Robust, Expressive, and Quantitative Linear Temporal Logics: Pick any Two for Free

Joint work with Daniel Neider and Alexander Weinert

Martin Zimmermann

University of Liverpool

September 3rd, 2019

GandALF 2019, Bordeaux, France

Linear Temporal Logic (LTL)

- The most prominent and most important specification language for reactive systems.

Linear Temporal Logic (LTL)

- The most prominent and most important specification language for reactive systems.

Examples

- $\Box(q \rightarrow \Diamond p)$: every request is responded to eventually.
- $\Box a \rightarrow \Box g$: if assumption holds always, then guarantee holds always.

Linear Temporal Logic (LTL)

- The most prominent and most important specification language for reactive systems.
- **Exponential Compilation Property** (ECP): every LTL formula can be translated into a Büchi automaton of exponential size.
- ECP yields model checking in $\textsc{PSpace}$ and synthesis in $2\textsc{ExpTime}$.

Linear Temporal Logic (LTL)

- The most prominent and most important specification language for reactive systems.
- **Exponential Compilation Property** (ECP): every LTL formula can be translated into a Büchi automaton of exponential size.
- ECP yields model checking in PSPACE and synthesis in 2EXPTIME.

Shortcomings

- Inability to express timing constraints
- Limited expressiveness (weaker than Büchi automata)
- Inability to capture robustness

Linear Temporal Logic (LTL)

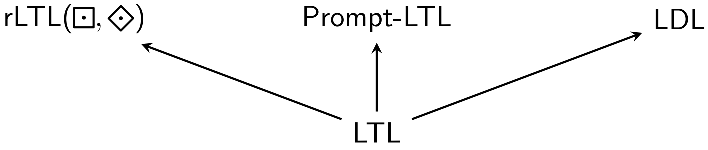
- The most prominent and most important specification language for reactive systems.
- **Exponential Compilation Property** (ECP): every LTL formula can be translated into a Büchi automaton of exponential size.
- ECP yields model checking in $\textsc{PSpace}$ and synthesis in $2\textsc{ExpTime}$.

Shortcomings

- Inability to express timing constraints
- Limited expressiveness (weaker than Büchi automata)
- Inability to capture robustness

All three shortcomings have been addressed before..

The Big Picture



Prompt-LTL

Kupferman, Piterman, Vardi ('09): Add timing constraints to LTL

Syntax

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \bigcirc \varphi \mid \varphi \mathbf{U} \varphi \mid \varphi \mathbf{R} \varphi \mid \Diamond_{\mathbf{p}} \varphi$$

Prompt-LTL

Kupferman, Piterman, Vardi ('09): Add timing constraints to LTL

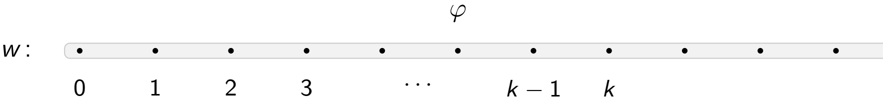
Syntax

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \bigcirc \varphi \mid \varphi \,\mathbf{U}\, \varphi \mid \varphi \,\mathbf{R}\, \varphi \mid \Diamond_{\mathbf{p}} \varphi$$

Semantics

via evaluation function V^{P} mapping a trace w, a bound k, and a formula φ to a truth value in $\{0,1\}$.

- $V^{\mathrm{P}}(w, k, \Diamond_{\mathbf{p}} \varphi) = 1$ iff



Prompt-LTL

Kupferman, Piterman, Vardi ('09): Add timing constraints to LTL

Syntax

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \bigcirc \varphi \mid \varphi \mathbf{U} \varphi \mid \varphi \mathbf{R} \varphi \mid \Diamond_{\mathbf{p}} \varphi$$

Example

- $\Box(\mathtt{q} \to \Diamond_{\mathbf{p}} \mathtt{p})$: every request is responded to within k steps.

Linear Dynamic Logic

Vardi ('11): Add guards to $\Diamond$ and $\Box$ to restrict their scope

Syntax

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle r \rangle \varphi \mid [r] \varphi$$
$$r ::= \phi \mid \varphi? \mid r + r \mid r \,; r \mid r^*$$

where ϕ ranges over boolean formulas over the atomic propositions.

Linear Dynamic Logic

Vardi ('11): Add guards to $\Diamond$ and $\Box$ to restrict their scope

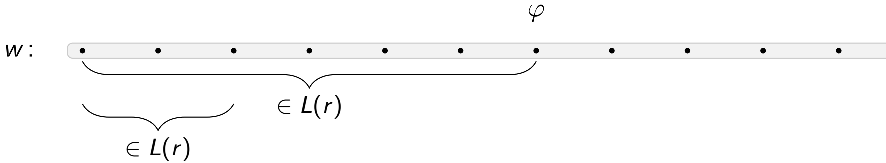
Syntax

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle r \rangle \varphi \mid [r] \varphi$$
$$r ::= \phi \mid \varphi? \mid r + r \mid r \,;\, r \mid r^*$$

where ϕ ranges over boolean formulas over the atomic propositions.

Semantics

- $V^{\mathrm{D}}(w, \langle r \rangle \varphi) = 1$ iff



Linear Dynamic Logic

Vardi ('11): Add guards to $\Diamond$ and $\Box$ to restrict their scope

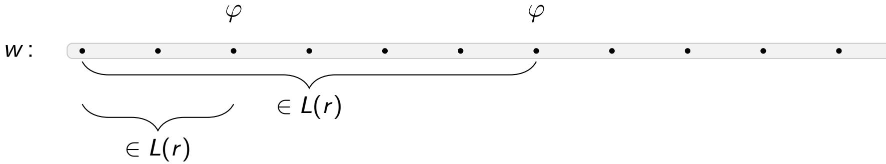
Syntax

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle r \rangle \varphi \mid [r] \varphi$$
$$r ::= \phi \mid \varphi? \mid r + r \mid r \,; r \mid r^*$$

where ϕ ranges over boolean formulas over the atomic propositions.

Semantics

- $V^{\mathrm{D}}(w, [r]\varphi) = 1$ iff



Linear Dynamic Logic

Vardi ('11): Add guards to $\Diamond$ and $\Box$ to restrict their scope

Syntax

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle r \rangle \varphi \mid [r] \varphi$$
$$r ::= \phi \mid \varphi? \mid r + r \mid r \,;\, r \mid r^*$$

where ϕ ranges over boolean formulas over the atomic propositions.

Example

- $[r]\,\mathtt{p}$ with $r = (\mathtt{tt}\,;\mathtt{tt})^*$: $\mathtt{p}$ holds at every even position.

Robust LTL

Tabuada and Neider ('16): Capture robustness in LTL semantics

Consider the five (canonical) ways $\Box a$ can be satisfied/violated:

1. a holds always ($\Box a$)
2. a holds almost always ($\Diamond \Box a$)
3. a holds infinitely often ($\Box \Diamond a$)
4. a holds at least once ($\Diamond a$)
5. a holds never ($\Box \neg a$)

Robust LTL

Tabuada and Neider ('16): Capture robustness in LTL semantics

Consider the five (canonical) ways $\Box a$ can be satisfied/violated:

1. a holds always ($\Box a$)
2. a holds almost always ($\Diamond \Box a$)
3. a holds infinitely often ($\Box \Diamond a$)
4. a holds at least once ($\Diamond a$)
5. a holds never ($\Box \neg a$)

Note that **1.** $\Rightarrow$ **2.** $\Rightarrow$ **3.** $\Rightarrow$ **4.**

Robust LTL

Tabuada and Neider ('16): Capture robustness in LTL semantics

Consider the five (canonical) ways $\Box a$ can be satisfied/violated:

1. a holds always ($\Box a$) 1111
2. a holds almost always ($\Diamond\Box a$) 0111
3. a holds infinitely often ($\Box\Diamond a$) 0011
4. a holds at least once ($\Diamond a$) 0001
5. a holds never ($\Box \neg a$) 0000

Note that **1.** $\Rightarrow$ **2.** $\Rightarrow$ **3.** $\Rightarrow$ **4.**

Robust LTL

Tabuada and Neider ('16): Capture robustness in LTL semantics

Consider the five (canonical) ways $\Box a$ can be satisfied/violated:

1. a holds always ($\Box a$) 1111
2. a holds almost always ($\Diamond\Box a$) 0111
3. a holds infinitely often ($\Box\Diamond a$) 0011
4. a holds at least once ($\Diamond a$) 0001
5. a holds never ($\Box\neg a$) 0000

Note that **1.** $\Rightarrow$ **2.** $\Rightarrow$ **3.** $\Rightarrow$ **4.**

Basis of five-valued robust semantics for LTL.

Robust Semantics

Truth values $\mathbb{B}_4 = \{1111 > 0111 > 0011 > 0001 > 0000\}$

- Truth value for atomic propositions always in $\{1111, 0000\}$

Robust Semantics

Truth values $\mathbb{B}_4 = \{1111 > 0111 > 0011 > 0001 > 0000\}$

- Truth value for atomic propositions always in $\{1111, 0000\}$
- Conjunction and disjunction via minimization and maximization over $\mathbb{B}_4$

Robust Semantics

Truth values $\mathbb{B}_4 = \{1111 > 0111 > 0011 > 0001 > 0000\}$

- Truth value for atomic propositions always in $\{1111, 0000\}$
- Conjunction and disjunction via minimization and maximization over $\mathbb{B}_4$
- Negation based on 1111 representing satisfaction and all other truth values representing shades of violation

Robust Semantics

Truth values $\mathbb{B}_4 = \{1111 > 0111 > 0011 > 0001 > 0000\}$

- Truth value for atomic propositions always in $\{1111, 0000\}$
- Conjunction and disjunction via minimization and maximization over $\mathbb{B}_4$
- Negation based on 1111 representing satisfaction and all other truth values representing shades of violation
- Implication "$\psi \rightarrow \varphi$" satisfied (1111) if truth value of consequence φ not smaller than truth value of antecedent ψ (otherwise truth value of consequence)

Robust Semantics

Truth values $\mathbb{B}_4 = \{1111 > 0111 > 0011 > 0001 > 0000\}$

- Truth value for atomic propositions always in $\{1111, 0000\}$
- Conjunction and disjunction via minimization and maximization over $\mathbb{B}_4$
- Negation based on 1111 representing satisfaction and all other truth values representing shades of violation
- Implication "$\psi \rightarrow \varphi$" satisfied (1111) if truth value of consequence φ not smaller than truth value of antecedent ψ (otherwise truth value of consequence)
- Eventually classical

Robust Semantics

Truth values $\mathbb{B}_4 = \{1111 > 0111 > 0011 > 0001 > 0000\}$

- Truth value for atomic propositions always in $\{1111, 0000\}$
- Conjunction and disjunction via minimization and maximization over $\mathbb{B}_4$
- Negation based on 1111 representing satisfaction and all other truth values representing shades of violation
- Implication "$\psi \rightarrow \varphi$" satisfied (1111) if truth value of consequence φ not smaller than truth value of antecedent ψ (otherwise truth value of consequence)
- Eventually classical
- Always based on intuition from last slide

Robust Semantics

Truth values $\mathbb{B}_4 = \{1111 > 0111 > 0011 > 0001 > 0000\}$

- Truth value for atomic propositions always in $\{1111, 0000\}$
- Conjunction and disjunction via minimization and maximization over $\mathbb{B}_4$
- Negation based on 1111 representing satisfaction and all other truth values representing shades of violation
- Implication "$\psi \rightarrow \varphi$" satisfied (1111) if truth value of consequence φ not smaller than truth value of antecedent ψ (otherwise truth value of consequence)
- Eventually classical
- Always based on intuition from last slide
- Until and release ignored for simplicity

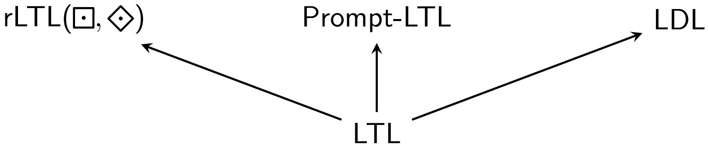
Robust Semantics

Truth values $\mathbb{B}_4 = \{1111 > 0111 > 0011 > 0001 > 0000\}$

Example

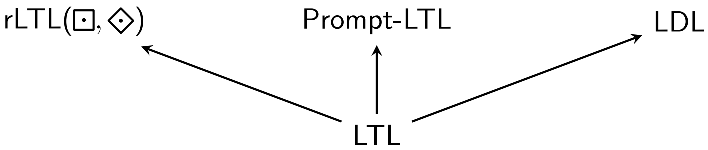
- $\boxdot a \rightarrow \boxdot g$: the level of satisfaction of the guarantee is at least as large as the level of satisfaction of the assumption.

The Big Picture



All three extensions have the ECP..

The Big Picture



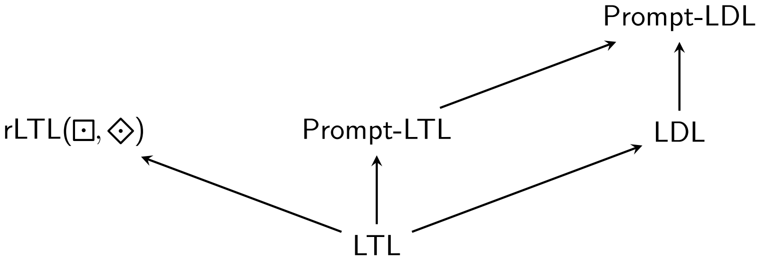
All three extensions have the ECP.. hence model checking is still in PSPACE and synthesis is still in 2EXPTIME!

The Big Picture

$$\begin{tikzcd}
\mathrm{rLTL}(\boxdot, \Diamonddot) & \text{Prompt-LTL} & \mathrm{LDL} \\
 & \mathrm{LTL} \arrow[ul] \arrow[u] \arrow[ur] &
\end{tikzcd}$$

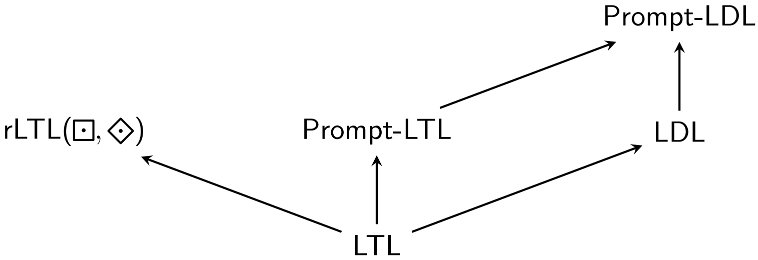
What about combinations of the extensions?

The Big Picture



Faymonville and Z. ('14): the combination of Prompt-LTL and LDL has the ECP, i.e., model checking is still in PSPACE and synthesis is still in 2EXPTIME!

The Big Picture



Here: investigate the remaining combinations

rPrompt-LTL

Syntax

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \boxdot\!\!\!\!\diamond\, \varphi \mid \boxdot\, \varphi \mid \diamond_{\mathbf{p}}\, \varphi$$

Semantics

Via evaluation function V^{RP} (defined as expected).

rPrompt-LTL

Syntax

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \boxdot\!\!\!\diamond \varphi \mid \boxdot \varphi \mid \diamond_{\mathbf{p}} \varphi$$

Example

$$V^{\mathrm{RP}}(w, k, \boxdot \diamond_{\mathbf{p}} \mathtt{s}) = b_1 b_2 b_3 b_4$$

- $b_1 = 1$: distance between synchronizations is bounded by k,
- $b_2 = 1$: from some point onwards, the distance between synchronizations is bounded by k,
- $b_3 = 1$: there are infinitely many synchronizations, and
- $b_4 = 1$: there is at least one synchronization.

rPrompt-LTL

Syntax

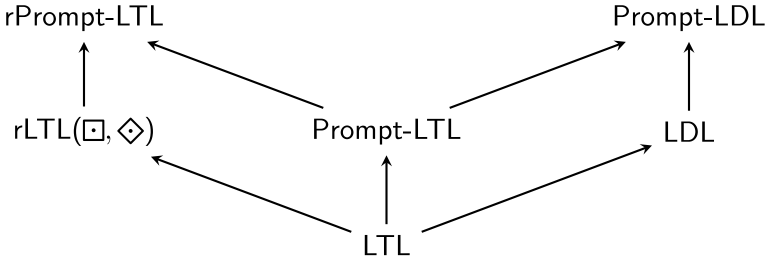
$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \boxdot\!\!\!\!\diamond\, \varphi \mid \boxdot \varphi \mid \boxdot\!\!\!\!\diamond_{\mathbf{p}}\, \varphi$$

Theorem

For every rPrompt-LTL formula φ and every truth value $\beta \in \mathbb{B}_4$, there is a Prompt-LTL formula φ_β of size $\mathcal{O}(|\varphi|)$ such that $V^{\mathrm{RP}}(w, k, \varphi) \geq \beta$ if and only if $V^{\mathrm{P}}(w, k, \varphi_\beta) = 1$.

Hence, rPrompt-LTL has the ECP, i.e., model checking is still in PSPACE and synthesis is still in 2EXPTIME!

The Big Picture



rLDL

Syntax

Add dots to LDL operators.

Semantics

$V^{\mathrm{RD}}(w, [\cdot r \cdot]\, \mathtt{a})$ in case r has infinitely many matches in w:

1. a holds at every match	1111
2. a holds at almost all matches	0111
3. a holds at infinitely many matches	0011
4. a holds at some match	0001
5. a holds at no match	0000

Additionally: rules for case of finitely many matches.

rLDL

Syntax

Add dots to LDL operators.

Example

- $[\cdot r \cdot]\, \mathtt{q} \rightarrow [\cdot \mathtt{tt}\,;\, r \cdot]\, \mathtt{p})$ with $r = (\mathtt{tt}\,;\,\mathtt{tt})^*$: the level of satisfaction of $\mathtt{p}$ at odd positions is at least as large as the level of satisfaction of $\mathtt{q}$ at even positions.

rLDL

Syntax

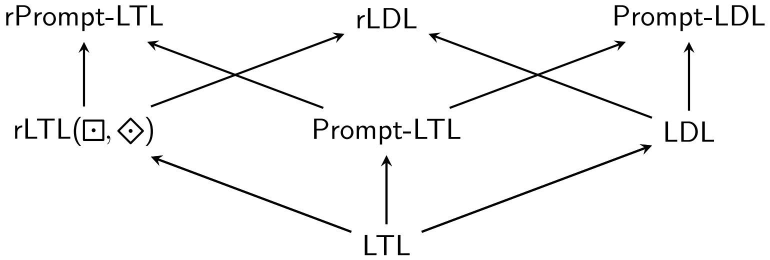
Add dots to LDL operators.

Theorem

Let φ be an rLDL formula, $n = |\varphi|$, and $\beta \in \mathbb{B}_4$. There is a non-deterministic Büchi automaton with $2^{\mathcal{O}(n \log n)}$ states recognizing the language $\{w \in (2^P)^\omega \mid V^{\mathrm{RD}}(w, \varphi) \geq \beta\}$.

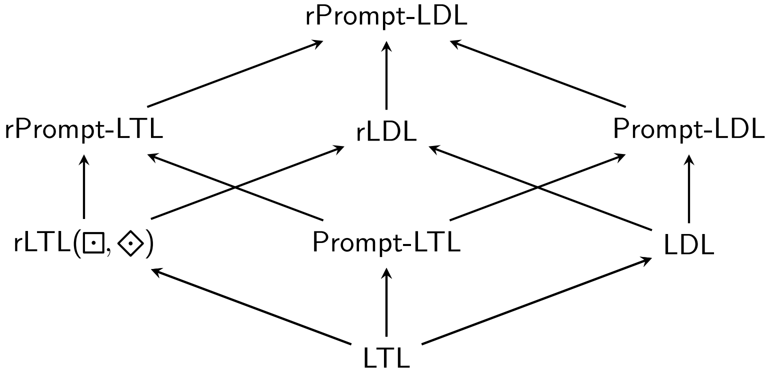
Hence, rLDL has the ECP, i.e., model checking is still in $\textsc{PSpace}$ and synthesis is still in $2\textsc{ExpTime}$!

The Big Picture



All these logics have the ECP, i.e., model checking is still in PSPACE and synthesis is still in 2EXPTIME!

The Big Picture



Open problem what about the combination of all three extensions?