UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR SOFTWARE ENGINEERING
AND PROGRAMMING LANGUAGES

isp

On an Extension of Freeze LTL

Ordered Attributes and Decidability

Normann Decker Daniel Thoma

{decker,thoma}@isp.uni-luebeck.de

Institute for Software Engineering and Programming Languages
University of Lübeck, Germany

Laboratoire d'Informatique de Paris-Nord
Université Paris 13, Sorbonne Paris Cité

Paris, 4 March 2015

Outline

Setting and Motivation

Ordered Attributes in Freeze LTL

Results and Techniques

- Undecidability and Decidable Fragments
- Nested Counter Systems (NCS)
- Reductions To and From NCS

Summary

Multi-attributed Data Words

$$lock \quad use \quad unlock \quad lock \quad lock \quad use$$

- Letters $\Sigma = \{lock, use, unlock\}$,
- Attributes $A = \{\mathrm{p}, \mathrm{r}\}$, data values $\Delta = (\mathbb{N}, =)$
- Data valuation $\mathbf{d} : A \to \Delta$

Multi-attributed Data Words

	lock	*use*	*unlock*	*lock*	*lock*	*use*
r						
p						

- Letters $\Sigma = \{lock, use, unlock\}$,
- Attributes $A = \{\mathrm{p}, \mathrm{r}\}$, data values $\Delta = (\mathbb{N}, =)$
- Data valuation $\mathbf{d} : A \to \Delta$

Multi-attributed Data Words

	lock	*use*	*unlock*	*lock*	*lock*	*use*
r						
p						

- Letters $\Sigma = \{lock, use, unlock\}$,
- Attributes $A = \{\mathrm{p}, \mathrm{r}\}$, data values $\Delta = (\mathbb{N}, =)$
- Data valuation $\mathbf{d} : A \to \Delta$

$$w = (a_1, \mathbf{d}_1)(a_2, \mathbf{d}_2) \ldots (a_n, \mathbf{d}_n) \in (\Sigma \times \Delta^A)^+$$

Multi-attributed Data Words

	lock	*use*	*unlock*	*lock*	*lock*	*use*
r	1	1	1	2	1	2
p	1	1	1	2	2	1

- Letters $\Sigma = \{lock, use, unlock\}$,
- Attributes $A = \{\mathrm{p}, \mathrm{r}\}$, data values $\Delta = (\mathbb{N}, =)$
- Data valuation $\mathbf{d} : A \to \Delta$

$$\vec{w} = (a_1, \mathbf{d}_1)(a_2, \mathbf{d}_2) \ldots (a_n, \mathbf{d}_n) \in (\Sigma \times \Delta^A)^+$$

Multi-attributed Data Words

	lock	*use*	*unlock*	*lock*	*lock*	*use*
r	1	1	1	2	1	2
p	1	1	1	2	2	1

- Letters $\Sigma = \{lock, use, unlock\}$,
- Attributes $A = \{\mathrm{p}, \mathrm{r}\}$, data values $\Delta = (\mathbb{N}, =)$
- Data valuation $\mathbf{d} : A \to \Delta$

$$w = (a_1, \mathbf{d}_1)(a_2, \mathbf{d}_2)\ldots(a_n, \mathbf{d}_n) \in (\Sigma \times \Delta^A)^+$$

No Foreign Access

	lock	*use*	*unlock*	*lock*	*lock*	*use*
r	1	1	1	2	1	2
p	1	1	1	2	2	1

Release every lock eventually

No Foreign Access

	lock	*use*	*unlock*	*lock*	*lock*	*use*
r	1	1	1	2	1	2
p	1	1	1	2	2	1

Release every lock eventually

$$\mathrm{G}\left(\mathit{lock} \rightarrow \downarrow^{\mathrm{r}} \left(\mathrm{F}\, \mathit{unlock} \wedge \uparrow^{\mathrm{r}} \right) \right)$$

No Foreign Access

	lock	*use*	*unlock*	*lock*	*lock*	*use*
r	1	1	1	2	1	2
p	1	1	1	2	2	1

No foreign access to locked resources

No Foreign Access

	lock	*use*	*unlock*	*lock*	*lock*	*use*
r	1	1	1	2	1	2
p	1	1	1	2	2	1

No foreign access to locked resources

$$\mathrm{G}\left(\mathit{lock} \rightarrow \downarrow^{(\mathrm{r},\mathrm{p})} (\mathit{use} \wedge \uparrow^{\mathrm{r}} \rightarrow \uparrow^{\mathrm{p}}) \, \mathrm{U} (\mathit{unlock} \wedge \uparrow^{(\mathrm{r},\mathrm{p})}) \right)$$

Undecidability Results

- Freeze LTL with *2 registers*: $\downarrow_1 \downarrow_2$

  [Demri, Lazic, Nowak '05]

- LRV$_{vec}$: $(a_1, a_2) = \mathrm{X}\,\mathrm{F}(b_1, b_2)$

  [Demri, Figueira, Praveen '13]

- Basic Data LTL over tuples $\mathcal{C}_{(a_1,a_2)}\,\mathrm{X}^{=}\,@(b_1, b_2)$

  [Kara, Schwentick, Zeume '10]

- FO$^2(\sim_1, \sim_2, +1, <)$

  [Bojanczyk, David, Muscholl, Schwentick, Segoufin '06]

No Foreign Access

	lock	*use*	*unlock*	*lock*	*lock*	*use*
r	1	1	1	2	1	2
p	1	1	1	2	2	1

No foreign access to locked resources

$$\mathrm{G}\left(\mathit{lock} \rightarrow \downarrow^{(\mathrm{r},\mathrm{p})} (\mathit{use} \wedge \uparrow^{\mathrm{r}} \rightarrow \uparrow^{\mathrm{p}}) \, \mathrm{U} (\mathit{unlock} \wedge \uparrow^{(\mathrm{r},\mathrm{p})}) \right)$$

No Foreign Access

	lock	*use*	*unlock*	*lock*	*lock*	*use*
r	1	1	1	2	1	2
p	1	1	1	2	2	1

No foreign access to locked resources

$$\mathrm{G}\left(\mathit{lock} \rightarrow \downarrow^{(\mathrm{r},\mathrm{p})} (\mathit{use} \wedge \uparrow^{\mathrm{r}} \rightarrow \uparrow^{(\mathrm{r},\mathrm{p})}) \,\mathrm{U}(\mathit{unlock} \wedge \uparrow^{(\mathrm{r},\mathrm{p})}) \right)$$

No Foreign Access

	lock	*use*	*unlock*	*lock*	*lock*	*use*
r	1	1	1	2	1	2
$\sqsubseteq$						
p	1	1	1	2	2	1

No foreign access to locked resources

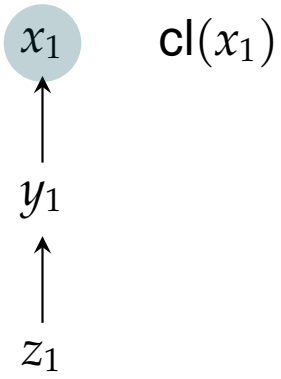
$$\mathrm{G}\left(\mathit{lock} \rightarrow \downarrow^{(\mathrm{r},\mathrm{p})} (\mathit{use} \wedge \uparrow^{\mathrm{r}} \rightarrow \uparrow^{(\mathrm{r},\mathrm{p})}) \,\mathrm{U}(\mathit{unlock} \wedge \uparrow^{(\mathrm{r},\mathrm{p})}) \right)$$

Attribute Dependencies

$$\begin{array}{c} x_1 \\ \uparrow \\ y_1 \\ \uparrow \\ z_1 \end{array}$$

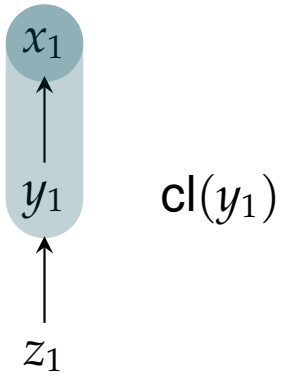
- Storing mechanism
  - Downward closures $\mathsf{cl}(x) := \{x' \in A \mid x' \sqsubseteq x\}$
  - Restrictions $\mathbf{d}|_{A'} : \mathsf{dom}(\mathbf{d}) \cap A' \to \Delta$
- $(A, \sqsubseteq)$ reflexive, transitive relation (*quasi-ordering*)

Attribute Dependencies



- Storing mechanism
  - Downward closures $\mathsf{cl}(x) := \{x' \in A \mid x' \sqsubseteq x\}$
  - Restrictions $\mathbf{d}|_{A'} : \mathsf{dom}(\mathbf{d}) \cap A' \to \Delta$
- $(A, \sqsubseteq)$ reflexive, transitive relation (*quasi-ordering*)

Attribute Dependencies



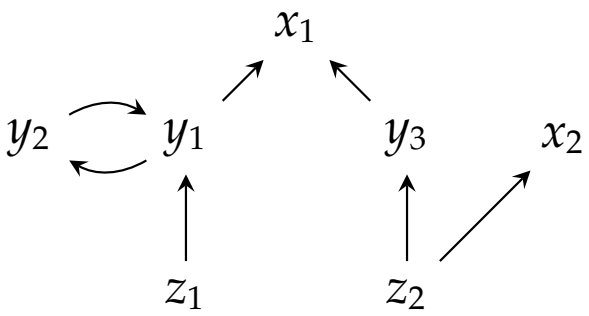
- Storing mechanism
  - Downward closures $\mathsf{cl}(x) := \{x' \in A \mid x' \sqsubseteq x\}$
  - Restrictions $\mathbf{d}|_{A'} : \mathsf{dom}(\mathbf{d}) \cap A' \to \Delta$
- $(A, \sqsubseteq)$ reflexive, transitive relation (*quasi-ordering*)

Attribute Dependencies

$\mathsf{cl}(z_1) = A$

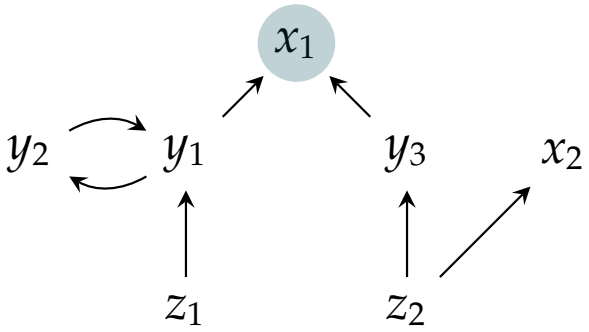
- Storing mechanism
  - Downward closures $\mathsf{cl}(x) := \{x' \in A \mid x' \sqsubseteq x\}$
  - Restrictions $\mathbf{d}|_{A'} : \mathsf{dom}(\mathbf{d}) \cap A' \to \Delta$
- $(A, \sqsubseteq)$ reflexive, transitive relation (*quasi-ordering*)

Attribute Dependencies



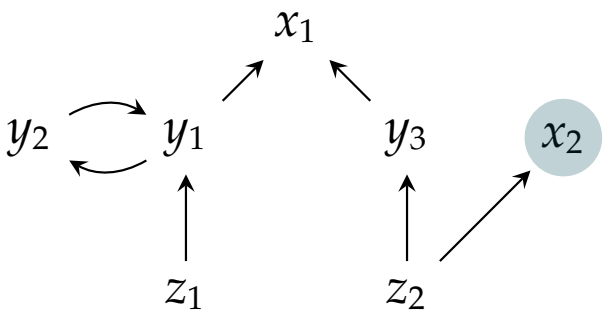
- Storing mechanism
  - Downward closures $\mathsf{cl}(x) := \{x' \in A \mid x' \sqsubseteq x\}$
  - Restrictions $\mathbf{d}|_{A'} : \mathsf{dom}(\mathbf{d}) \cap A' \to \Delta$
- $(A, \sqsubseteq)$ reflexive, transitive relation (*quasi-ordering*)

Attribute Dependencies



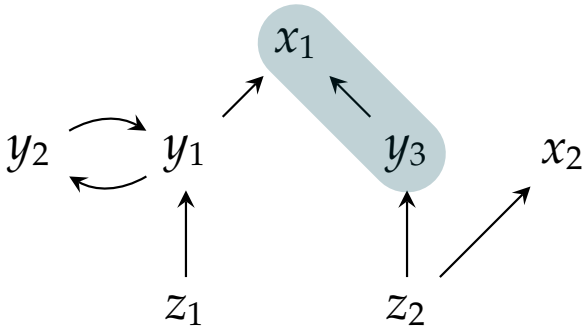
- Storing mechanism
  - Downward closures $\mathrm{cl}(x) := \{x' \in A \mid x' \sqsubseteq x\}$
  - Restrictions $\mathbf{d}|_{A'} : \mathrm{dom}(\mathbf{d}) \cap A' \to \Delta$
- $(A, \sqsubseteq)$ reflexive, transitive relation (*quasi-ordering*)

Attribute Dependencies



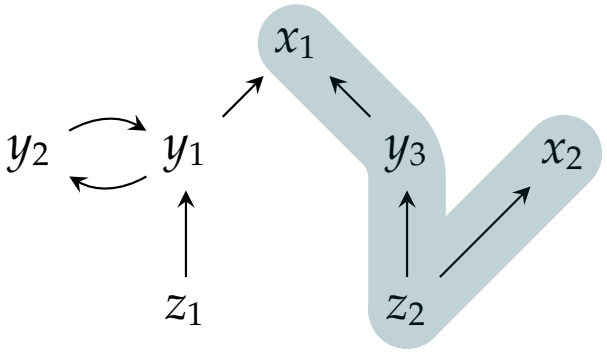
- Storing mechanism
  - Downward closures $\mathrm{cl}(x) := \{x' \in A \mid x' \sqsubseteq x\}$
  - Restrictions $\mathbf{d}|_{A'} : \mathrm{dom}(\mathbf{d}) \cap A' \to \Delta$
- $(A, \sqsubseteq)$ reflexive, transitive relation (*quasi-ordering*)

Attribute Dependencies



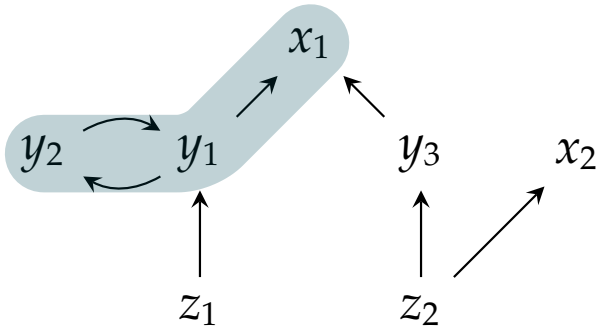
- Storing mechanism
  - Downward closures $\mathsf{cl}(x) := \{x' \in A \mid x' \sqsubseteq x\}$
  - Restrictions $\mathbf{d}|_{A'} : \mathsf{dom}(\mathbf{d}) \cap A' \to \Delta$
- $(A, \sqsubseteq)$ reflexive, transitive relation (*quasi-ordering*)

Attribute Dependencies



- Storing mechanism
  - Downward closures $\mathsf{cl}(x) := \{x' \in A \mid x' \sqsubseteq x\}$
  - Restrictions $\mathbf{d}|_{A'} : \mathsf{dom}(\mathbf{d}) \cap A' \to \Delta$
- $(A, \sqsubseteq)$ reflexive, transitive relation (*quasi-ordering*)

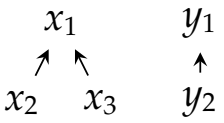
Attribute Dependencies

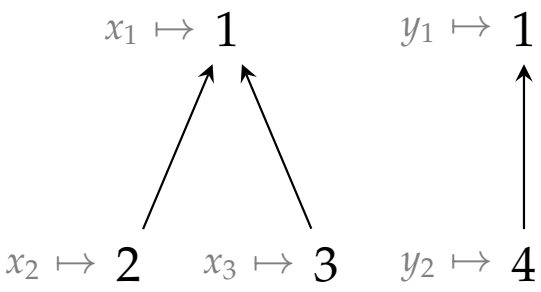


- Storing mechanism
  - Downward closures $\mathsf{cl}(x) := \{x' \in A \mid x' \sqsubseteq x\}$
  - Restrictions $\mathbf{d}|_{A'} : \mathsf{dom}(\mathbf{d}) \cap A' \to \Delta$
- $(A, \sqsubseteq)$ reflexive, transitive relation (*quasi-ordering*)

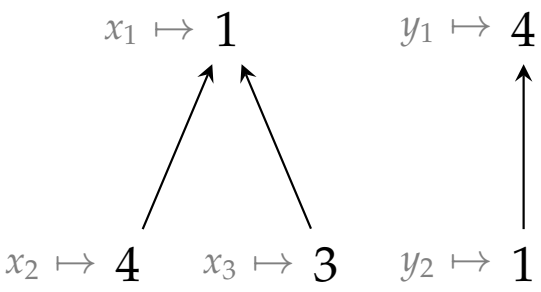
Comparison

$(A, \sqsubseteq) =$





$\mathbf{d} : A \to \Delta$



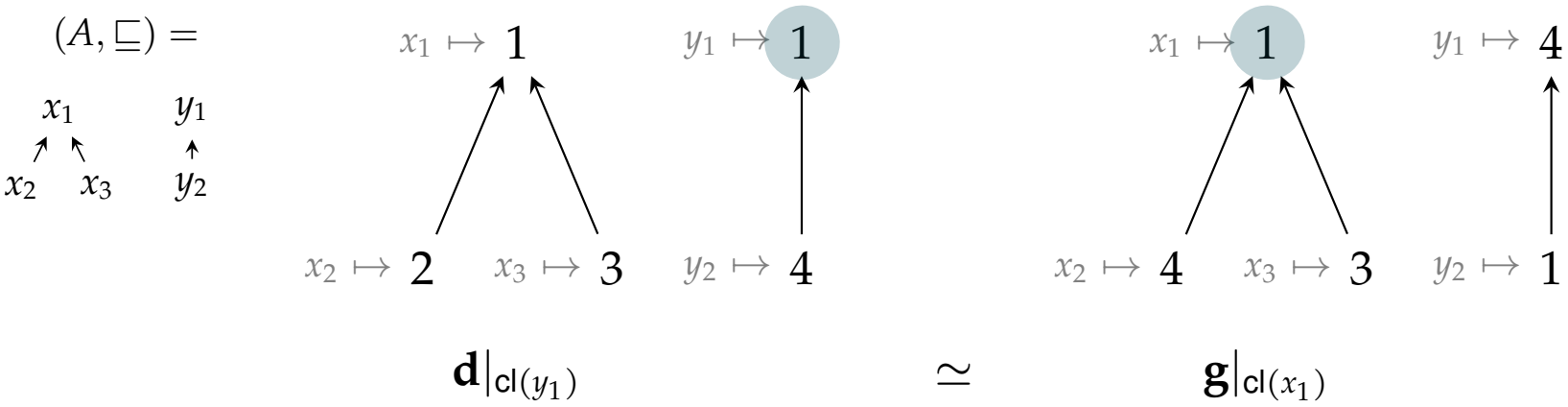
$\mathbf{g} : A \to \Delta$

Definition (Valuation Equivalence)

Let $\mathbf{d} \simeq \mathbf{g}$ iff

- $\exists$ *order preserving bijection* $h : \mathrm{dom}(\mathbf{d}) \to \mathrm{dom}(\mathbf{g})$ s.t.
- $\forall x \in \mathrm{dom}(\mathbf{d}) : \mathbf{d}(x) = \mathbf{g}(h(x))$.

Comparison

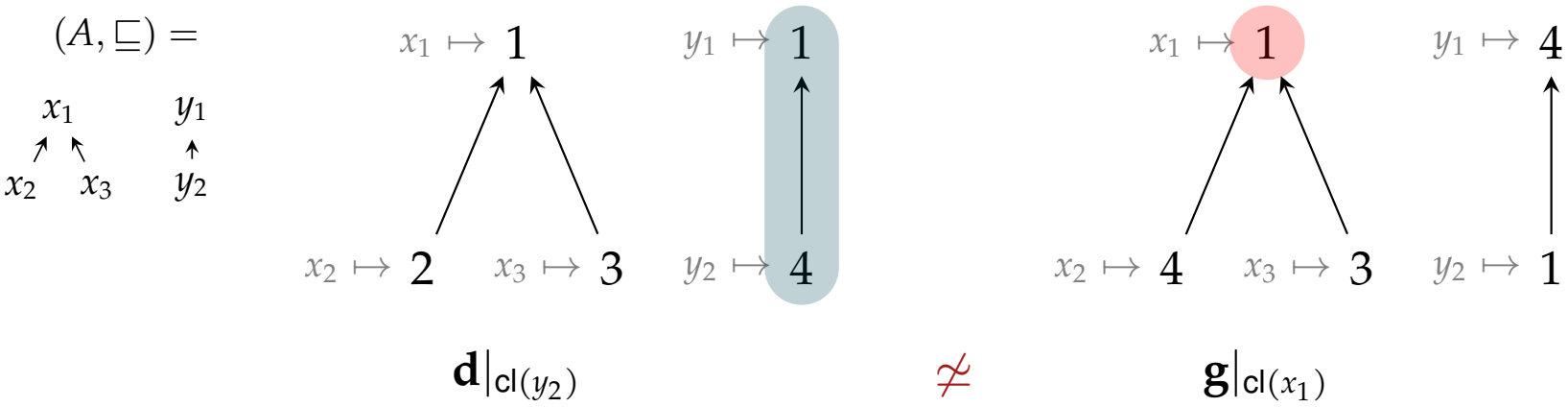


Definition (Valuation Equivalence)

Let $\mathbf{d} \simeq \mathbf{g}$ iff

- $\exists$ *order preserving bijection* $h : \mathsf{dom}(\mathbf{d}) \to \mathsf{dom}(\mathbf{g})$ s.t.
- $\forall x \in \mathsf{dom}(\mathbf{d}) : \mathbf{d}(x) = \mathbf{g}(h(x))$.

Comparison

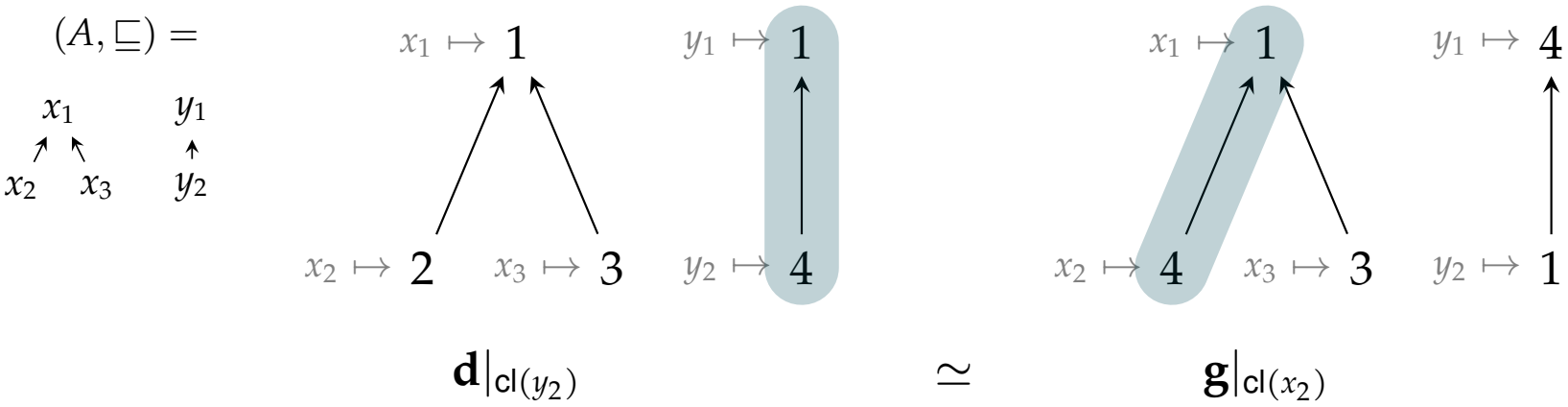


Definition (Valuation Equivalence)

Let $\mathbf{d} \simeq \mathbf{g}$ iff

- $\exists$ *order preserving bijection* $h : \mathsf{dom}(\mathbf{d}) \to \mathsf{dom}(\mathbf{g})$ s.t.
- $\forall x \in \mathsf{dom}(\mathbf{d}) : \mathbf{d}(x) = \mathbf{g}(h(x))$.

Comparison

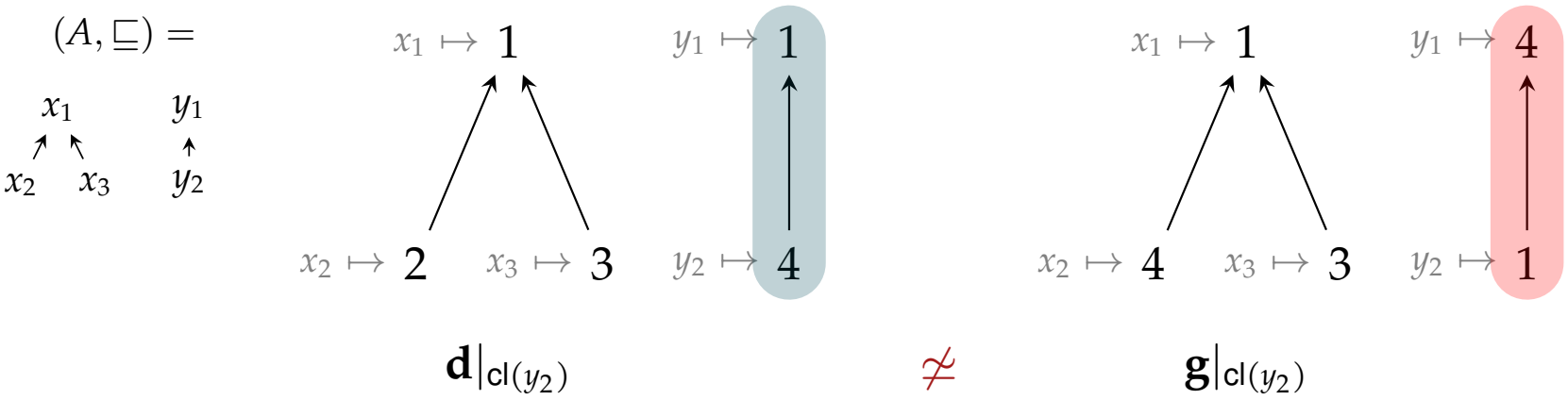


Definition (Valuation Equivalence)

Let $\mathbf{d} \simeq \mathbf{g}$ iff

- $\exists$ *order preserving bijection* $h : \mathsf{dom}(\mathbf{d}) \to \mathsf{dom}(\mathbf{g})$ s.t.
- $\forall x \in \mathsf{dom}(\mathbf{d}) : \mathbf{d}(x) = \mathbf{g}(h(x))$.

Comparison



Definition (Valuation Equivalence)

Let $\mathbf{d} \simeq \mathbf{g}$ iff

- $\exists$ *order preserving bijection* $h : \mathsf{dom}(\mathbf{d}) \to \mathsf{dom}(\mathbf{g})$ s.t.
- $\forall x \in \mathsf{dom}(\mathbf{d}) : \mathbf{d}(x) = \mathbf{g}(h(x))$.

$\mathbf{LTL}^{\downarrow}_{qo}$

Syntax

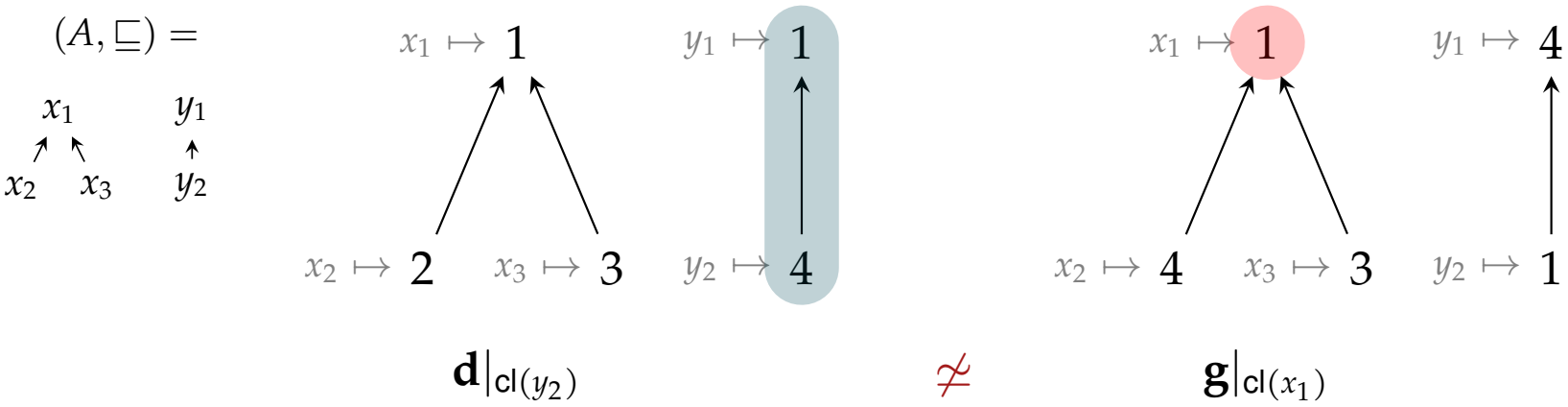
- Boolean + temporal operators (e.g., $\neg$, $\wedge$, X, U)
- Freeze operators: $\downarrow^x \varphi$, $\uparrow^x$

Semantics

$$w = (a_1, \mathbf{d}_1)(a_2, \mathbf{d}_2)\ldots(a_n, \mathbf{d}_n) \in (\Sigma \times \Delta^A)^+$$

$$\begin{array}{lcll} (w, i, \mathbf{h}) & \models & \downarrow^x \varphi & :\Leftrightarrow \quad (w, i, \mathbf{d}_i|_{\mathsf{cl}(x)}) \models \varphi \\ (w, i, \mathbf{g}) & \models & \uparrow^x & :\Leftrightarrow \quad \exists_{y \in A} : \mathbf{d}_i|_{\mathsf{cl}(x)} \simeq \mathbf{g}|_{\mathsf{cl}(y)} \end{array}$$

Comparison

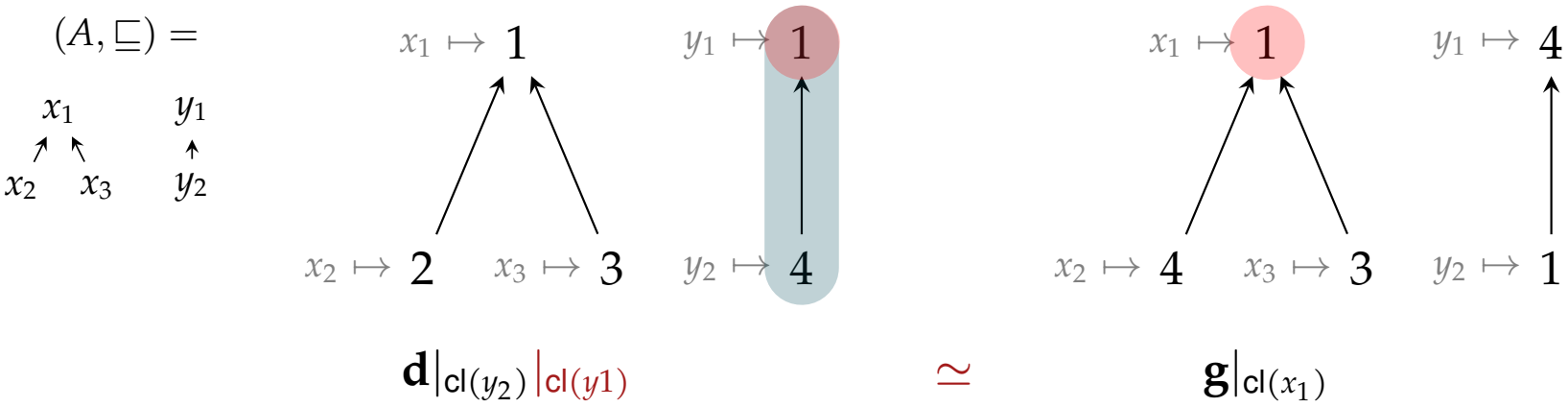


Definition (Valuation Equivalence)

Let $\mathbf{d} \simeq \mathbf{g}$ iff

- $\exists$ *order preserving bijection* $h : \mathsf{dom}(\mathbf{d}) \to \mathsf{dom}(\mathbf{g})$ s.t.
- $\forall x \in \mathsf{dom}(\mathbf{d}) : \mathbf{d}(x) = \mathbf{g}(h(x))$.

Comparison



Definition (Valuation Equivalence)

Let $\mathbf{d} \simeq \mathbf{g}$ iff

- $\exists$ *order preserving bijection* $h : \mathsf{dom}(\mathbf{d}) \to \mathsf{dom}(\mathbf{g})$ s.t.
- $\forall x \in \mathsf{dom}(\mathbf{d}) : \mathbf{d}(x) = \mathbf{g}(h(x))$.

$\mathbf{LTL}^{\downarrow}_{qo}$

Syntax

- Boolean + temporal operators (e.g., $\neg$, $\wedge$, X, U)
- Freeze operators: $\downarrow^x \varphi$, $\uparrow^x$

Semantics

$$w = (a_1, \mathbf{d}_1)(a_2, \mathbf{d}_2)\ldots(a_n, \mathbf{d}_n) \in (\Sigma \times \Delta^A)^+$$

$$\begin{array}{lcll} (w, i, \mathbf{h}) & \models & \downarrow^x \varphi & :\Leftrightarrow \quad (w, i, \mathbf{d}_i|_{\mathsf{cl}(x)}) \models \varphi \\ (w, i, \mathbf{g}) & \models & \uparrow^x & :\Leftrightarrow \quad \exists_{y \in A} : \mathbf{d}_i|_{\mathsf{cl}(x)} \simeq \mathbf{g}|_{\mathsf{cl}(y)} \end{array}$$

Undecidability

Theorem (LTL$^{\downarrow}_{qo}$ Satisfiability)

LTL$^{\downarrow}_{A}$ satisfiability is undecidable for $A = \{x \sqsubseteq z \sqsupseteq y\}$.

- Encode PCP Variant
- comparing sets $\{\mathbf{d}_i(x), \mathbf{d}_i(y)\}$ and $\{\mathbf{d}_j(x), \mathbf{d}_j(y)\}$ suffices

Tree-quasi-ordering

- quasi-ordering
- cl(x) is *totally* ordered for every $x \in A$

Undecidability

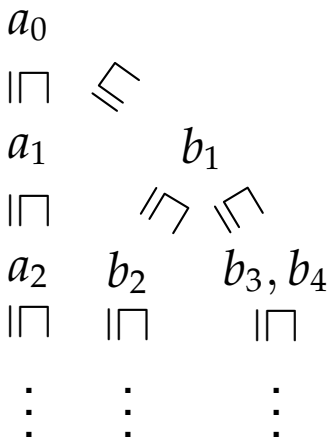
Theorem ($\mathrm{LTL}^{\downarrow}_{qo}$ Satisfiability)

$\mathrm{LTL}^{\downarrow}_{A}$ satisfiability is undecidable for $A = \{x \sqsubseteq z \sqsupseteq y\}$.

- Encode PCP Variant
- comparing sets $\{\mathbf{d}_i(x), \mathbf{d}_i(y)\}$ and $\{\mathbf{d}_j(x), \mathbf{d}_j(y)\}$ suffices

Tree-quasi-ordering

- quasi-ordering
- $\mathrm{cl}(x)$ is *totally* ordered for every $x \in A$



Undecidability

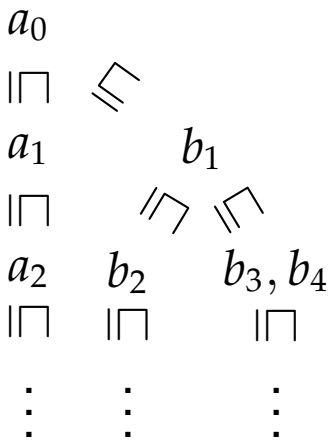
Theorem ($\mathrm{LTL}^{\downarrow}_{qo}$ Satisfiability)

$\mathrm{LTL}^{\downarrow}_{A}$ satisfiability is undecidable if A is not a tree-quasi-ordering.

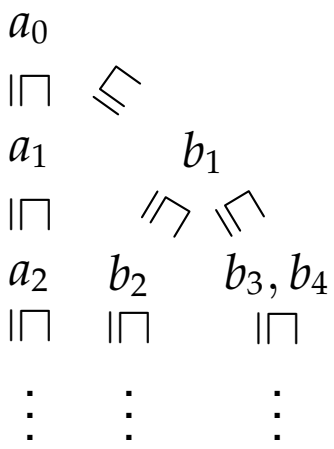
- Encode PCP Variant
- comparing sets $\{\mathbf{d}_i(x), \mathbf{d}_i(y)\}$ and $\{\mathbf{d}_j(x), \mathbf{d}_j(y)\}$ suffices

Tree-quasi-ordering

- quasi-ordering
- $\mathrm{cl}(x)$ is *totally* ordered for every $x \in A$



$\mathbf{LTL}^{\downarrow}_{tqo}$

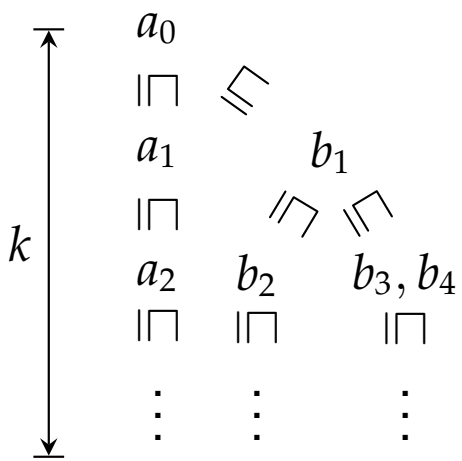


- Restrict $\mathrm{LTL}^{\downarrow}_{qo}$ to *tree-quasi-ordered* attributes
- Subsumes $\mathrm{LTL}^{\downarrow}_{1}$ [Demri, Lazić '06]

Towards a decision procedure

- Reduction to *linearly* ordered attributes $[k] = \{1, \ldots, k\} \subset \mathbb{N}$
- Upper bound: encoding into *nested counter systems (NCS)*
- Lower bound: encoding NCS into $\mathrm{LTL}^{\downarrow}_{tqo}$

$\mathbf{LTL}^{\downarrow}_{tqo}$



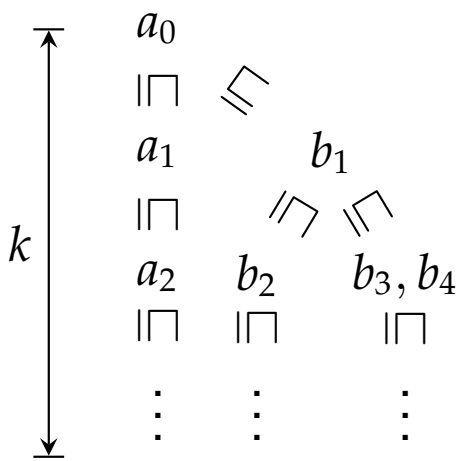
- Restrict $\mathrm{LTL}^{\downarrow}_{qo}$ to *tree-quasi-ordered* attributes
- Subsumes $\mathrm{LTL}^{\downarrow}_{1}$ [Demri, Lazić '06]

Towards a decision procedure

- Reduction to *linearly* ordered attributes $[k] = \{1, \ldots, k\} \subset \mathbb{N}$
- Upper bound: encoding into *nested counter systems (NCS)*
- Hardness: specifying covering runs of NCS

$$\textsc{Sat}(\mathrm{LTL}^{\downarrow}_{A}) \leq \textsc{Sat}(\mathrm{LTL}^{\downarrow}_{[k]})$$

$\mathbf{LTL}^{\downarrow}_{tqo}$



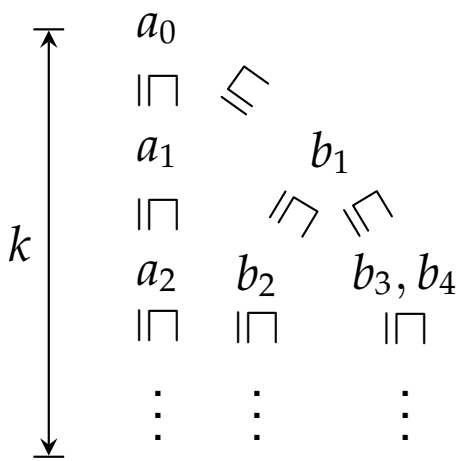
- Restrict $\mathrm{LTL}^{\downarrow}_{qo}$ to *tree-quasi-ordered* attributes
- Subsumes $\mathrm{LTL}^{\downarrow}_{1}$ [Demri, Lazić '06]

Towards a decision procedure

- Reduction to *linearly* ordered attributes $[k] = \{1, \ldots, k\} \subset \mathbb{N}$
- Upper bound: encoding into *nested counter systems (NCS)*
- Hardness: specifying covering runs of NCS

$$\textsc{Sat}(\mathrm{LTL}^{\downarrow}_{A}) \leq \textsc{Sat}(\mathrm{LTL}^{\downarrow}_{[k]}) \leq \textsc{Cov}((k+1)\text{-NCS})$$

$\mathbf{LTL}^{\downarrow}_{tqo}$



- Restrict $\mathrm{LTL}^{\downarrow}_{qo}$ to *tree-quasi-ordered* attributes
- Subsumes $\mathrm{LTL}^{\downarrow}_{1}$ [Demri, Lazić '06]

Towards a decision procedure

- Reduction to *linearly* ordered attributes $[k] = \{1, \ldots, k\} \subset \mathbb{N}$
- Upper bound: encoding into *nested counter systems (NCS)*
- Hardness: specifying covering runs of NCS

$$\textsc{Sat}(\mathrm{LTL}^{\downarrow}_{A}) \leq \textsc{Sat}(\mathrm{LTL}^{\downarrow}_{[k]}) \leq \textsc{Cov}((k+1)\text{-NCS}) \leq \textsc{Sat}(\mathrm{LTL}^{\downarrow}_{[k+1]})$$

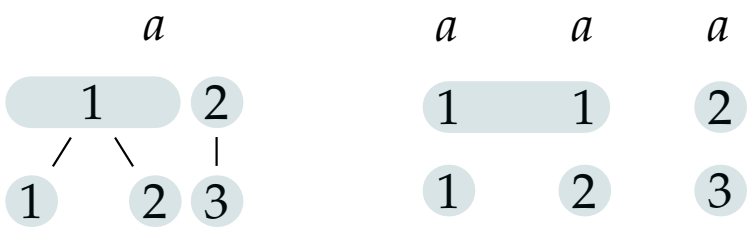
Linearisation

Theorem

Let $(A, \sqsubseteq)$ *be a tree-quasi-ordering of depth* k*. Then*

$$\textsc{Sat}(\mathit{LTL}_A^{\downarrow}) \leq \textsc{Sat}(\mathit{LTL}_{[k]}^{\downarrow}).$$

- Eliminate SCCs from A.
- Frame encoding



Counter Systems

- Configuration: Control State, Valuation of Counters

$$(q, \{x \mapsto 2, y \mapsto 3\})$$

- Can be seen as multiset

$$(q, \{x, x, y, y, y\})$$

- Transition rule: change control state and counter

$$(q, x)\delta(q', y)$$

Counter Systems

- Configuration: Control State, Valuation of Counters

$$(q, \{x \mapsto 2, y \mapsto 3\})$$

- Can be seen as multiset

$$(q, \{x, x, y, y, y\})$$

- Transition rule: change control state and counter

$$(q, x)\delta(q', y)$$

Counter Systems

- Configuration: Control State, Valuation of Counters

$$(q, \{x \mapsto 2, y \mapsto 3\})$$

- Can be seen as multiset

$$(q, \{x, x, y, y, y\})$$

- Transition rule: change control state and counter

$$(q, x)\delta(q', y)$$

Counter Systems

- Configuration: Control State, Valuation of Counters

$$(q, \{x \mapsto 2, y \mapsto 3\})$$

- Can be seen as multiset

$$(q, \{x, x, y, y, y\})$$

- Transition rule: change control state and counter

$$(q, x)\delta(q', y)$$

$$(q, \{x, x, y, y, y\}) \rightarrow (q', \{x, y, y, y, y\})$$

Counter Systems

- Configuration: Control State, Valuation of Counters

$$(q, \{x \mapsto 2, y \mapsto 3\})$$

- Can be seen as multiset

$$(q, \{x, x, y, y, y\})$$

- Transition rule: change control state and counter

$$(q, x)\delta(q', y)$$

$$(q, \{x, x, y, y, y\}) \rightarrow (q', \{x, y, y, y, y\})$$

Nested Counter Systems

- Configuration: nested multisets

$$\left(q, \{\left(x, \{a, a, b\}\right), \left(x, \{a, a, b\}\right), \left(y, \{a, b, c\}\right)\}\right)$$

- Transition rule: change "path"

$$(q, x, a)\delta(q', y, b)$$

Nested Counter Systems

- Configuration: nested multisets

$$\left(q, \{\left(x, \{a, a, b\}\right), \left(x, \{a, a, b\}\right), \left(y, \{a, b, c\}\right)\}\right)$$

- Transition rule: change "path"

$$(q, x, a)\delta(q', y, b)$$

Nested Counter Systems

- Configuration: nested multisets

$$\left(q, \{\left(x, \{a, a, b\}\right), \left(x, \{a, a, b\}\right), \left(y, \{a, b, c\}\right)\}\right)$$

- Transition rule: change "path"

$$(q, x, a)\delta(q', y, b)$$

$$\left(q, \{\left(x, \{a, a, b\}\right), \left(x, \{a, a, b\}\right), \left(y, \{a, b, c\}\right)\}\right)$$

Nested Counter Systems

- Configuration: nested multisets

$$\left(q, \{\left(x, \{a, a, b\}\right), \left(x, \{a, a, b\}\right), \left(y, \{a, b, c\}\right)\}\right)$$

- Transition rule: change "path"

$$(q, x, a)\delta(q', y, b)$$

$$\left(q, \{\left(x, \{a, a, b\}\right), \left(x, \{a, a, b\}\right), \left(y, \{a, b, c\}\right)\}\right)$$

Nested Counter Systems

- Configuration: nested multisets

$$\Big(q, \{\big(x, \{a, a, b\}\big), \big(x, \{a, a, b\}\big), \big(y, \{a, b, c\}\big)\}\Big)$$

- Transition rule: change "path"

$$(q, x, a)\delta(q', y, b)$$

$$\Big(q, \{\big(x, \{a, a, b\}\big), \big(x, \{a, a, b\}\big), \big(y, \{a, b, c\}\big)\}\Big)$$
$$\rightarrow \Big(q', \{\big(x, \{a, a, b\}\big), \big(y, \{a, b, b\}\big), \big(y, \{a, b, c\}\big)\}\Big)$$

Nested Counter Systems

- Configuration: nested multisets

$$\Big(q, \{\big(x, \{a, a, b\}\big), \big(x, \{a, a, b\}\big), \big(y, \{a, b, c\}\big)\}\Big)$$

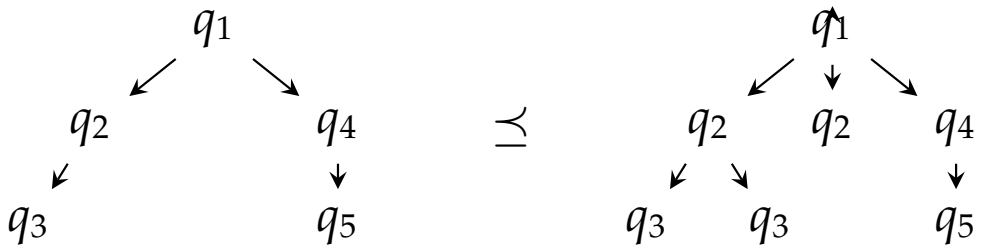
- Transition rule: change "path"

$$(q, x, a)\delta(q', y, b)$$

$$\Big(q, \{\big(x, \{a, a, b\}\big), \big(x, \{a, a, b\}\big), \big(y, \{a, b, c\}\big)\}\Big)$$
$$\rightarrow \Big(q', \{\big(x, \{a, a, b\}\big), \big(y, \{a, b, b\}\big), \big(y, \{a, b, c\}\big)\}\Big)$$

Coverability

- Nested multiset ordering on NCS configurations $\preceq \subseteq \mathcal{C}_{\mathcal{N}} \times \mathcal{C}_{\mathcal{N}}$



- C_1 coverable from C_0 iff

  $\exists_{C'} : C_0 \rightarrow^* C' \succeq C_0$

$\mathrm{LTL}^{\downarrow}_{[k]}$ to NCS

- $\mathrm{LTL}^{\downarrow}_{[k]}$ formula $\Phi \Rightarrow$ NCS $\mathcal{N}_\Phi$, configurations C_0, C_1
- C_1 coverable from C_0 iff Φ satisfiable

Construction

- $\mathcal{N}_\Phi$ maintains $(a, \mathbf{g}, \mathrm{G}) \in \Sigma \times \Delta^k \times 2^{\mathrm{sub}(\Phi)}$ comprising
  - letter, valuation $(a, \mathbf{g}) \in \Sigma \times \Delta^k$
  - guarantees $\mathrm{G} = \{(\varphi_1, \mathbf{d}_1), \ldots, (\varphi_n, \mathbf{d}_n)\} \subseteq \mathrm{sub}(\Phi) \times \Delta^k$
- Invariant: $\exists_{\mathrm{w}} \forall_{(\varphi, \mathbf{d}) \in \mathrm{G}} : \left((a, \mathbf{g}) \mathrm{w}, \mathbf{1}, \mathbf{d} \right) \models \varphi$
- Check if $(a_0, \mathbf{g}_0, \emptyset) \to^* (a_n, \mathbf{g}_n, \{(\varphi, \mathbf{d}), \ldots\})$

$\mathbf{LTL}^{\downarrow}_{[k]}$ to NCS

- $\mathsf{LTL}^{\downarrow}_{[k]}$ formula $\Phi \Rightarrow$ NCS $\mathcal{N}_\Phi$, configurations C_0, C_1
- C_1 coverable from C_0 iff Φ satisfiable

Construction

- $\mathcal{N}_\Phi$ maintains $(a, \mathbf{g}, G) \in \Sigma \times \Delta^k \times 2^{\mathsf{sub}(\Phi)}$ comprising
  - letter, valuation $(a, \mathbf{g}) \in \Sigma \times \Delta^k$
  - guarantees $G = \{(\varphi_1, \mathbf{d}_1), \ldots, (\varphi_n, \mathbf{d}_n)\} \subseteq \mathsf{sub}(\Phi) \times \Delta^k$
- Invariant: $\exists_w \forall_{(\varphi, \mathbf{d}) \in G} : ((a, \mathbf{g})w, 1, \mathbf{d}) \models \varphi$
- Check if $(a_0, \mathbf{g}_0, \emptyset) \to^* (a_m, \mathbf{g}_m, \{(\Phi, \mathbf{d}), \ldots\})$

$\mathbf{LTL}^{\downarrow}_{[k]}$ to NCS

- $\mathsf{LTL}^{\downarrow}_{[k]}$ formula $\Phi \Rightarrow$ NCS $\mathcal{N}_\Phi$, configurations C_0, C_1
- C_1 coverable from C_0 iff Φ satisfiable

Construction

- $\mathcal{N}_\Phi$ maintains $(a, \mathbf{g}, G) \in \Sigma \times \Delta^k \times 2^{\mathsf{sub}(\Phi)}$ comprising
  - letter, valuation $(a, \mathbf{g}) \in \Sigma \times \Delta^k$
  - guarantees $G = \{(\varphi_1, \mathbf{d}_1), \ldots, (\varphi_n, \mathbf{d}_n)\} \subseteq \mathsf{sub}(\Phi) \times \Delta^k$
- Invariant: $\exists_w \forall_{(\varphi, \mathbf{d}) \in G} : ((a, \mathbf{g})w, 1, \mathbf{d}) \models \varphi$
- Check if $(a_0, \mathbf{g}_0, \emptyset) \rightarrow^* (a_m, \mathbf{g}_m, \{(\Phi, \mathbf{d}), \ldots\})$

$\mathbf{LTL}^{\downarrow}_{[k]}$ to NCS

- $\mathsf{LTL}^{\downarrow}_{[k]}$ formula $\Phi \Rightarrow$ NCS $\mathcal{N}_\Phi$, configurations C_0, C_1
- C_1 coverable from C_0 iff Φ satisfiable

Construction

- $\mathcal{N}_\Phi$ maintains $(a, \mathbf{g}, G) \in \Sigma \times \Delta^k \times 2^{\mathsf{sub}(\Phi)}$ comprising
  - letter, valuation $(a, \mathbf{g}) \in \Sigma \times \Delta^k$
  - guarantees $G = \{(\varphi_1, \mathbf{d}_1), \ldots, (\varphi_n, \mathbf{d}_n)\} \subseteq \mathsf{sub}(\Phi) \times \Delta^k$
- Invariant: $\exists_w \forall_{(\varphi, \mathbf{d}) \in G} : ((a, \mathbf{g})w, 1, \mathbf{d}) \models \varphi$
- Check if $(a_0, \mathbf{g}_0, \emptyset) \rightarrow^* (a_m, \mathbf{g}_m, \{(\Phi, \mathbf{d}), \ldots\})$

$\mathbf{LTL}^{\downarrow}_{[k]}$ to NCS

Example (1)

$$\Phi = \mathrm{G}(\neg b \vee \mathrm{X}\, a)$$

- $\mathsf{sub}(\Phi) = \{$

  $\mathrm{G}(\neg b \vee \mathrm{X}\, a), \quad (\neg b \vee \mathrm{X}\, a) \wedge \overline{\mathrm{X}}\, \mathrm{G}(\neg b \vee \mathrm{X}\, a),$

  $\neg b \vee \mathrm{X}\, a, \quad \neg b, \quad b, \quad \mathrm{X}\, a, \quad a$

  $\}$

Example $\Phi = \mathrm{G}(\neg b \vee \mathrm{X}\, a)$

$$\Big(a, \mathbf{d}, \quad \emptyset \quad \Big)$$

$$\downarrow \qquad \overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a) \in \mathsf{sub}(\Phi), \ \text{(init)}$$

$$\Big(a, \mathbf{d}, \quad \{(\overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a), \mathbf{g})\} \quad \Big)$$

Example $\Phi = \mathrm{G}(\neg b \vee \mathrm{X}\, a)$

$$\Big(a, \mathbf{d}, \quad \emptyset \quad \Big)$$

$$\downarrow \qquad \overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a) \in \mathsf{sub}(\Phi), \ \text{(init)}$$

$$\Big(a, \mathbf{d}, \quad \{(\overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a), \mathbf{g})\} \quad \Big)$$

$$\downarrow \qquad a \neq b \in \Sigma, \ \neg b \in \mathsf{sub}(\Phi)$$

$$\Big(a, \mathbf{d}, \quad \{(\overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a), \mathbf{g}), (\neg b, \mathbf{g})\} \quad \Big)$$

Example $\Phi = \mathrm{G}(\neg b \vee \mathrm{X}\, a)$

$$\Big(a, \mathbf{d}, \quad \emptyset \quad \Big)$$

$$\downarrow \qquad \overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a) \in \mathsf{sub}(\Phi), \ \text{(init)}$$

$$\Big(a, \mathbf{d}, \quad \{(\overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a), \mathbf{g})\} \quad \Big)$$

$$\downarrow \qquad a \neq b \in \Sigma, \ \neg b \in \mathsf{sub}(\Phi)$$

$$\Big(a, \mathbf{d}, \quad \{(\overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a), \mathbf{g}), (\neg b, \mathbf{g})\} \quad \Big)$$

$$\downarrow \qquad (\varphi, \mathbf{g}) \in G, \varphi \vee \psi \in \mathsf{sub}(\Phi)$$

$$\Big(a, \mathbf{d}, \{(\overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a), \mathbf{g}), (\neg b \vee \mathrm{X}\, a, \mathbf{g})\}\Big)$$

Example $\Phi = \mathrm{G}(\neg b \vee \mathrm{X}\, a)$

$$\Big(a, \mathbf{d},\ \{(\overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a), \mathbf{g}),\ (\neg b \vee \mathrm{X}\, a, \mathbf{g})\}\ \Big)$$

$$\downarrow$$

$(\varphi, \mathbf{g}), (\psi, \mathbf{g}) \in G,$

$\varphi \wedge \psi \in \mathsf{sub}(\Phi)$

$$\Big(a, \mathbf{d}, \left\{ \begin{array}{l} (\overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a), \mathbf{g}),\ (\neg b \vee \mathrm{X}\, a, \mathbf{g}), \\ \quad (\overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a) \wedge (\neg b \vee \mathrm{X}\, a), \mathbf{g}) \end{array} \right\} \Big)$$

$$\downarrow$$

$(\varphi \wedge \overline{\mathrm{X}}\,\mathrm{G}(\varphi), \mathbf{g}) \in G,$

$\mathrm{G}(\varphi) \in \mathsf{sub}(\Phi)$

$$\Big(a, \mathbf{d}, \left\{ \begin{array}{l} (\overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a), \mathbf{g}),\ (\neg b \vee \mathrm{X}\, a, \mathbf{g}), \\ (\overline{\mathrm{X}}\,\mathrm{G}(\neg b \vee \mathrm{X}\, a) \wedge (\neg b \vee \mathrm{X}\, a), \mathbf{g}), \\ (\mathrm{G}(\neg b \vee \mathrm{X}\, a), \mathbf{g}) \end{array} \right\} \Big)$$

$\mathbf{LTL}^{\downarrow}_{[k]}$ to NCS

Example (2)

$$\Psi = \downarrow^2 (\mathrm{X} \neg \uparrow^1 \wedge \mathrm{X}\mathrm{X} \uparrow^2)$$

- $\mathsf{sub}(\Psi) = \{\downarrow^2 (\mathrm{X} \neg \uparrow^1 \wedge \mathrm{X}\mathrm{X} \uparrow^2), \quad \mathrm{X} \neg \uparrow^1 \wedge \mathrm{X}\mathrm{X} \uparrow^2, \quad \neg \uparrow^1, \ldots\}$

Example $\Psi = \downarrow^2 (\mathrm{X} \neg \uparrow^1 \wedge \mathrm{X}\,\mathrm{X} \uparrow^2)$

$$\Big(a, \langle 1,2,3\rangle, \qquad \emptyset \qquad \Big)$$

$$\downarrow$$

$$\Big(a, \langle 1,2,3\rangle, \ \{(\uparrow^2, \langle 1,2\rangle)\} \ \Big)$$

$\uparrow^2 \in \mathsf{sub}(\Psi)$,

$((a, \langle 1,2,3\rangle)w, 1, \langle 1,2\rangle) \models \uparrow^2$

Example $\Psi = \downarrow^2 (\mathrm{X} \neg \uparrow^1 \wedge \mathrm{X}\,\mathrm{X} \uparrow^2)$

$$\Big(a, \langle 1,2,3\rangle, \quad \emptyset \quad \Big)$$

$\uparrow^2 \in \mathsf{sub}(\Psi)$,

$$\downarrow$$

$((a, \langle 1,2,3\rangle)w, 1, \langle 1,2\rangle) \models \uparrow^2$

$$\Big(a, \langle 1,2,3\rangle, \ \{(\uparrow^2, \langle 1,2\rangle)\} \ \Big)$$

$\uparrow^2 \in G, \ \mathrm{X} \uparrow^2 \in \mathsf{sub}(\Psi)$,

$$\downarrow_*$$

"guess" $(b, \langle 2,2,3\rangle) \in \Sigma \times \Delta^k$

$$\Big(b, \langle 2,2,3\rangle, \{(\mathrm{X} \uparrow^2, \langle 1,2\rangle)\}\Big)$$

Example $\Psi = \downarrow^2 (\mathrm{X} \neg \uparrow^1 \wedge \mathrm{X}\,\mathrm{X} \uparrow^2)$

$$\Big(b, \langle 2,2,3\rangle, \quad \{(\mathrm{X} \uparrow^2, \langle 1,2\rangle)\} \quad \Big)$$

$$\downarrow$$

$$\Big(b, \langle 2,2,3\rangle, \quad \{(\mathrm{X} \uparrow^2, \langle 1,2\rangle), (\neg \uparrow^1, \langle 1,2\rangle)\} \quad \Big)$$

$\uparrow^1 \in \mathsf{sub}(\Psi)$,

$((b, \langle 2,2,3\rangle)w, 1, \langle 1,2\rangle) \models \neg \uparrow^1$

Example $\Psi = \downarrow^2 (\mathrm{X} \neg \uparrow^1 \wedge \mathrm{X}\,\mathrm{X} \uparrow^2)$

$$\Big(b, \langle 2,2,3\rangle, \quad \{(\mathrm{X} \uparrow^2, \langle 1,2\rangle)\} \quad \Big)$$

$\uparrow^1 \in \mathsf{sub}(\Psi)$,

$$\downarrow$$

$((b, \langle 2,2,3\rangle)w, 1, \langle 1,2\rangle) \models \neg \uparrow^1$

$$\Big(b, \langle 2,2,3\rangle, \quad \{(\mathrm{X} \uparrow^2, \langle 1,2\rangle), (\neg \uparrow^1, \langle 1,2\rangle)\} \quad \Big)$$

$$\downarrow_*$$

"guess" $(a, \langle 1,2,1\rangle) \in \Sigma \times \Delta^k$

$$\Big(a, \langle 1,2,1\rangle, \{(\mathrm{X}\,\mathrm{X} \uparrow^2, \langle 1,2\rangle), (\mathrm{X} \neg \uparrow^1, \langle 1,2\rangle)\}\Big)$$

Example $\Psi = \downarrow^2 (\mathrm{X} \neg \uparrow^1 \wedge \mathrm{X}\,\mathrm{X} \uparrow^2)$

$$\Big(b, \langle 2,2,3\rangle, \quad \{(\mathrm{X} \uparrow^2, \langle 1,2\rangle)\} \quad \Big)$$

$$\downarrow$$

$$\Big(b, \langle 2,2,3\rangle, \quad \{(\mathrm{X} \uparrow^2, \langle 1,2\rangle), (\neg \uparrow^1, \langle 1,2\rangle)\} \quad \Big)$$

$$\downarrow_*$$

$$\Big(a, \langle 1,2,1\rangle, \{(\mathrm{X}\,\mathrm{X} \uparrow^2, \langle 1,2\rangle), (\mathrm{X} \neg \uparrow^1, \langle 1,2\rangle)\}\Big)$$

$$\downarrow$$

$$\Big(a, \langle 1,2,1\rangle, \ \{\ldots, (\mathrm{X}\,\mathrm{X} \uparrow^2 \wedge \mathrm{X} \neg \uparrow^1, \langle 1,2\rangle)\} \ \Big)$$

$\uparrow^1 \in \mathsf{sub}(\Psi)$,

$((b, \langle 2,2,3\rangle)w, 1, \langle 1,2\rangle) \models \neg \uparrow^1$

"guess" $(a, \langle 1,2,1\rangle) \in \Sigma \times \Delta^k$

Example $\Psi = \downarrow^2 (\mathrm{X} \neg \uparrow^1 \wedge \mathrm{X}\,\mathrm{X} \uparrow^2)$

$$\Big(a, \langle 1,2,1\rangle, \quad \{\ldots, (\mathrm{X}\,\mathrm{X} \uparrow^2 \wedge \mathrm{X} \neg \uparrow^1, \langle 1,2\rangle)\}\Big)$$

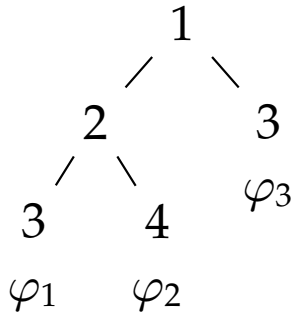
$$\downarrow \qquad \begin{array}{l} ((a, \langle 1,2,1\rangle)w, 1, \langle 1,2\rangle) \models \quad \mathrm{X}\,\mathrm{X} \uparrow^2 \wedge \mathrm{X} \neg \uparrow^1 \\ \qquad\qquad\qquad\qquad\qquad\qquad \Leftrightarrow \\ ((a, \langle 1,2,1\rangle)w, 1, \quad \mathbf{x} \quad) \models \downarrow^2 (\mathrm{X}\,\mathrm{X} \uparrow^2 \wedge \mathrm{X} \neg \uparrow^1) \end{array}$$

$$\Big(a, \langle 1,2,1\rangle, \quad \{\ldots, (\downarrow^2 (\mathrm{X}\,\mathrm{X} \uparrow^2 \wedge \mathrm{X} \neg \uparrow^1), \langle 3,4,5\rangle)\}\Big)$$

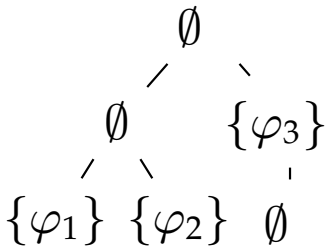
Encoding

Store $G = \{(\varphi_1, \mathbf{d}_1), \ldots, (\varphi_n, \mathbf{d}_n)\}$ modulo "nested permutation":

$$\left\{ \underset{\varphi_1}{\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}}, \underset{\varphi_2}{\begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}}, \underset{\varphi_3}{\begin{pmatrix} 1 \\ 3 \end{pmatrix}} \right\} \Rightarrow$$



$\Rightarrow$

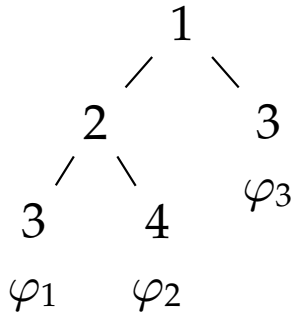


$(a, \langle 1, 3, 4 \rangle)$

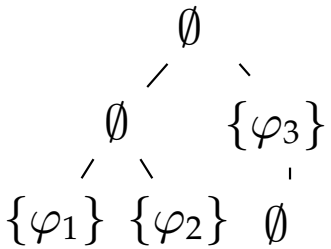
Encoding

Store $G = \{(\varphi_1, \mathbf{d}_1), \ldots, (\varphi_n, \mathbf{d}_n)\}$ modulo "nested permutation":

$$\left\{ \underset{\varphi_1}{\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}}, \underset{\varphi_2}{\begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}}, \underset{\varphi_3}{\begin{pmatrix} 1 \\ 3 \end{pmatrix}} \right\} \Rightarrow$$



$\Rightarrow$

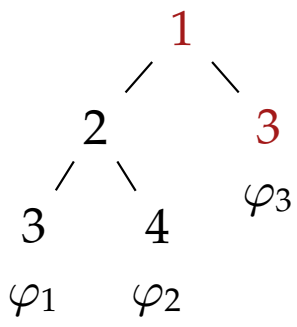


$(a, \langle 1, 3, 4 \rangle)$

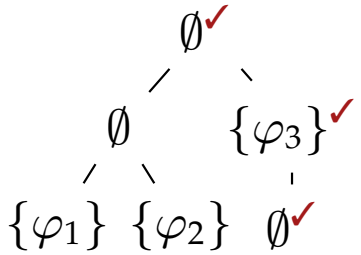
Encoding

Store $G = \{(\varphi_1, \mathbf{d}_1), \ldots, (\varphi_n, \mathbf{d}_n)\}$ modulo "nested permutation":

$$\left\{ \underset{\varphi_1}{\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}}, \underset{\varphi_2}{\begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}}, \underset{\varphi_3}{\begin{pmatrix} 1 \\ 3 \end{pmatrix}} \right\} \Rightarrow$$



$\Rightarrow$



$(a, \langle 1, 3, 4 \rangle)$

$\mathbf{LTL}^{\downarrow}_{[k]}$ to NCS

Remarks

- Find configuration containing Φ ($\Rightarrow$ coverability)
- Synchronous processing of guarantees $\varphi_1, \ldots, \varphi_n \Rightarrow \mathrm{X}\,\varphi_1, \ldots, \mathrm{X}\,\varphi_n$
  - no atomic traversal in NCS
  - partial copy
  - $\Rightarrow$ backward approach
  - $\Rightarrow$ additional level k + 1

$\mathbf{LTL}^{\downarrow}_{[k]}$ to NCS

Remarks

- Find configuration containing Φ ($\Rightarrow$ coverability)
- Synchronous processing of guarantees $\varphi_1, \ldots, \varphi_n \Rightarrow \mathrm{X}\,\varphi_1, \ldots, \mathrm{X}\,\varphi_n$
  - no atomic traversal in NCS
  - partial copy

  $\Rightarrow$ backward approach

  $\Rightarrow$ additional level $k+1$

$\mathbf{LTL}^{\downarrow}_{[k]}$ to NCS

Remarks

- Find configuration containing Φ ($\Rightarrow$ coverability)
- Synchronous processing of guarantees $\varphi_1, \ldots, \varphi_n \Rightarrow \mathrm{X}\,\varphi_1, \ldots, \mathrm{X}\,\varphi_n$
  - no atomic traversal in NCS
  - partial copy

  $\Rightarrow$ backward approach

  $\Rightarrow$ additional level $k+1$

$\mathbf{LTL}^{\downarrow}_{[k]}$ to NCS

Remarks

- Find configuration containing Φ ($\Rightarrow$ coverability)
- Synchronous processing of guarantees $\varphi_1, \ldots, \varphi_n \Rightarrow \mathrm{X}\,\varphi_1, \ldots, \mathrm{X}\,\varphi_n$
  - no atomic traversal in NCS
  - partial copy

  $\Rightarrow$ backward approach

  $\Rightarrow$ additional level $k+1$

$\mathbf{LTL}^{\downarrow}_{[k]}$ to NCS

Remarks

- Find configuration containing Φ ($\Rightarrow$ coverability)
- Synchronous processing of guarantees $\varphi_1, \ldots, \varphi_n \Rightarrow \mathrm{X}\,\varphi_1, \ldots, \mathrm{X}\,\varphi_n$
  - no atomic traversal in NCS
  - partial copy

  $\Rightarrow$ backward approach

  $\Rightarrow$ additional level $k+1$

$\mathbf{LTL}^{\downarrow}_{[k]}$ to NCS

Remarks

- Find configuration containing Φ ($\Rightarrow$ coverability)
- Synchronous processing of guarantees $\varphi_1, \ldots, \varphi_n \Rightarrow \mathrm{X}\,\varphi_1, \ldots, \mathrm{X}\,\varphi_n$
  - no atomic traversal in NCS
  - partial copy

  $\Rightarrow$ backward approach

  $\Rightarrow$ additional level $k+1$

$\mathrm{LTL}^{\downarrow}_{[k]}$ to NCS

Remarks

- Find configuration containing Φ ($\Rightarrow$ coverability)
- Synchronous processing of guarantees $\varphi_1, \ldots, \varphi_n \Rightarrow \mathrm{X}\,\varphi_1, \ldots, \mathrm{X}\,\varphi_n$
  - no atomic traversal in NCS
  - partial copy

  $\Rightarrow$ backward approach

  $\Rightarrow$ additional level $k+1$

Theorem

For tree-quasi-ordered attribute sets A of depth k

$$\textsc{Sat}(\mathit{LTL}^{\downarrow}_{A}) \leq_{\textsc{exp}} \textsc{Cov}((k+1)\text{-}\mathit{NCS})$$

k-NCS to $\mathrm{LTL}^{\downarrow}_{[k]}$

Theorem

$$\textsc{Cov}(k\text{-}NCS) \leq_{\textsc{exp}} \textsc{Sat}(LTL^{\downarrow}_{[k]})$$

- Frame encoding for sequences of configurations
  - hierarchical chaining
  - expressing transition effects by formulae
- Future-time modalities vs. coverability (“lossy” reachability)
  - specifying reversed, lossy runs

Summary

Freeze LTL

- Generalised storing mechanism for collections of data values
- Dependency relation on attributes
- Specification and verification of dynamic systems

Decidable fragments

- Precise characterisation of maximal decidable fragment $\mathrm{LTL}^{\downarrow}_{tqo}$
- Syntactic and semantic extension $\mathrm{LTL}^{\downarrow}_{1}$

Methodology

- Close relation to nested counter systems
- Equivalence of satisfiability and coverability problems
- Carry over decidability and complexity results

Summary

Freeze LTL

- Generalised storing mechanism for collections of data values
- Dependency relation on attributes
- Specification and verification of dynamic systems

Decidable fragments

- Precise characterisation of maximal decidable fragment $\mathrm{LTL}^{\downarrow}_{tqo}$
- Syntactic and semantic extension $\mathrm{LTL}^{\downarrow}_{1}$

Methodology

- Close relation to nested counter systems
- Equivalence of satisfiability and coverability problems
- Carry over decidability and complexity results

Summary

Freeze LTL

- Generalised storing mechanism for collections of data values
- Dependency relation on attributes
- Specification and verification of dynamic systems

Decidable fragments

- Precise characterisation of maximal decidable fragment $\mathrm{LTL}^{\downarrow}_{tqo}$
- Syntactic and semantic extension $\mathrm{LTL}^{\downarrow}_{1}$

Methodology

- Close relation to nested counter systems
- Equivalence of satisfiability and coverability problems
- Carry over decidability and complexity results