatory monitoring task

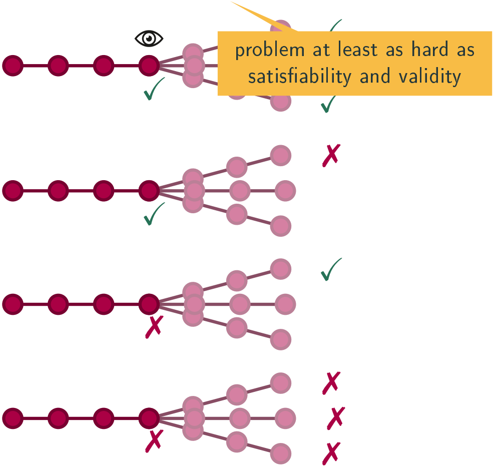
given trace and ALTL$_f$ property, determine monitoring state:

ps: permanent satisfaction

cs: current satisfaction

cv: current violation

pv: permanent violation



problem at least as hard as satisfiability and validity

A. Bauer, M. Leucker, and C. Schallhart: Comparing LTL Semantics for Runtime Verification. J. Logic and Comput., 20(3): 651–674, 2010.

Monitoring without lookahead

Theorem

monitoring of lookahead-free properties is solvable

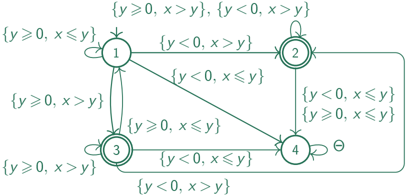
Monitoring without lookahead

Theorem

monitoring of lookahead-free properties is solvable

Example

- construct DFA for $(y \geqslant 0)\ \mathsf{U}\ (\mathsf{G}(x > y))$, treating constraints as propositions



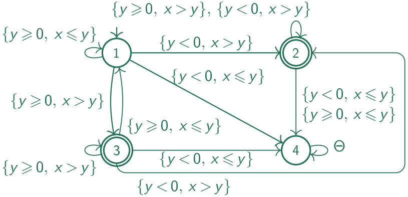
Monitoring without lookahead

Theorem

monitoring of lookahead-free properties is solvable

Example

- construct DFA for $(y \geqslant 0)\ \mathsf{U}\ (\mathsf{G}(x > y))$, treating constraints as propositions


$\{y \geqslant 0,\ x > y\},\ \{y < 0,\ x > y\}$
$\{y \geqslant 0,\ x \leqslant y\}$
1
$\{y < 0,\ x > y\}$
2
$\{y < 0,\ x \leqslant y\}$
$\{y \geqslant 0,\ x > y\}$
$\{y < 0,\ x \leqslant y\}$
$\{y \geqslant 0,\ x \leqslant y\}$
$\{y \geqslant 0,\ x \leqslant y\}$
3
4
$\ominus$
$\{y \geqslant 0,\ x > y\}$
$\{y < 0,\ x \leqslant y\}$
$\{y < 0,\ x > y\}$

- every trace prefix leads to unique DFA state

$x = 0$	$x = 1$	$x = 4$	$x = 5$	$x = 6$
$y = 0$	$y = 3$	$y = 3$	$y = 4$	$y = -4$
A	A	C	C	B

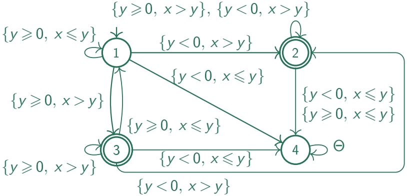
Monitoring without lookahead

Theorem

monitoring of lookahead-free properties is solvable

Example

- construct DFA for $(y \geqslant 0)\ \mathsf{U}\ (\mathsf{G}(x > y))$, treating constraints as propositions



$\{y \geqslant 0,\ x > y\}, \{y < 0,\ x > y\}$
$\{y \geqslant 0,\ x \leqslant y\}$
1
$\{y < 0,\ x > y\}$
2
$\{y < 0,\ x \leqslant y\}$
$\{y \geqslant 0,\ x > y\}$
$\{y < 0,\ x \leqslant y\}$
$\{y \geqslant 0,\ x \leqslant y\}$
$\{y \geqslant 0,\ x \leqslant y\}$
3
4
$\ominus$
$\{y \geqslant 0,\ x > y\}$
$\{y < 0,\ x \leqslant y\}$
$\{y < 0,\ x > y\}$

- every trace prefix leads to unique DFA state

$x = 0$ $y = 0$	$x = 1$ $y = 3$	$x = 4$ $y = 3$	$x = 5$ $y = 4$	$x = 6$ $y = -4$
A **cv**	A **cv**	C **cs**	C **cs**	B **cs**

Monitoring without lookahead

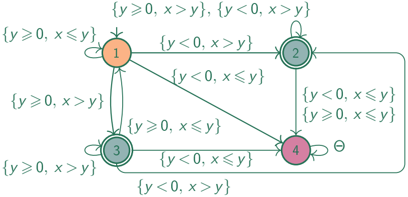
every DFA state q corresponds to unique monitoring state

Theorem

monitoring of lookahead-free properties is solvable: DFAs serve as monitors

Example

- construct DFA for $(y \geqslant 0)\ \mathsf{U}\ (\mathsf{G}(x > y))$, treating constraints as propositions



$\{y \geqslant 0,\ x > y\},\ \{y < 0,\ x > y\}$
$\{y \geqslant 0,\ x \leqslant y\}$
1
$\{y < 0,\ x > y\}$
2
$\{y < 0,\ x \leqslant y\}$
$\{y \geqslant 0,\ x > y\}$
$\{y < 0,\ x \leqslant y\}$
$\{y \geqslant 0,\ x \leqslant y\}$
$\{y \geqslant 0,\ x \leqslant y\}$
3
4
$\ominus$
$\{y < 0,\ x \leqslant y\}$
$\{y \geqslant 0,\ x > y\}$
$\{y < 0,\ x > y\}$

- every trace prefix leads to unique DFA state

$x=0$	$x=1$	$x=4$	$x=5$	$x=\ \ 6$
$y=0$	$y=3$	$y=3$	$y=4$	$y=-4$

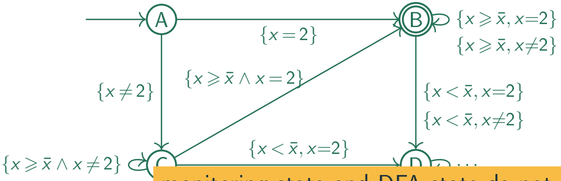
A A C C B

cv **cv** **cs** **cs** **cs**

Monitoring with lookahead is not solvable

Example (DFAs are not monitors)

- DFAs construction for $\boxed{\mathsf{G}(x' > x) \wedge \mathsf{F}(x = 2)}$


A
B
C
D
$\{x = 2\}$
$\{x \neq 2\}$
$\{x \geqslant \bar{x} \wedge x = 2\}$
$\{x \geqslant \bar{x}, x{=}2\}$
$\{x \geqslant \bar{x}, x{\neq}2\}$
$\{x < \bar{x}, x{=}2\}$
$\{x < \bar{x}, x{\neq}2\}$
$\{x < \bar{x}, x{=}2\}$
$\{x \geqslant \bar{x} \wedge x \neq 2\}$
$\cdots$

monitoring state and DFA state do not correspond

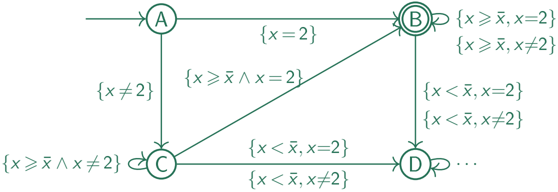
- sequence of monitoring states and DFA states

$x = 0$ $y = 0$	$x = 1$ $y = 3$	$x = 4$ $y = 3$	$x = 5$ $y = 4$	$x = \ \ 6$ $y = -4$
cv	**cv**	**pv**	**pv**	**pv**
C	C	C	C	C

Monitoring with lookahead is not solvable

Example (DFAs are not monitors)

- DFAs construction for $\boxed{\mathsf{G}(x' > x) \wedge \mathsf{F}(x = 2)}$



A
B
C
D
$\{x=2\}$
$\{x \geqslant \bar{x}, x{=}2\}$
$\{x \geqslant \bar{x}, x{\neq}2\}$
$\{x \neq 2\}$
$\{x \geqslant \bar{x} \wedge x = 2\}$
$\{x < \bar{x}, x{=}2\}$
$\{x < \bar{x}, x{\neq}2\}$
$\{x \geqslant \bar{x} \wedge x \neq 2\}$
$\{x < \bar{x}, x{=}2\}$
$\{x < \bar{x}, x{\neq}2\}$
$\cdots$

- sequence of monitoring states and DFA states

$x=0$	$x=1$	$x=4$	$x=5$	$x=\ \ 6$
$y=0$	$y=3$	$y=3$	$y=4$	$y=-4$
cv	**cv**	**pv**	**pv**	**pv**
C	C	C	C	C

Fact

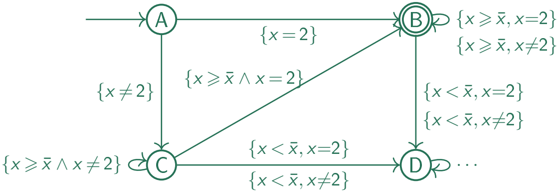
Monitoring with lookahead is not solvable: reduction from reachability in 2CM

Monitoring with lookahead is not solvable

problem: state reachability depends on assignment

Example (DFAs are not monitors)

- DFAs construction for $\boxed{G(x' > x) \wedge F(x = 2)}$


A
B
C
D
$\{x = 2\}$
$\{x \neq 2\}$
$\{x \geqslant \bar{x} \wedge x = 2\}$
$\{x \geqslant \bar{x}, x{=}2\}$
$\{x \geqslant \bar{x}, x{\neq}2\}$
$\{x < \bar{x}, x{=}2\}$
$\{x < \bar{x}, x{\neq}2\}$
$\{x \geqslant \bar{x} \wedge x \neq 2\}$
$\{x < \bar{x}, x{=}2\}$
$\{x < \bar{x}, x{\neq}2\}$
$\cdots$

- sequence of monitoring states and DFA states

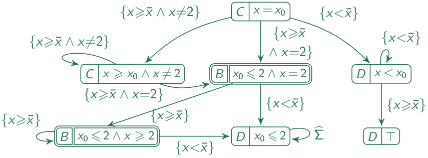
$x = 0$ $y = 0$	$x = 1$ $y = 3$	$x = 4$ $y = 3$	$x = 5$ $y = 4$	$x = 6$ $y = -4$
cv C	**cv** C	**pv** C	**pv** C	**pv** C

Fact

Monitoring with lookahead is not solvable: reduction from reachability in 2CM

Constraint graphs: Symbolic finite state abstraction

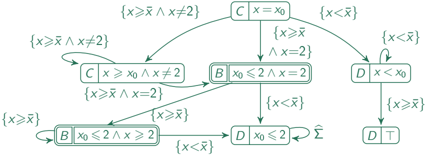
- CG(q) represents accumulated constraints for all paths from q in DFA


$\{x \geqslant \bar{x} \wedge x \neq 2\}$
C $x = x_0$
$\{x < \bar{x}\}$
$\{x \geqslant \bar{x} \wedge x \neq 2\}$
$\{x \geqslant \bar{x}$
$\wedge\, x = 2\}$
$\{x < \bar{x}\}$
C $x \geqslant x_0 \wedge x \neq 2$
B $x_0 \leqslant 2 \wedge x = 2$
D $x < x_0$
$\{x \geqslant \bar{x} \wedge x = 2\}$
$\{x < \bar{x}\}$
$\{x \geqslant \bar{x}\}$
$\{x \geqslant \bar{x}\}$
$\{x \geqslant \bar{x}\}$
B $x_0 \leqslant 2 \wedge x \geqslant 2$
D $x_0 \leqslant 2$
$\widehat{\Sigma}$
D $\top$
$\{x < \bar{x}\}$

Constraint graphs: Symbolic finite state abstraction

- CG(q) represents accumulated constraints for all paths from

can be infinite

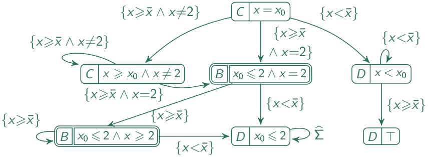

$\{x \geqslant \bar{x} \wedge x \neq 2\}$
C | $x = x_0$
$\{x < \bar{x}\}$
$\{x \geqslant \bar{x} \wedge x \neq 2\}$
$\{x \geqslant \bar{x}$
$\wedge\, x = 2\}$
$\{x < \bar{x}\}$
C | $x \geqslant x_0 \wedge x \neq 2$
B | $x_0 \leqslant 2 \wedge x = 2$
D | $x < x_0$
$\{x \geqslant \bar{x} \wedge x = 2\}$
$\{x < \bar{x}\}$
$\{x \geqslant \bar{x}\}$
$\{x \geqslant \bar{x}\}$
$\{x \geqslant \bar{x}\}$
B | $x_0 \leqslant 2 \wedge x \geqslant 2$
D | $x_0 \leqslant 2$
$\widehat{\Sigma}$
D | $\top$
$\{x < \bar{x}\}$

Key property

if CG is finite, it is faithful finite state abstraction

Constraint graphs: Symbolic finite state abstraction

- CG(q) represents accumulated constraints for all paths from q in DFA



$\{x \geqslant \bar{x} \wedge x \neq 2\}$
C | $x = x_0$
$\{x < \bar{x}\}$
$\{x \geqslant \bar{x} \wedge x \neq 2\}$
$\{x \geqslant \bar{x}$ $\wedge\, x = 2\}$
$\{x < \bar{x}\}$
C | $x \geqslant x_0 \wedge x \neq 2$
B | $x_0 \leqslant 2 \wedge x = 2$
D | $x < x_0$
$\{x \geqslant \bar{x} \wedge x = 2\}$
$\{x < \bar{x}\}$
$\{x \geqslant \bar{x}\}$
$\{x \geqslant \bar{x}\}$
$\{x \geqslant \bar{x}\}$
B | $x_0 \leqslant 2 \wedge x \geqslant 2$
D | $x_0 \leqslant 2$
$\widehat{\Sigma}$
D | $\top$
$\{x < \bar{x}\}$

- formulas in nodes give **condition** on initial variable values
  - to reach final states: FSat(CG(q))
  - to reach non-final states: FUns(CG(q))

Key property

if CG is finite, it is faithful finite state abstraction

Monitoring procedure

all monitoring structures can be computed upfront (DFA, CGs, FSat, FUns)

```
procedure MONITOR($\psi$, $\tau$)
    compute DFA for $\psi$
    $w \leftarrow$ word over constraints consistent with $\tau$
    $q \leftarrow$ DFA state in such that $\{q_0\} \rightarrow^*_w q$
    $\alpha \leftarrow$ last assignment in $\tau$
    if $q$ accepting in DFA then
        return (cs if $\alpha \models$ FUns(CG($q$)) else ps)
    else return (cv if $\alpha \models$ FSat(CG($q$)) else pv)
```

Monitoring procedure

all monitoring structures can be computed upfront
(DFA, CGs, FSat, FUns)

```
procedure MONITOR($\psi$, $\tau$)
    compute DFA for $\psi$
    $w \leftarrow$ word over constraints consistent with $\tau$
    $q \leftarrow$ DFA state in such that $\{q_0\} \rightarrow_w^* q$
    $\alpha \leftarrow$ last assignment in $\tau$
    if $q$ accepting in DFA then
        return (cs if $\alpha \models$ FUns(CG($q$)) else ps)
    else return (cv if $\alpha \models$ FSat(CG($q$)) else pv)
```

Theorem (Correctness)
if $\textsc{monitor}(\psi, \tau) = s$ *then s is monitoring state for* ψ *and* τ

Monitoring procedure

all monitoring structures can be computed upfront (DFA, CGs, FSat, FUns)

```
**procedure** MONITOR($\psi$, $\tau$)
    compute DFA for $\psi$
    $w \leftarrow$ word over constraints consistent with $\tau$
    $q \leftarrow$ DFA state in such that $\{q_0\} \rightarrow_w^* q$
    $\alpha \leftarrow$ last assignment in $\tau$
    **if** $q$ accepting in DFA **then**
        **return** (**cs if** $\alpha \models$ FUns(CG($q$)) **else ps**)
    **else return** (**cv if** $\alpha \models$ FSat(CG($q$)) **else pv**)
```

does not terminate if CGs infinite

Theorem (Correctness)
if $\textsc{monitor}(\psi, \tau) = s$ *then* s *is monitoring state for* ψ *and* τ

Abstract solvability criterion

previously used in context of model checking [FMW22]

Definition (Finite summary)

property ψ has finite summary if paths in DFA for ψ are covered by finitely many history constraints

[FMW22] P. Felli, M. Montali, S. Winkler. Linear-time verification of data-aware dynamic systems with arithmetic. AAAI-36(5), 5642-5650, 2022

Abstract solvability criterion

Definition (Finite summary)

property ψ has finite summary if paths in DFA for ψ are covered by finitely many history constraints

Observation

for properties with finite summary, constraint graphs are finite

Theorem

monitoring task is solvable for any ψ that has finite summary, and MONITOR *is monitoring procedure*

[FMW22] P. Felli, M. Montali, S. Winkler. Linear-time verification of data-aware dynamic systems with arithmetic. AAAI-36(5), 5642-5650, 2022

Concrete solvable property classes

Property classes that enjoy finite summary

- **monotonicity constraint** properties over $\mathbb{Q}$ or $\mathbb{Z}$ $\mathsf{G}(x' > x) \land \mathsf{F}(x = 2)$
  (all constraints are variable-to-variable/constant comparisons)

S. Demri and D. D'Souza: An automata-theoretic approach to constraint LTL. Inform. Comput., 205(3): 380-415, 2007.

Concrete solvable property classes

Property classes that enjoy finite summary

- **monotonicity constraint** properties over $\mathbb{Q}$ or $\mathbb{Z}$ $\mathsf{G}(x' > x) \land \mathsf{F}(x = 2)$
  (all constraints are variable-to-variable/constant comparisons)
- **integer periodicity constraint** properties $\mathsf{F}(x' > 3) \land \mathsf{G}(x \equiv_7 2)$
  (variable-to-variable/constant comparisons with modulo operator)

S. Demri: LTL over integer periodicity constraints. Theor. Comput. Sci., 360(1-3): 96–123, 2006.

Concrete solvable property classes

Property classes that enjoy finite summary

- **monotonicity constraint** properties over $\mathbb{Q}$ or $\mathbb{Z}$ $\mathsf{G}(x' > x) \wedge \mathsf{F}(x = 2)$
  (all constraints are variable-to-variable/constant comparisons)
- **integer periodicity constraint** properties $\mathsf{F}(x' > 3) \wedge \mathsf{G}(x \equiv_7 2)$
  (variable-to-variable/constant comparisons with modulo operator)
- **bounded lookback** properties $\mathsf{F}(x' > y) \wedge \mathsf{G}(x + z = 7)$
  (restrict constraint interaction via lookahead, generalizes feedback freedom)

E. Damaggio, A. Deutsch and V. Vianu: Artifact systems with data dependencies and arithmetic. ACM Trans. Database Syst., 37(3): 22:1–22:36, 2012

Concrete solvable property classes

Property classes that enjoy finite summary

- **monotonicity constraint** properties over $\mathbb{Q}$ or $\mathbb{Z}$ $\mathsf{G}(x' > x) \wedge \mathsf{F}(x = 2)$
  (all constraints are variable-to-variable/constant comparisons)
- **integer periodicity constraint** properties $\mathsf{F}(x' > 3) \wedge \mathsf{G}(x \equiv_7 2)$
  (variable-to-variable/constant comparisons with modulo operator)
- **bounded lookback** properties $\mathsf{F}(x' > y) \wedge \mathsf{G}(x + z = 7)$
  (restrict constraint interaction via lookahead, generalizes feedback freedom)

Non-solvable class

- **gap-order** properties $\mathsf{G}(x' - y \geqslant 3) \wedge \mathsf{F}(x - z' \geqslant 2)$
  (all constraints are gap-order comparisons)

L. Bozzelli and S. Pinchinat: Verification of gap- order constraint abstractions of counter systems. Theor. Comput. Sci., 523: 1–36, 2014

Concrete solvable property classes

Property classes that enjoy finite summary

- **monotonicity constraint** properties over $\mathbb{Q}$ or $\mathbb{Z}$ $\mathsf{G}(x' > x) \wedge \mathsf{F}(x = 2)$
  (all constraints are variable-to-variable/constant comparisons)
- **integer periodicity constraint** properties $\mathsf{F}(x' > 3) \wedge \mathsf{G}(x \equiv_7 2)$
  (variable-to-variable/constant comparisons with modulo operator)
- **bounded lookback** properties $\mathsf{F}(x' > y) \wedge \mathsf{G}(x + z = 7)$
  (restrict constraint interaction via lookahead, generalizes feedback freedom)

Non-solvable class

model checking is decidable

- **gap-order** properties $\mathsf{G}(x' - y \geqslant 3) \wedge \mathsf{F}(x - z' \geqslant 2)$
  (all constraints are gap-order comparisons)

L. Bozzelli and S. Pinchinat: Verification of gap- order constraint abstractions of counter systems. Theor. Comput. Sci., 523: 1–36, 2014

Summary

1. ALTL$_f$ monitoring with linear arithmetic constraints:

   without lookahead: solvable (DFA construction for monitors)

   with lookahead: not solvable

Summary

1. ALTL_f monitoring with linear arithmetic constraints:

   without lookahead: solvable (DFA construction for monitors)

   with lookahead: not solvable

2. general monitoring procedure for lookahead properties:

   terminates for finite summary properties

Summary

1. ALTL_f monitoring with linear arithmetic constraints:

   without lookahead: solvable (DFA construction for monitors)

   with lookahead: not solvable

2. general monitoring procedure for lookahead properties:

   terminates for finite summary properties

3. solvability for several practical classes of formulae:

   monotonicity and integer periodicity constraints, bounded lookback

Summary

1. ALTL$_f$ monitoring with linear arithmetic constraints:
   without lookahead: solvable (DFA construction for monitors)
   with lookahead: not solvable
2. general monitoring procedure for lookahead properties:
   terminates for finite summary properties
3. solvability for several practical classes of formulae:
   monotonicity and integer periodicity constraints, bounded lookback
4. SMT-based prototype ada witnesses feasibility of approach

Summary

1. ALTL$_f$ monitoring with linear arithmetic constraints:

   without lookahead: solvable (DFA construction for monitors)

   with lookahead: not solvable

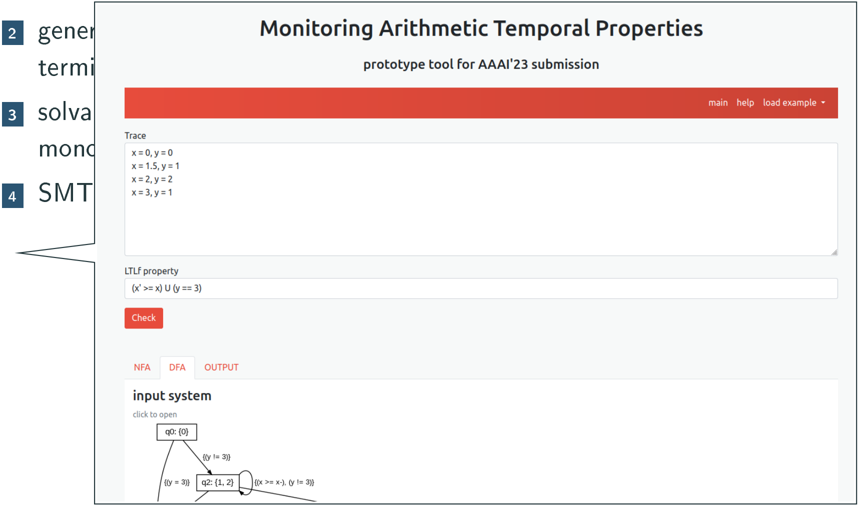
2. gener

   termi

3. solva

   mono

4. SMT



Monitoring Arithmetic Temporal Properties

prototype tool for AAAI'23 submission

Trace

```
x = 0, y = 0
x = 1.5, y = 1
x = 2, y = 2
x = 3, y = 1
```

LTLf property

```
(x' >= x) U (y == 3)
```

input system

click to open

Summary

1. ALTL_f monitoring with linear arithmetic constraints:
   without lookahead: solvable (DFA construction for monitors)
   with lookahead: not solvable
2. general monitoring procedure for lookahead properties:
   terminates for finite summary properties
3. solvability for several practical classes of formulae:
   monotonicity and integer periodicity constraints, bounded lookback
4. SMT-based prototype ada witnesses feasibility of approach

Future work

- lift approach to richer properties equipped with full-fledged relations
- possibly study more general, controlled first-order quantification across time