The Tail-Recursive Fragment of Timed Recursive CTL

Florian Bruse[1] Martin Lange[1] Etienne Lozes[2]

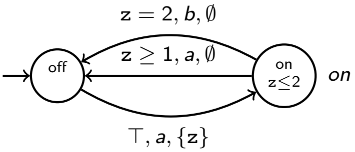
[1] University of Kassel, Germany
[2] Université Cote d'Azur, France

29th Int. Symp. on Temporal Representation and Reasoning, TIME'22
07/11 - 09/11/2022

Some Extensions of CTL

Alur/Courcoubetis/Dill, LICS'90: Timed CTL (TCTL)

- interpreted over real-time systems (e.g. timed automata)
- model checking: PSPACE-complete



Bruse/L., TIME'20: Recursive CTL (RecCTL)

- extension of discrete-time CTL for greater (non-regular) expressiveness
- model checking: EXPTIME-complete

Bruse/L., TIME'21: Timed Recursive CTL (TRecCTL)

- ... for greater expressiveness over real-time systems
- model checking: 2EXPTIME-complete

Is it possible to have high expressiveness over real-time systems and slightly lower model checking complexity? ... Yes!

(Real-Time) CTL with Recursion

syntax of TRecCTL:

$$\varphi \ ::= \ q \mid x \mid \mathtt{z} < r \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathtt{E}(\varphi\, \mathtt{U}^J\, \varphi) \mid \Phi(\varphi_1, \ldots, \varphi_k)$$
$$\Phi \ ::= \ \mathcal{F} \mid \mathtt{rec}\, \mathcal{F}(x_1, \ldots, x_k).\varphi$$

with

- q an atomic proposition, e.g. "*the traffic light is red*"
- x and $\mathcal{F}$ formula variables
- $\mathtt{z}$ a clock, $r \in \mathbb{Q}$
- J an interval over $\mathbb{Q}$

two types of formulas:

- φ: propositional formulas interpreted as sets of states, e.g. $\mathtt{AG}^{>3}(\mathsf{req} \rightarrow \mathtt{EF}^{\leq 1}\mathsf{grnt})$
- Φ: first-order formulas interpreted as functions mapping multiple sets to one set

Example

TRecCTL can express the property "*whenever a request can be issued within time* $2n$*, then a grant can be done within time* $3n$ (for any $n \in \mathbb{N}$)"

let $\Phi := \mathtt{rec}\,\mathcal{F}(x,y).(x \rightarrow y) \wedge \mathcal{F}(\mathrm{EF}^{\leq 2}x, \mathrm{EF}^{\leq 3}y)$

the property above is formalised by $\Phi(\mathsf{req}, \mathsf{grnt})$

main tool for understanding complex formulas: unfolding+β-reduction into potentially infinitary TCTL-formula:

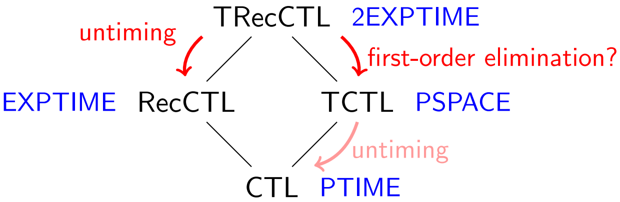
$$\begin{aligned}
\Phi(\mathsf{req}, \mathsf{grnt}) &\equiv \big((x \rightarrow y) \wedge \Phi(\mathrm{EF}^{\leq 2}x, \mathrm{EF}^{\leq 3}y)\big)(\mathsf{req}, \mathsf{grnt}) \\
&\equiv (\mathsf{req} \rightarrow \mathsf{grnt}) \wedge \Phi(\mathrm{EF}^{\leq 2}\mathsf{req}, \mathrm{EF}^{\leq 3}\mathsf{grnt}) \\
&\equiv (\mathsf{req} \rightarrow \mathsf{grnt}) \wedge (\mathrm{EF}^{\leq 2}\mathsf{req} \rightarrow \mathrm{EF}^{\leq 3}\mathsf{grnt}) \wedge \Phi(\mathrm{EF}^{\leq 2}\mathrm{EF}^{\leq 2}\mathsf{req}, \mathrm{EF}^{\leq 3}\mathrm{EF}^{\leq 3}\mathsf{grnt}) \\
&\equiv (\mathsf{req} \rightarrow \mathsf{grnt}) \wedge (\mathrm{EF}^{\leq 2}\mathsf{req} \rightarrow \mathrm{EF}^{\leq 3}\mathsf{grnt}) \wedge \Phi(\mathrm{EF}^{\leq 4}\mathsf{req}, \mathrm{EF}^{\leq 6}\mathsf{grnt}) \\
&\equiv \ldots \equiv \bigwedge_{n \geq 0} (\mathrm{EF}^{\leq 2n}\mathsf{req} \rightarrow \mathrm{EF}^{\leq 3n}\mathsf{grnt})
\end{aligned}$$

Model Checking Timed Recursive CTL

Proposition 1 (Bruse/L., TIME'21)

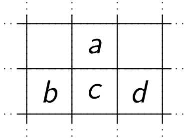
Model Checking TRecCTL (over timed systems represented as timed automata) is 2EXPTIME-complete

upper bound using untiming construction: exponential reduction to exponential model checking of discrete-time RecCTL



lower bound: describe behaviour of det. 2EXPTIME machine:

$\texttt{rec}\, \mathcal{F}_a(s,t). \ldots \mathcal{F}_b(s-1,t-1) \land \mathcal{F}_c(s,t-1) \land \mathcal{F}_d(s+1,t-1) \ldots$



An Attempt to Eliminate First-Order Formulas

reconsider $\big(\texttt{rec}\,\mathcal{F}(x,y).(x \to y) \wedge \mathcal{F}(\texttt{EF}^{\leq 2}x, \texttt{EF}^{\leq 3}y)\big)(\text{req},\text{grnt})$ from above

idea:

- replace $\mathcal{F}(x,y)$ with formal parameters x, y with $\mathcal{F}_{x,y}$ for suitably many actual parameters x, y
- connect them all using simultaneous fixpoint definitions (syntactic sugar)

$$\texttt{rec}\,\mathcal{F}_0.\begin{pmatrix} \mathcal{F}_0 & . & (\text{req} \to \text{grnt}) \wedge \mathcal{F}_1 \\ & \vdots & \\ \mathcal{F}_i & . & (\texttt{EF}^{\leq 2i}\text{req} \to \texttt{EF}^{\leq 3i}\text{grnt}) \wedge \mathcal{F}_{i+1} \\ & \vdots & \end{pmatrix}$$

three problems:

1. obviously not a finite formula $\rightsquigarrow$ not a problem: we're doing model checking!
2. would only work when there is no fixpoint nesting (as in $\ldots \mathcal{F}(q \wedge \mathcal{F}x)$)
3. the result would not be TCTL (but of the timed μ-calculus – with EXPTIME model checking!)

Tail-Recursive Formulas

Obs.: a fragment with genuinely lower model checking complexity than 2EXPTIME will have to restrict recursion structurally $\rightsquigarrow$ the tail-recursive fragment

tail-recursive formulas intuitively:

- free variables in $\bar{\psi}$ amongst $\bar{x}$ in $\texttt{rec}\,\mathcal{F}(\bar{x}).\ldots\mathcal{F}(\bar{\psi})$
  $\rightsquigarrow$ no $\ldots\mathcal{F}(q \land \mathcal{F}x)\ldots$
- recursive formulas are aconjunctive: $\mathcal{F}$ occurs not in both ψ_1, ψ_2 in $\mathcal{F}(\bar{x}).\ldots(\psi_1 \land \psi_2)$
  $\rightsquigarrow$ top-down model checking can avoid backtracking for first-order formulas
- similar restriction for temporal formulas

formal definition via simple type system:

$$\frac{}{\emptyset \vdash_{\mathrm{tr}} p} \qquad \frac{}{\emptyset \vdash_{\mathrm{tr}} x} \qquad \frac{}{\emptyset \vdash_{\mathrm{tr}} \chi} \; \frac{}{\{\mathcal{F}\} \vdash_{\mathrm{tr}} \mathcal{F}} \qquad \frac{\emptyset \vdash_{\mathrm{tr}} \varphi}{\emptyset \vdash_{\mathrm{tr}} \neg\varphi} \qquad \frac{\mathcal{V} \vdash_{\mathrm{tr}} \varphi_1 \quad \mathcal{V}' \vdash_{\mathrm{tr}} \varphi_2}{\mathcal{V} \cup \mathcal{V}' \vdash_{\mathrm{tr}} \varphi_1 \lor \varphi_2} \qquad \frac{\mathcal{V} \vdash_{\mathrm{tr}} \varphi}{\mathcal{V} \setminus \{\mathcal{F}\} \vdash_{\mathrm{tr}} \texttt{rec}\,\mathcal{F}(x_1,\ldots,x_m).\varphi}$$

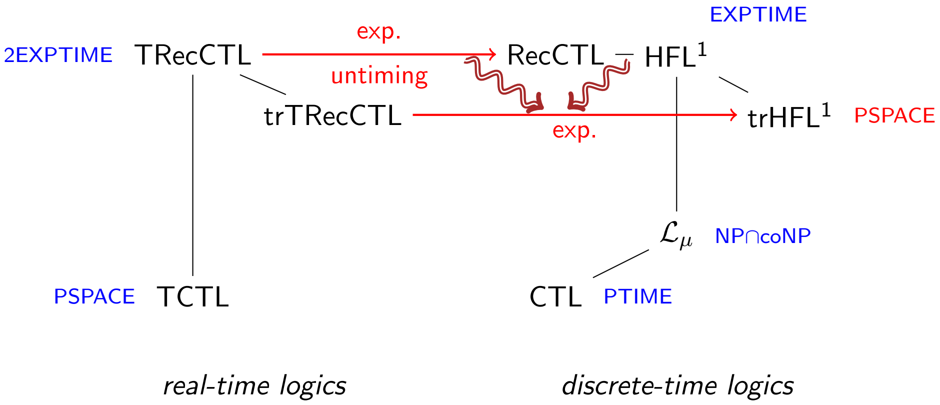
$$\frac{\emptyset \vdash_{\mathrm{tr}} \varphi_1 \quad \mathcal{V} \vdash_{\mathrm{tr}} \varphi_2}{\mathcal{V} \vdash_{\mathrm{tr}} \varphi_1 \land \varphi_2} \qquad \frac{\emptyset \vdash_{\mathrm{tr}} \varphi_1 \quad \mathcal{V} \vdash_{\mathrm{tr}} \varphi_2}{\mathcal{V} \vdash_{\mathrm{tr}} \mathtt{E}(\varphi_1 \,\mathtt{U}^J\, \varphi_2)} \qquad \frac{\emptyset \vdash_{\mathrm{tr}} \varphi_1 \quad \emptyset \vdash_{\mathrm{tr}} \varphi_2}{\emptyset \vdash_{\mathrm{tr}} \mathtt{A}(\varphi_1 \,\mathtt{U}^J\, \varphi_2)} \qquad \frac{\mathcal{V} \vdash_{\mathrm{tr}} \Phi \quad \emptyset \vdash_{\mathrm{tr}} \varphi_1 \quad \cdots \quad \emptyset \vdash_{\mathrm{tr}} \varphi_n}{\mathcal{V} \vdash_{\mathrm{tr}} \Phi(\varphi_1,\ldots,\varphi_n)}$$

Model Checking Tail-Recursive TRecCTL: Upper Bound

Theorem 1

The model checking problem for tail-recursive TRecCTL is in EXPSPACE.

PROOF: using the untiming construction wisely

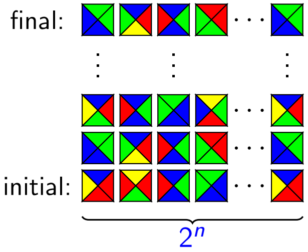


Model Checking Tail-Recursive TRecCTL: Lower Bound

Theorem 2

The model checking problem for tail-recursive TRecCTL is EXPSPACE-hard.

PROOF: by reduction from the $(2^n \times \infty)$-tiling problem



existence of successful tiling expressed by

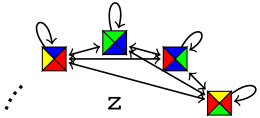
$$\Big(\texttt{rec}\, \mathcal{F}(r).\mathit{fin}(r) \lor \exists r'.\mathit{match}(r, r') \land \mathcal{F}(r')\Big)(\mathit{init})$$

over suitable timed automaton s.t. proposition r can encode a row

Encoding Rows

remember: propositions interpreted in $\mathcal{P}(Loc \times (Clocks \to \mathbb{R}^{\geq 0}))$

consider TA



Obs.: sets of the form $\{(t_0, 0), (t_1, 1), \ldots, (t_{2^n-1}, 2^n - 1)\}$ can be maintained, and they naturally encode a (potential) row!

$$isRow(r) := \mathtt{AG}\Big(r \to \big(\mathtt{rec}\,\mathcal{F}().\mathtt{z}{=}2^n{-}1 \vee \mathtt{EF}^{=1}\mathcal{F}()\big) \wedge \big(\bigvee_{t \in T} t \wedge \bigwedge_{t' \neq t} \neg t'\big) \wedge \big(\mathtt{z} < 2^n{-}1 \to \bigwedge_{t \in T} t \to \bigvee_{\substack{t' \in T \\ (t,t') \in H}} \mathtt{EF}^{=1} t'\big)\Big)$$

Formalising Existential Quantification over Rows

Obs.: rows are naturally enumerable as base-$|T|$-numbers (with least significant bit at position 2^{n-1})

Ex.: $T = \{$  (0),  (1),  (2) $\}$, $n = 2$, i.e. $2^n - 1 = 3$

$\{($$^{2}, 0), ($$^{2}, 1), ($$^{0}, 2), ($$^{1}, 3)\} \triangleq 2 \cdot 3^{3-0} + 2 \cdot 3^{3-1} + 0 \cdot 3^{3-2} + 1 \cdot 3^{3-3} = 73$

Exc.: suppose r encodes $\widehat{r} \in \{0, \ldots, |T|^{2^n} - 1\}$. Write formula $next(r)$ s.t. $\widehat{next(r)} = \widehat{r} + 1$ by formalising digit-wise base-$|T|$-increments!

do existential quantification by enumeration of all candidates (in a tail-recursive way!):

$$\exists r'.\varphi(r') \;:=\; \big(\mathtt{rec}\, \mathcal{F}(x).\varphi(x) \vee \mathcal{F}(next(x))\big)(zero)$$

Conclusion

summary:

- TRecTCTL is a highly expressive specification logic for real-time systems
- model checking becomes more space efficient when recursion structure is restricted accordingly
- this is in line with findings on similar (discrete-time) temporal logics
- upper bounds unlikely to be improvable without giving up a lot of expressiveness: 2EXPTIME-, resp. EXPSPACE-hardness hold for very simple TA over a single clock already

outlook:

- whole other story: how to handle such high complexity in practice . . .
- decidability over extensions of plain timed automata

The End