Asynchronous Composition of Local Interface LTL Properties

Alberto Bombardelli Stefano Tonetta

Fondazione Bruno Kessler

Introduction

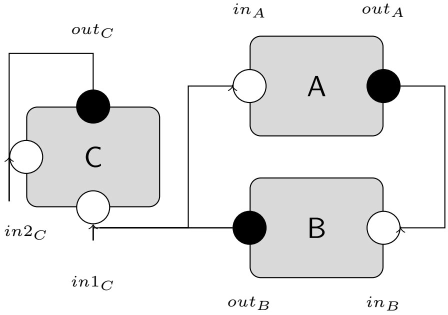
Introduction

Topic: Asynchronous composition of LTL properties

Introduction

Topic: Asynchronous composition of LTL properties

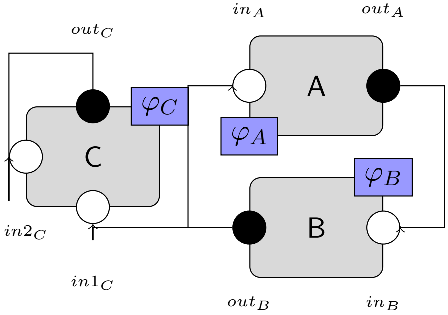
Type of composition		
Variable setup	Sync	Async
Local Vars		
I/O		X



Introduction

Topic: Asynchronous composition of LTL properties

Type of composition		
Variable setup	Sync	Async
Local Vars		
I/O		X

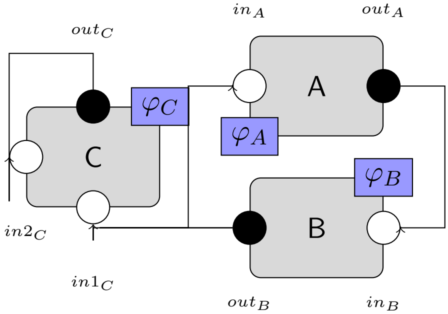


$$\underbrace{\gamma_P(\overbrace{\varphi_C, \varphi_B, \varphi_A}^{\text{Local props}})}_{\text{Async composition}} \rightarrow \overbrace{\varphi}^{\text{global prop}}$$

Introduction

Topic: Asynchronous composition of LTL properties

Type of composition		
Variable setup	Sync	Async
Local Vars		
I/O		X



$$\underbrace{\gamma_P(\overbrace{\varphi_C, \varphi_B, \varphi_A}^{\text{Local props}})}_{\text{Async composition}} \rightarrow \overbrace{\varphi}^{\text{global prop}}$$

Use case: Verification of contract refinement of asynchronously composed A/G LTL contracts

Iinterface Transition System and composition

$$\mathcal{M} = \langle \mathcal{V}^I, \mathcal{V}^O, \mathcal{V}^H, \mathcal{I}, \mathcal{T}, \mathcal{F} \rangle$$

- $\mathcal{V}^I$ (input vars)
- $\mathcal{V}^O$ (output vars)
- $\mathcal{V}^H$ (internal vars)
- $\mathcal{I}$ (init)
- $\mathcal{T}$ (transitions)
- $\mathcal{F}$ (fairness)

$$\mathcal{M} = \mathcal{M}_1 \otimes \mathcal{M}_2$$

- $\mathcal{V}^I = (\mathcal{V}_1^I \cup \mathcal{V}_2^I) \setminus Shared(\mathcal{M}_1, \mathcal{M}_2)$
- $\mathcal{V}^O = (\mathcal{V}_1^O \cup \mathcal{V}_2^O) \setminus Shared(\mathcal{M}_1, \mathcal{M}_2)$
- $\mathcal{V}^H =$ $\mathcal{V}_1^H \cup \mathcal{V}_2^H \cup Shared(\mathcal{M}_1, \mathcal{M}_2) \cup \{st_1, st_2\}$
- $\mathcal{I} = \mathcal{I}_1 \wedge \mathcal{I}_2$
- $\mathcal{T} = (\neg st_1 \rightarrow \mathcal{T}_1) \wedge$ $(\neg st_2 \rightarrow \mathcal{T}_2) \wedge$ $(st_1 \rightarrow \bigwedge_{v^o \in \mathcal{V}_1^O \cup \mathcal{V}_1^H} (v^{o\prime} = v^o)) \wedge$ $(st_2 \rightarrow \bigwedge_{v^o \in \mathcal{V}_2^O \cup \mathcal{V}_2^H} (v^{o\prime} = v^o))$
- $\mathcal{F} = \{\varphi_1 \wedge \neg st_1 | \varphi_1 \in \mathcal{F}_1\} \cup$ $\{\varphi_2 \wedge \neg st_2 | \varphi_2 \in \mathcal{F}_2\} \cup \{\neg st_1, \neg st_2\}$

Embedding local traces in global traces

Embedding local traces in global traces

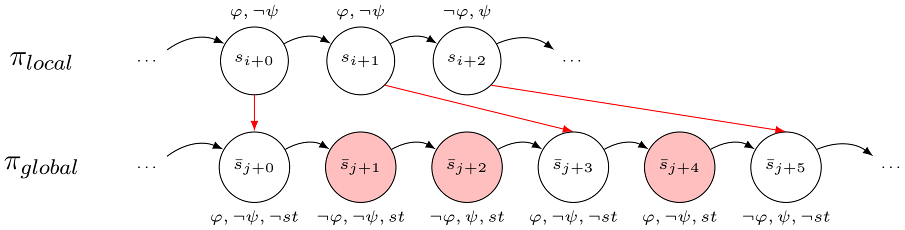
Local to global traces

- Traces are embedded in global traces
- States of the local trace are labelled with $\neg st$
- We assume to work with local infinite traces/run of local component ($GF\neg st$)

Embedding local traces in global traces

Local to global traces

- Traces are embedded in global traces
- States of the local trace are labelled with $\neg st$
- We assume to work with local infinite traces/run of local component ($GF\neg st$)



LTL rewriting

LTL rewriting

We defined a rewriting for LTL + past + FO

LTL rewriting

We defined a rewriting for LTL + past + FO

Property "Theorem 1"

$$\pi_{local} \models \varphi \Leftrightarrow \pi_{global} \models \mathcal{R}^*(\varphi)$$

LTL rewriting

We defined a rewriting for LTL + past + FO

Property "Theorem 1"

$$\pi_{local} \models \varphi \Leftrightarrow \pi_{global} \models \mathcal{R}^*(\varphi)$$

$\mathcal{R}$ it assumes to be on a state where $\neg st$ holds

LTL rewriting

We defined a rewriting for LTL + past + FO

Property "Theorem 1"

$$\pi_{local} \models \varphi \Leftrightarrow \pi_{global} \models \mathcal{R}^*(\varphi)$$

$\mathcal{R}$ it assumes to be on a state where $\neg st$ holds

- a: $\mathcal{R}(a) := a$
- $\neg$: $\mathcal{R}(\neg\varphi) := \neg\mathcal{R}(\varphi)$
- $\vee$: $\mathcal{R}(\varphi \vee \psi) := \mathcal{R}(\varphi) \vee \mathcal{R}(\psi)$

LTL rewriting

We defined a rewriting for LTL + past + FO

Property "Theorem 1"

$$\pi_{local} \models \varphi \Leftrightarrow \pi_{global} \models \mathcal{R}^*(\varphi)$$

$\mathcal{R}$ it assumes to be on a state where $\neg st$ holds

- a: $\mathcal{R}(a) := a$
- $\neg$: $\mathcal{R}(\neg\varphi) := \neg\mathcal{R}(\varphi)$
- $\vee$: $\mathcal{R}(\varphi \vee \psi) := \mathcal{R}(\varphi) \vee \mathcal{R}(\psi)$
- **X**: $\mathcal{R}(\mathbf{X}\varphi) := \mathbf{X}(\neg st \mathbf{R}(st \vee \mathcal{R}(\varphi)))$

LTL rewriting

We defined a rewriting for LTL + past + FO

Property "Theorem 1"

$$\pi_{local} \models \varphi \Leftrightarrow \pi_{global} \models \mathcal{R}^*(\varphi)$$

$\mathcal{R}$ — it assumes to be on a state where $\neg st$ holds

- a: $\mathcal{R}(a) := a$
- $\neg$: $\mathcal{R}(\neg\varphi) := \neg\mathcal{R}(\varphi)$
- $\vee$: $\mathcal{R}(\varphi \vee \psi) := \mathcal{R}(\varphi) \vee \mathcal{R}(\psi)$
- $\mathbf{X}$: $\mathcal{R}(\mathbf{X}\varphi) := \mathbf{X}(\neg st \mathbf{R}(st \vee \mathcal{R}(\varphi)))$
- $\mathbf{U}$: $\mathcal{R}(\varphi\mathbf{U}\psi) := (st \vee \mathcal{R}(\varphi))\mathbf{U}(\neg st \wedge \mathcal{R}(\psi))$

LTL rewriting

We defined a rewriting for LTL + past + FO

Property "Theorem 1"

$$\pi_{local} \models \varphi \Leftrightarrow \pi_{global} \models \mathcal{R}^*(\varphi)$$

$\mathcal{R}$ it assumes to be on a state where $\neg st$ holds

- a: $\mathcal{R}(a) := a$
- $\neg$: $\mathcal{R}(\neg\varphi) := \neg\mathcal{R}(\varphi)$
- $\vee$: $\mathcal{R}(\varphi \vee \psi) := \mathcal{R}(\varphi) \vee \mathcal{R}(\psi)$
- **X**: $\mathcal{R}(\mathbf{X}\varphi) := \mathbf{X}(\neg st \mathbf{R}(st \vee \mathcal{R}(\varphi)))$
- **U**: $\mathcal{R}(\varphi \mathbf{U} \psi) := (st \vee \mathcal{R}(\varphi))\mathbf{U}(\neg st \wedge \mathcal{R}(\psi))$
- **Y**: $\mathcal{R}(\mathbf{Y}\varphi) := \mathbf{Y}(st \mathbf{S}(\neg st \wedge \mathcal{R}(\varphi)))$

LTL rewriting

We defined a rewriting for LTL + past + FO

Property "Theorem 1"

$$\pi_{local} \models \varphi \Leftrightarrow \pi_{global} \models \mathcal{R}^*(\varphi)$$

$\mathcal{R}$ — it assumes to be on a state where $\neg st$ holds

- a: $\mathcal{R}(a) := a$
- $\neg$: $\mathcal{R}(\neg\varphi) := \neg\mathcal{R}(\varphi)$
- $\vee$: $\mathcal{R}(\varphi \vee \psi) := \mathcal{R}(\varphi) \vee \mathcal{R}(\psi)$
- **X**: $\mathcal{R}(\mathbf{X}\varphi) := \mathbf{X}(\neg st \mathbf{R}(st \vee \mathcal{R}(\varphi)))$
- **U**: $\mathcal{R}(\varphi \mathbf{U} \psi) := (st \vee \mathcal{R}(\varphi))\mathbf{U}(\neg st \wedge \mathcal{R}(\psi))$
- **Y**: $\mathcal{R}(\mathbf{Y}\varphi) := \mathbf{Y}(st \mathbf{S}(\neg st \wedge \mathcal{R}(\varphi)))$
- **S**: $\mathcal{R}(\varphi \mathbf{S} \psi) := (st \vee \mathcal{R}(\varphi))\mathbf{S}(\neg st \wedge \mathcal{R}(\psi))$
- ...

LTL rewriting

We defined a rewriting for LTL + past + FO

Property "Theorem 1"

$$\pi_{local} \models \varphi \Leftrightarrow \pi_{global} \models \mathcal{R}^*(\varphi)$$

$\mathcal{R}$ it assumes to be on a state where $\neg st$ holds

- a: $\mathcal{R}(a) := a$
- $\neg$: $\mathcal{R}(\neg\varphi) := \neg\mathcal{R}(\varphi)$
- $\vee$: $\mathcal{R}(\varphi \vee \psi) := \mathcal{R}(\varphi) \vee \mathcal{R}(\psi)$
- $\mathbf{X}$: $\mathcal{R}(\mathbf{X}\varphi) := \mathbf{X}(\neg st\mathbf{R}(st \vee \mathcal{R}(\varphi)))$
- $\mathbf{U}$: $\mathcal{R}(\varphi\mathbf{U}\psi) := (st \vee \mathcal{R}(\varphi))\mathbf{U}(\neg st \wedge \mathcal{R}(\psi))$
- $\mathbf{Y}$: $\mathcal{R}(\mathbf{Y}\varphi) := \mathbf{Y}(st\mathbf{S}(\neg st \wedge \mathcal{R}(\varphi)))$
- $\mathbf{S}$: $\mathcal{R}(\varphi\mathbf{S}\psi) := (st \vee \mathcal{R}(\varphi))\mathbf{S}(\neg st \wedge \mathcal{R}(\psi))$
- ...

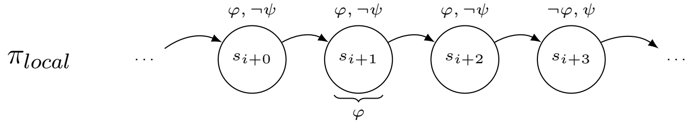
$\mathcal{R}^*$ "maps" 0 to the first transition with $\neg st$

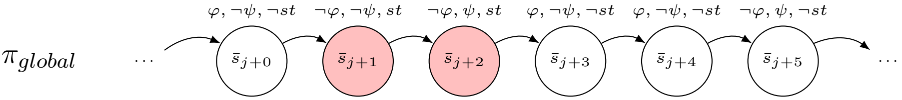
$$\mathcal{R}^*(\varphi) = \neg st\mathbf{R}(st \vee \mathcal{R}(\varphi))$$

Example neXt

$\mathcal{R}$ example: **X**

$Prop_{loc} = \mathbf{X}\varphi$ $\qquad$ $Prop_{glob} = \mathcal{R}(\mathbf{X}\varphi) = \mathbf{X}\neg st\mathbf{R}(st \vee \varphi)$

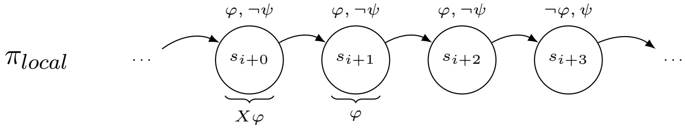


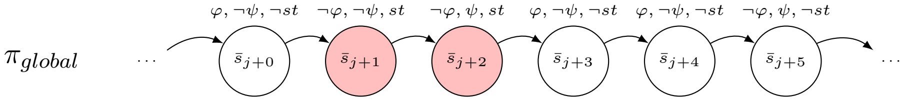


Example neXt

$\mathcal{R}$ example: $\mathbf{X}$

$Prop_{loc} = \mathbf{X}\varphi$ $Prop_{glob} = \mathcal{R}(\mathbf{X}\varphi) = \mathbf{X}\neg st\mathbf{R}(st \vee \varphi)$

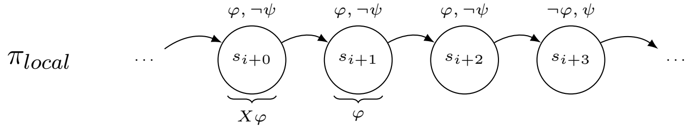


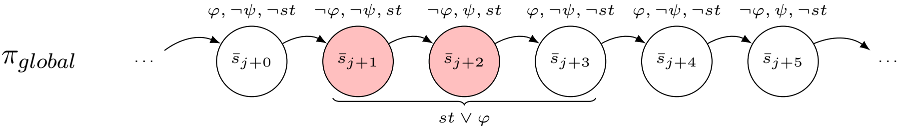


Example neXt

$\mathcal{R}$ example: **X**

$Prop_{loc} = \mathbf{X}\varphi$ $\quad$ $Prop_{glob} = \mathcal{R}(\mathbf{X}\varphi) = \mathbf{X}\neg st\mathbf{R}(st \vee \varphi)$

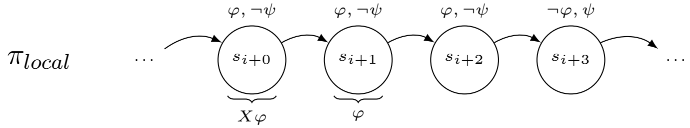


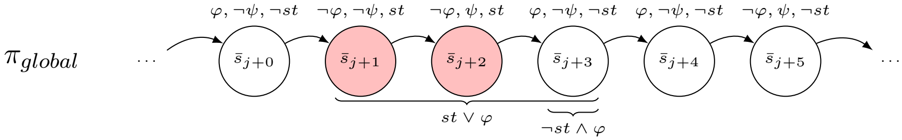


Example neXt

$\mathcal{R}$ example: **X**

$Prop_{loc} = \mathbf{X}\varphi$ $\qquad$ $Prop_{glob} = \mathcal{R}(\mathbf{X}\varphi) = \mathbf{X}\neg st\mathbf{R}(st \vee \varphi)$



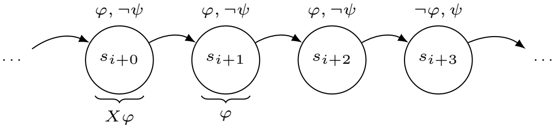


Example neXt

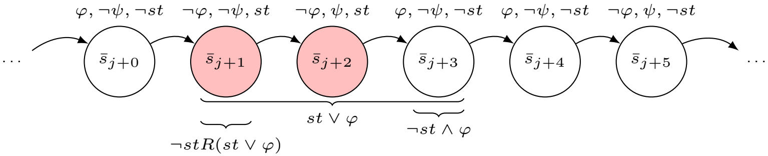
$\mathcal{R}$ example: **X**

$Prop_{loc} = \mathbf{X}\varphi$ $\qquad$ $Prop_{glob} = \mathcal{R}(\mathbf{X}\varphi) = \mathbf{X}\neg st\mathbf{R}(st \vee \varphi)$

π_{local}

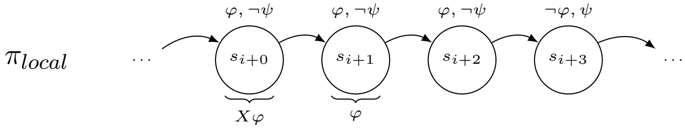


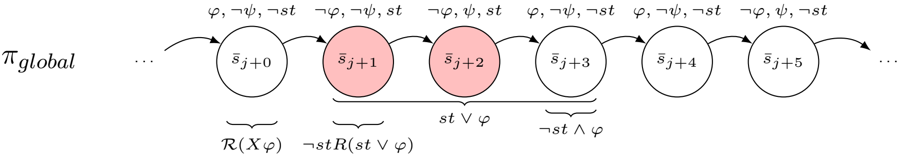
π_{global}



$\mathcal{R}$ example: **X**

$Prop_{loc} = \mathbf{X}\varphi$ $\qquad$ $Prop_{glob} = \mathcal{R}(\mathbf{X}\varphi) = \mathbf{X}\neg st\mathbf{R}(st \vee \varphi)$

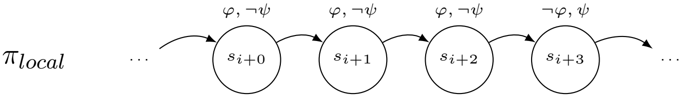




Example Until

$\mathcal{R}$ example: U

$Prop_{loc} = \varphi \mathbf{U} \psi$ $\quad$ $Prop_{glob} = \mathcal{R}(\varphi \mathbf{U} \psi) = (st \vee \varphi)\mathbf{U}(\neg st \wedge \psi)$

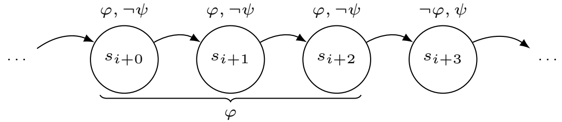


Example Until

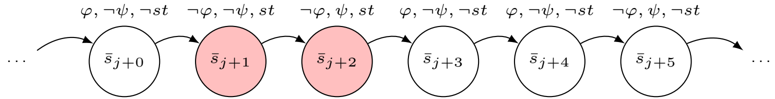
$\mathcal{R}$ example: U

$Prop_{loc} = \varphi \mathbf{U} \psi$ $\qquad$ $Prop_{glob} = \mathcal{R}(\varphi \mathbf{U} \psi) = (st \vee \varphi)\mathbf{U}(\neg st \wedge \psi)$

π_{local}



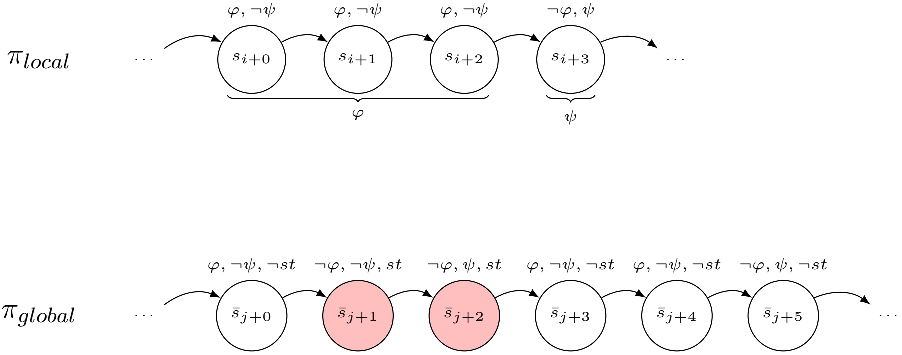
π_{global}



Example Until

$\mathcal{R}$ example: U

$Prop_{loc} = \varphi \mathbf{U} \psi$ $\qquad$ $Prop_{glob} = \mathcal{R}(\varphi \mathbf{U} \psi) = (st \vee \varphi)\mathbf{U}(\neg st \wedge \psi)$

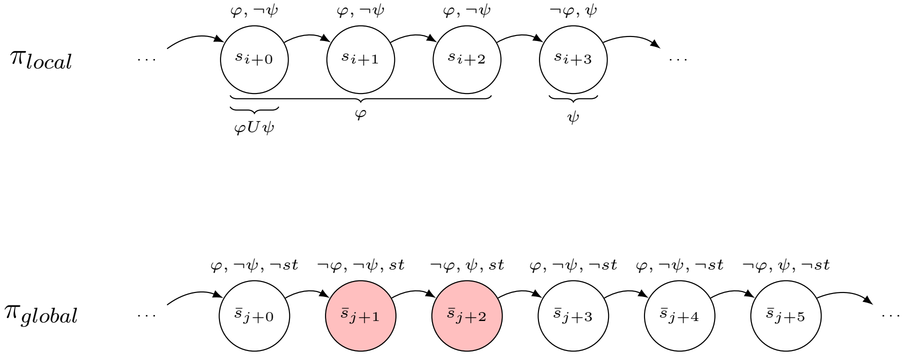


Example Until

$\mathcal{R}$ example: U

$Prop_{loc} = \varphi \mathbf{U} \psi$

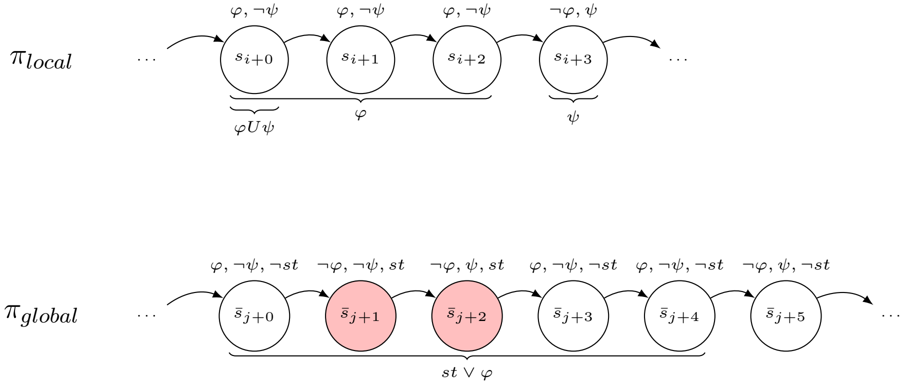
$Prop_{glob} = \mathcal{R}(\varphi \mathbf{U} \psi) = (st \vee \varphi)\mathbf{U}(\neg st \wedge \psi)$



Example Until

$\mathcal{R}$ example: U

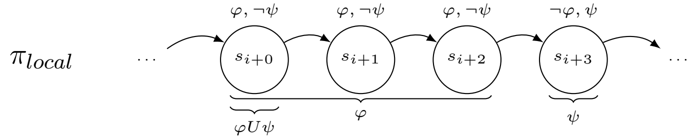
$Prop_{loc} = \varphi \mathbf{U} \psi$ $\qquad$ $Prop_{glob} = \mathcal{R}(\varphi \mathbf{U} \psi) = (st \vee \varphi)\mathbf{U}(\neg st \wedge \psi)$

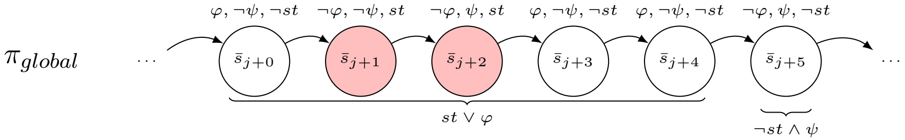


Example Until

$\mathcal{R}$ example: U

$Prop_{loc} = \varphi \mathbf{U} \psi$ $\quad$ $Prop_{glob} = \mathcal{R}(\varphi \mathbf{U} \psi) = (st \vee \varphi)\mathbf{U}(\neg st \wedge \psi)$



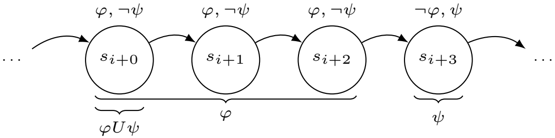


Example Until

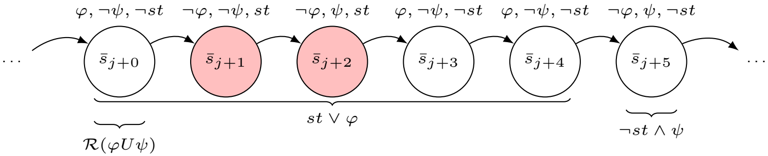
$\mathcal{R}$ example: U

$Prop_{loc} = \varphi \mathbf{U} \psi$ $\qquad$ $Prop_{glob} = \mathcal{R}(\varphi \mathbf{U} \psi) = (st \vee \varphi)\mathbf{U}(\neg st \wedge \psi)$

π_{local}



π_{global}



Optimization

Properties with input and outputs

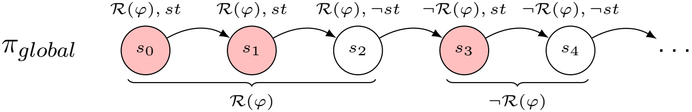
- Properties are over **input** and **output** variables
- **output** variables do not change when st holds

Optimization

Properties with input and outputs

- Properties are over **input** and **output** variables
- **output** variables do not change when st holds

"Stutter tolerance" of φ:

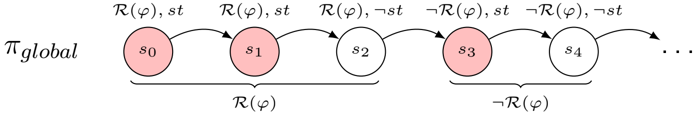


Optimization

Properties with input and outputs

- Properties are over **input** and **output** variables
- **output** variables do not change when st holds

"Stutter tolerance" of φ:



Applying stutter tolerance

- Stutter tolerant formulae are found syntactically (e.g. $\mathbf{U}, o_{var}$)
- If sub-formula is *"syntactically" stutter tolerant*, then it is not necessary to apply rewriting to the current op:
  - $\mathcal{R}^\theta(o1_{var}\mathbf{U}o2_{var}) = o1_{var}\mathbf{U}o2_{var}$
  - $\mathcal{R}^\theta(\mathbf{X}(o1_{var}\mathbf{U}o2_{var})) = \mathbf{X}(o1_{var}\mathbf{U}o2_{var})$
- Stutter tolerance also used for $\mathcal{R}^*$

Verification of composition

$$\gamma_P(\{\varphi_i\}) := \bigwedge_i \Big(\overbrace{\mathcal{R}^*(\varphi_i)}^{\text{Apply rewriting}} \wedge \underbrace{\psi^i_{cond}}_{GF\neg st_i \wedge (st_i \rightarrow \bigwedge_{v \in \mathcal{V}_i^O} (v = v'))} \Big)$$

Verification of composition

$$\gamma_P(\{\varphi_i\}) := \bigwedge_i \Big(\overbrace{\mathcal{R}^*(\varphi_i)}^{\text{Apply rewriting}} \wedge \underbrace{\psi^i_{cond}}_{GF\neg st_i \wedge (st_i \rightarrow \bigwedge_{v \in \mathcal{V}_i^O} (v = v'))} \Big)$$

$\{\varphi_i\}$

Verification of composition

$$\gamma_P(\{\varphi_i\}) := \bigwedge_i \Big(\overbrace{\mathcal{R}^*(\varphi_i)}^{\text{Apply rewriting}} \wedge \underbrace{\psi^i_{cond}}_{GF\neg st_i \wedge (st_i \rightarrow \bigwedge_{v \in \mathcal{V}^O_i}(v = v'))} \Big)$$

$$\{\varphi_i\} \xrightarrow{\text{async composition}} \gamma_P(\{\varphi_i\})$$

Verification of composition

$$\gamma_P(\{\varphi_i\}) := \bigwedge_i \big(\overbrace{\mathcal{R}^*(\varphi_i)}^{\text{Apply rewriting}} \wedge \underbrace{\psi^i_{cond}}_{GF\neg st_i \wedge (st_i \rightarrow \bigwedge_{v\in\mathcal{V}_i^O}(v=v'))} \big)$$

$$\{\varphi_i\} \xrightarrow{\text{async composition}} \gamma_P(\{\varphi_i\}) \xrightarrow{LTL2TS} \mathcal{M}$$

Alternative approach

Alternative approach

$$\begin{tikzcd}
\{\varphi_i\} \arrow[r, "\text{async composition}"] & \gamma_P(\{\varphi_i\}) \arrow[d, "LTL2TS"'] \\
& \mathcal{M}
\end{tikzcd}$$

- **Rewriting** approach:
  1. **Rewrite** local property with $\mathcal{R}^*$
  2. Construct the global automata

Alternative approach

$$
\begin{tikzcd}
\{\varphi_i\} \arrow[r, "\text{async composition}"] \arrow[d, "*LTL2TS"'] & \gamma_P(\{\varphi_i\}) \arrow[d, "LTL2TS"'] \\
\{\mathcal{M}_{\varphi_i}\} & \mathcal{M}
\end{tikzcd}
$$

- **Rewriting** approach:
  1. **Rewrite** local property with $\mathcal{R}^*$
  2. Construct the global automata
- **Alternative** approach:
  1. Transform φ in an equivalent TS

Alternative approach

$$\begin{tikzcd}
\{\varphi_i\} \arrow[r, "\text{async composition}"] \arrow[d, "*LTL2TS"'] & \gamma_P(\{\varphi_i\}) \arrow[d, "LTL2TS"'] \\
\{\mathcal{M}_{\varphi_i}\} \arrow[r, "\text{async composition}"] & \mathcal{M}
\end{tikzcd}$$

- **Rewriting** approach:
  1. **Rewrite** local property with $\mathcal{R}^*$
  2. Construct the global automata
- **Alternative** approach:
  1. Transform φ in an equivalent TS
  2. Compose asynchronously $\mathcal{M}_\varphi$ with other TS

Alternative approach

$$\begin{tikzcd}
\{\varphi_i\} \arrow[r, "\text{async composition}"] \arrow[d, "*LTL2TS"'] & \gamma_P(\{\varphi_i\}) \arrow[d, "LTL2TS"'] \\
\{\mathcal{M}_{\varphi_i}\} \arrow[r, "\text{async composition}"] & \mathcal{M}
\end{tikzcd}$$

- **Rewriting** approach:
  1. **Rewrite** local property with $\mathcal{R}^*$
  2. Construct the global automata
- **Alternative** approach:
  1. Transform φ in an equivalent TS
  2. Compose asynchronously $\mathcal{M}_\varphi$ with other TS

Approaches are equivalent

* Requires some modifications to handle input variables

Results

Results

Implementation and evaluation

- Implemented inside contract-based tool **OCRA**
- Theoretical work validated on random formula checking their trace
- Experimental evaluation carried out over diverse type of models
- Compared with simpler rewriting that supports only sync **events**

Results

Implementation and evaluation

- Implemented inside contract-based tool **OCRA**
- Theoretical work validated on random formula checking their trace
- Experimental evaluation carried out over diverse type of models
- Compared with simpler rewriting that supports only sync **events**

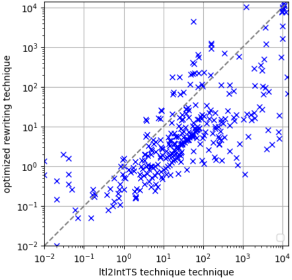


Figure: Alternative approach vs Opt rewriting

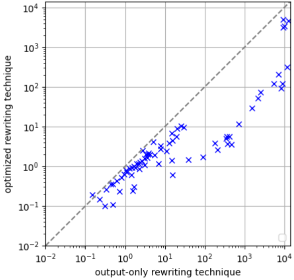


Figure: Event based rewriting vs Opt rewriting

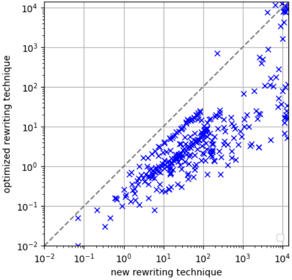


Figure: $\mathcal{R}^*$ vs Opt rewriting

Conclusion and future works

Contribution summary:

- Definition and demonstration $\mathcal{R}^*$ rewriting to compose local properties
- Optimization of $\mathcal{R}^*$ for properties with *input*/*output* variables
- Alternative compositional approach based on $LTL2SMV$
- Implementation inside *contract-based* tool **OCRA**

Conclusion and future works

Contribution summary:

- Definition and demonstration $\mathcal{R}^*$ rewriting to compose local properties
- Optimization of $\mathcal{R}^*$ for properties with *input*/*output* variables
- Alternative compositional approach based on $LTL2SMV$
- Implementation inside *contract-based* tool **OCRA**

Future works

- Extend this work for *timed* (with both local time and global time semantics)
- Drop assumption on local infinite traces (executions)
- Scheduling constraints synthesis

Questions?

Details: $\mathcal{R}$

- $\mathcal{R}(a) := a$
- $\mathcal{R}(\varphi \vee \psi) := \mathcal{R}(\varphi) \vee \mathcal{R}(\psi)$
- $\mathcal{R}(\neg\varphi) := \neg\mathcal{R}(\varphi)$
- $\mathcal{R}(\mathbf{X}\psi) := \mathbf{X}(\neg st\mathbf{R}(st \vee \mathcal{R}(\psi)))$
- $\mathcal{R}(\varphi\mathbf{U}\psi) := (st \vee \mathcal{R}(\varphi))\mathbf{U}(\neg st \wedge \mathcal{R}(\psi))$
- $\mathcal{R}(\mathbf{Y}\varphi) := \mathbf{Y}(st\mathbf{S}(\neg st \wedge \mathcal{R}(\varphi)))$
- $\mathcal{R}(\varphi\mathbf{S}\psi) := (st \vee \mathcal{R}(\varphi))\mathbf{S}(\neg st \wedge \mathcal{R}(\psi))$
- $\mathcal{R}(func(\psi_1, ..., \psi_n)) := func(\mathcal{R}(\psi_1), ..., \mathcal{R}(\psi_n))$
- $\mathcal{R}(pred(\psi_1, ..., \psi_n)) := pred(\mathcal{R}(\psi_1), ..., \mathcal{R}(\psi_n))$
- $\mathcal{R}(ite(\psi, \psi_1, \psi_2)) := ite(\mathcal{R}(\psi), \mathcal{R}(\psi_1)\mathcal{R}(\psi_2))$
- $\mathcal{R}(next(\psi) := \psi\tilde{F}@\neg st$
- $\mathcal{R}(\psi\tilde{F}@\psi_1) := \mathcal{R}(\psi)\tilde{F}@(\mathcal{R}(\psi_1) \wedge \neg st)$
- $\mathcal{R}(\psi\tilde{P}@\psi_1) := \mathcal{R}(\psi)\tilde{P}@(\mathcal{R}(\psi_1) \wedge \neg st)$

Details: $\mathcal{R}^\theta$

- $\mathcal{R}^\theta(s) = \mathcal{T}(s)$ if $s \in \mathcal{V}$
- $\mathcal{R}^\theta(\varphi \vee \psi) = \mathcal{R}^\theta(\varphi) \vee \mathcal{R}^\theta(\psi)$
- $\mathcal{R}^\theta(\neg\varphi) = \neg\mathcal{R}^\theta(\varphi)$
- $\mathcal{R}^\theta(\mathbf{X}\psi) = \begin{cases} \mathbf{X}(\mathcal{R}^\theta(\psi)) & \text{if } \psi \text{ is synt. st.tol.} \\ \mathbf{X}(\neg st\mathbf{R}(st \vee \mathcal{R}^\theta(\psi))) & \text{otherwise} \end{cases}$
- $\mathcal{R}^\theta(\varphi\mathbf{U}\psi) = \begin{cases} \mathcal{R}^\theta(\varphi)\mathbf{U}\mathcal{R}^\theta(\psi) & \text{if } \psi \text{ is synt. st.tol.} \\ (st \vee \mathcal{R}^\theta(\varphi))\mathbf{U}(\neg st \wedge \mathcal{R}^\theta(\psi)) & \text{otherwise} \end{cases}$
- $\mathcal{R}^\theta(\mathbf{Y}\psi) = \mathbf{Y}(st\mathbf{S}(\neg st \wedge \mathcal{R}^\theta(\psi)))$
- $\mathcal{R}^\theta(\varphi\mathbf{S}\psi) = \begin{cases} \mathcal{R}^\theta(\varphi)\mathbf{S}\mathcal{R}^\theta(\psi) & \text{if } \psi \text{ is synt. st.tol} \\ (st \vee \mathcal{R}^\theta(\varphi))\mathbf{S}(\neg st \wedge \mathcal{R}^\theta(\psi)) & \text{otherwise} \end{cases}$

Details: $\mathcal{R}^\theta$ contd.

- $\mathcal{R}^\theta(func(\psi_1, ..., \psi_n)) = func(\mathcal{R}^\theta(\psi_1), ..., \mathcal{R}^\theta(\psi_n))$
- $\mathcal{R}^\theta(pred(\psi_1, ..., \psi_n)) = pred(\mathcal{R}^\theta(\psi_1), ..., \mathcal{R}^\theta(\psi_n))$
- $\mathcal{R}^\theta(ite(\psi, \psi_1, \psi_2)) = ite(\mathcal{R}^\theta(\psi), \mathcal{R}^\theta(\psi_1), \mathcal{R}^\theta(\psi_2))$
- $\mathcal{R}^\theta(next(\psi)) = \begin{cases} next(\mathcal{R}^\theta(\psi)) & \text{if } \psi \text{ is synt. st.tol.} \\ \mathcal{R}^\theta(\psi)@F\neg st & \text{otherwise} \end{cases}$
- $\mathcal{R}^\theta(\psi@F\psi_1) = \begin{cases} \mathcal{R}^\theta(\psi)@F\mathcal{R}^\theta(\psi_1) & \text{if } \psi \text{ is synt. st. tol.} \\ \mathcal{R}^\theta(\psi)@F(\neg st \wedge \mathcal{R}^\theta(\psi_1)) & \text{otherwise} \end{cases}$
- $\mathcal{R}^\theta(\psi\tilde{P}@\psi_1) = \mathcal{R}^\theta(\psi)\tilde{P}@(\neg st \wedge \mathcal{R}^\theta(\psi_1))$

Details: Lemmas and theorem

Lemma 1:

For all π, for all $\pi^{ST} \in Pr^{-1}(\pi)$, for all i :

$$\pi, i \models \varphi \Leftrightarrow \pi^{ST}, map_{\pi^{ST}}(i) \models \mathcal{R}(\varphi)$$

Lemma 2:

For all π, for all $\pi^{ST} \in Pr^{-1}(\pi)$:

$$\pi^{ST}, 0 \models \mathcal{R}^*(\varphi) \Leftrightarrow \pi^{ST}, map_{\pi^{ST}}(0) \models \mathcal{R}(\varphi)$$

Theorem 1:

For all π, for all $\pi^{ST} \in Pr^{-1}(\pi)$:

$$\pi, \models \varphi \Leftrightarrow \pi^{ST} \models \mathcal{R}^*(\varphi)$$

Iinterface Transition System and composition

$$\mathcal{M} = \mathcal{M}_1 \otimes \mathcal{M}_2$$

$$\mathcal{M} = \langle \mathcal{V}^I, \mathcal{V}^O, \mathcal{V}^H, \mathcal{I}, \mathcal{T}, \mathcal{F} \rangle$$

- $\mathcal{V}^I$ (input vars)
- $\mathcal{V}^O$ (output vars)
- $\mathcal{V}^H$ (internal vars)
- $\mathcal{I}$ (init)
- $\mathcal{T}$ (transitions)
- $\mathcal{F}$ (fairness)

- $\mathcal{V}^I = (\mathcal{V}_1^I \cup \mathcal{V}_2^I) \setminus Shared(\mathcal{M}_1, \mathcal{M}_2)$
- $\mathcal{V}^O = (\mathcal{V}_1^O \cup \mathcal{V}_2^O) \setminus Shared(\mathcal{M}_1, \mathcal{M}_2)$
- $\mathcal{V}^H = \mathcal{V}_1^H \cup \mathcal{V}_2^H \cup Shared(\mathcal{M}_1, \mathcal{M}_2) \cup st_1, st_2$
- $\mathcal{I} = \mathcal{I}_1 \wedge \mathcal{I}_2$
- $\mathcal{T} = (\neg st_1 \rightarrow \mathcal{T}_1) \wedge (\neg st_2 \rightarrow \mathcal{T}_2) \wedge (st_1 \rightarrow \bigwedge_{v^o \in \mathcal{V}_1^O \cup \mathcal{V}_1^H} (v^{o\prime} = v^o)) \wedge (st_2 \rightarrow \bigwedge_{v^o \in \mathcal{V}_2^O \cup \mathcal{V}_2^H} (v^{o\prime} = v^o))$
- $\mathcal{F} = \{\varphi_1 \wedge \neg st_1 | \varphi_1 \in \mathcal{F}_1\} \cup \{\varphi_2 \wedge \neg st_2 | \varphi_2 \in \mathcal{F}_2\} \cup \{\neg st_1, \neg st_2\}$

Rewriting example

- $\mathcal{M}_1$ with c_2 **input** and c_1 **output**
- $\mathcal{M}_2$ with c_1 **input** and c_2 **output**
- $\varphi_1 := c_1 = 0 \wedge G((c_1 < c_2 \wedge c_1' = c_1 + 1) \vee (c_1 \geq c_2 \wedge c_1' = c_1))$
- $\varphi_2 := c_2 = p \wedge G((c_2' = c_2 - 1)U(c_2 = 0 \wedge c_2' = c_1))$
- $\mathcal{R}^*_{\mathcal{M}_1}(\varphi_1)$:
  $\neg st^{\mathcal{M}_1} R(st \vee (c_1 = 0 \wedge G(st^{\mathcal{M}_1} \vee (c_1 < c_2 \wedge c_1 \tilde{F} @ \neg st^{\mathcal{M}_1} = c1 + 1 \vee c_1 \geq c_2 \wedge c_1 \tilde{F} @ \neg st^{\mathcal{M}_1} = c_1)))$
- $\mathcal{R}^*_{\mathcal{M}_2}(\varphi_2)$:
  $\neg st^{\mathcal{M}_2} R(st^{\mathcal{M}_2} \vee c_2 = p \wedge G(st^{\mathcal{M}_2} \vee ((st^{\mathcal{M}_2} \vee c_2 \tilde{F} @ \neg st^{\mathcal{M}_2} = c_2 - 1)U(\neg st^{\mathcal{M}_2} \wedge c_2 = 0 \wedge c_2 \tilde{F} @ \neg st^{\mathcal{M}_2} = c_1))))$
- $\psi_{cond}(\mathcal{M}_1, \mathcal{M}_2) =$
  $G(\neg st^{\mathcal{M}_1} \vee c_1 = c_1') \wedge GF \neg st^{\mathcal{M}_1} \wedge G(\neg st^{\mathcal{M}_2} \vee c_2 = c_2') \wedge GF \neg st^{\mathcal{M}_2}$

Optimized rewriting example

- $\mathcal{M}_1$ with c_2 **input** and c_1 **output**
- $\mathcal{M}_2$ with c_1 **input** and c_2 **output**
- $\varphi_1 := c_1 = 0 \wedge G((c_1 < c_2 \wedge c_1' = c_1 + 1) \vee (c_1 \geq c_2 \wedge c_1' = c_1))$
- $\varphi_2 := c_2 = p \wedge G((c_2' = c_2 - 1)U(c_2 = 0 \wedge c_2' = c_1))$
- ${\mathcal{R}^{\theta}}^*_{\mathcal{M}_1}(\varphi_1) : c_1 = 0 \wedge G(st^{\mathcal{M}_1} \vee (c_1 < c_2 \wedge c_1' = c1 + 1 \vee c_1 \geq c_2 \wedge c_1' = c_1)))$
- ${\mathcal{R}^{\theta}}^*_{\mathcal{M}_2}(\varphi_2) : c_2 = p \wedge G((st^{\mathcal{M}_2} \vee c_2' = c_2 - 1)U(\neg st^{\mathcal{M}_2} \wedge c_2 = 0 \wedge c_2' = c_1))$

Experiments

LTL Patterns

Types:

- response
- precedence chain
- Universal

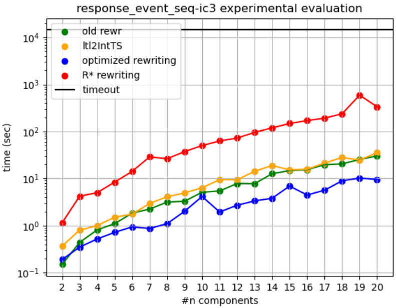
Component wiring:

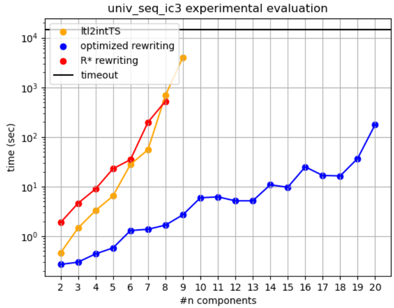
- sequential
- parallel

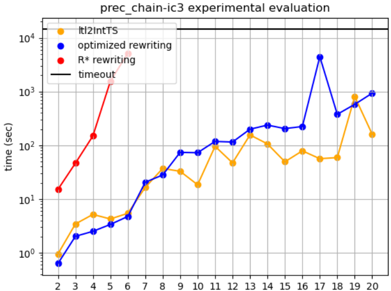
Nested X

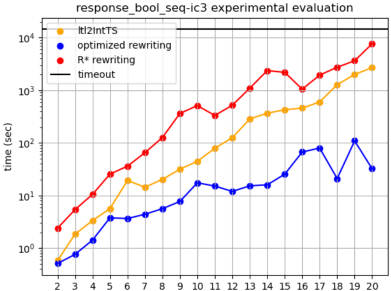
- n local components
- m nested **X**
- Global property entailed by local properties

Plots (1)

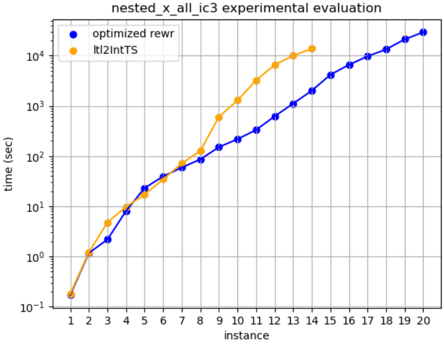

response_event_seq-ic3 experimental evaluation
old rewr
ltl2IntTS
optimized rewriting
R* rewriting
timeout
time (sec)
#n components


univ_seq_ic3 experimental evaluation
ltl2intTS
optimized rewriting
R* rewriting
timeout
time (sec)
#n components

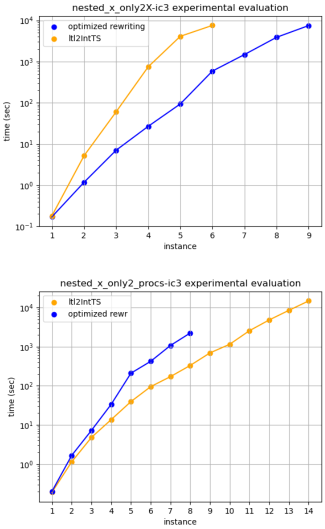

prec_chain-ic3 experimental evaluation
ltl2IntTS
optimized rewriting
R* rewriting
timeout
time (sec)


response_bool_seq-ic3 experimental evaluation
ltl2IntTS
optimized rewriting
R* rewriting
timeout
time (sec)

Plots (2)



(a) Overall incremental results



(b) Results with respectively fixed X and fixed components

Related work

Related work comparison				
Work	Logic	finite traces	I/O vars	next semantics
1	TLA+	Yes	I/O	global
2	LTL + weak X	Yes	I	local
3	LTL	No	O + events	local
4	LHA with LTL	No	O + events	local
$\mathcal{R}^*$	LTL+FO+@F	No	I/O	local

1. L. Lamport. The operators of tla. 06 1997
2. C. Eisner, D. Fisman, J. Havlicek, A. McIsaac, and D. V. Campenhout. The Definition of a Temporal Clock Operator. In ICALP, volume 2719 of Lecture Notes in Computer Science, pages 857–870. Springer, 2003.
3. N. Benes, L. Brim, I. Cerná, J. Sochor, P. Vareková, and B. Buhnova. Partial order reduction for state/event ltl. In IFM, 2009.
4. Cimatti, A., Griggio, A., Mover, S., Tonetta, S. (2015). HYCOMP: An SMT-Based Model Checker for Hybrid Systems.

TLA+

$$Init \wedge \Box(\mathcal{T}(v,v')) \underbrace{\bar{\mathcal{V}} \subseteq \mathcal{V}}_{\bigwedge_{v \in \bar{\mathcal{V}} v' = v}} \wedge Fair$$

- $next$ over *output symbols* is equivalent (symbols of $\bar{V}$)
- $next$ over *input symbols* $\neq$ $next$ over "local" trace
- TLA+ has implicit stuttering while we specify st

LTL clocked operator

Semantics

- $w \models^c p \Leftrightarrow \forall j < |w|$ s.t. $w^{0..j}$ is a clock tick of $c, p \in w^j$
- $w \models^c p! \Leftrightarrow \exists j < |w|$ s.t. $w^{0..j}$ is a clock tick of c and $p \in w^j$
- $w \models^c \neg f \Leftrightarrow w \not\models^c f$
- $w \models^c f_1 \wedge f_2 \Leftrightarrow w \models^c f_1$ and $w \models^c f_2$
- $w \models^c \mathbf{X}!f \Leftrightarrow \exists j < k < |w|$ s.t. $w^{0..j}$ is a clock tick of c and $w^{j+1..k}$ is a clock tick of c and $w^{k..} \models^c f$
- $w \models^c f\mathbf{U}g \Leftrightarrow \exists k < |w|$ s.t.$w^k \models c$ and $w^{k..} \models^c g$ and $\forall j < k$ s.t. $w^j \models c, w^{j..} \models^c f$
- $w \models^c f@c_1 \Leftrightarrow w \models^{c_1} f$

LTL clocked operator (2)

Rewriting form

- $\mathcal{T}^c(p) := \neg c \mathbf{W}(c \wedge p)$
- $\mathcal{T}^c(p!) := \neg c \mathbf{U}(c \wedge p)$
- $\mathcal{T}^c(f \vee g) := \mathcal{T}^c(f) \vee \mathcal{T}^c(g)$
- $\mathcal{T}^c(\mathbf{X}!f) := \neg c \mathbf{U}(c \wedge \mathbf{X}!(\neg c \mathbf{U}(c \wedge \mathcal{T}^c(f))$
- $\mathcal{T}^c(f \mathbf{U} g) := (\neg c \vee \mathcal{T}^c(f))\mathbf{U}(c \wedge T^c(g))$
- $\mathcal{T}^c(f@c_1) = \mathcal{T}^{c_1}(f)$

LTL clocked operators differences

- propositional LTL with weak strong X LTL + past + first order + at next
- Clocked handles finite executions
- Clocked uses stutter sequence for each proposition