Relaxing safety for metric first-order temporal logic (MFOTL) via dynamic free variables

Jonathan Julián Huerta y Munive

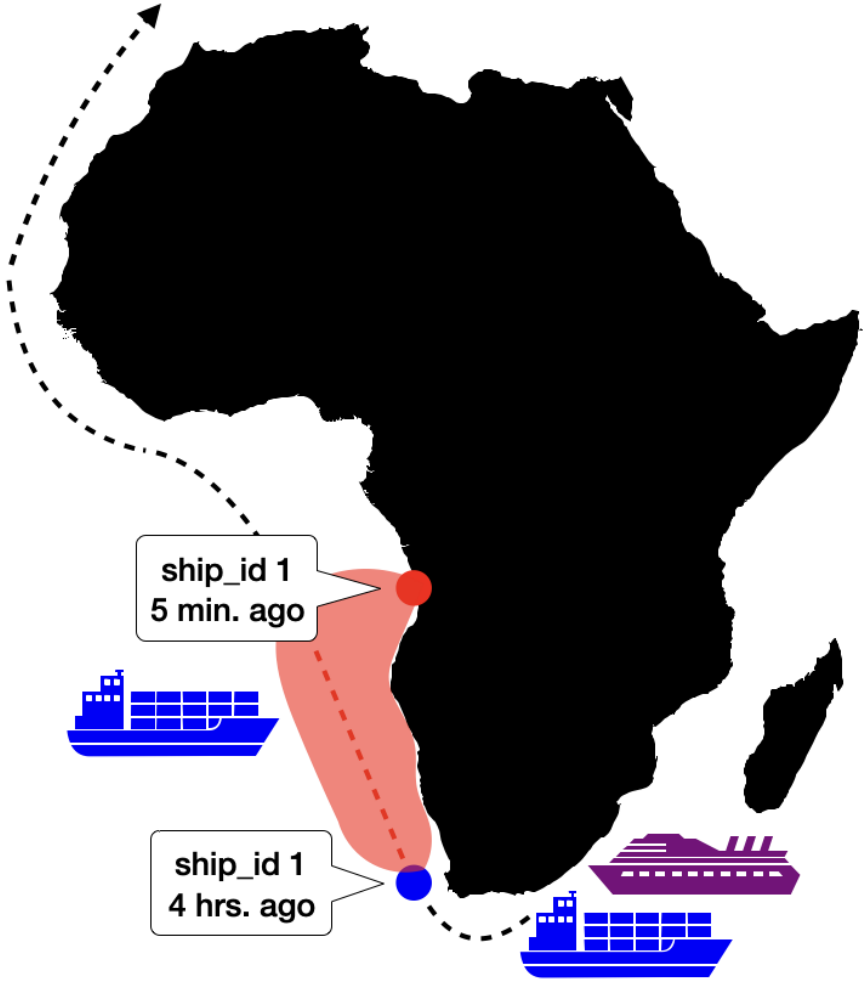
University of Copenhagen, Denmark

September 29, 2022

Motivation



time	ship_id 1	ship_id 2
0	no_sign	no_sign
1	no_sign	no_sign
2	no_sign	no_sign
3	off_route	no_sign
4	off_route	no_sign

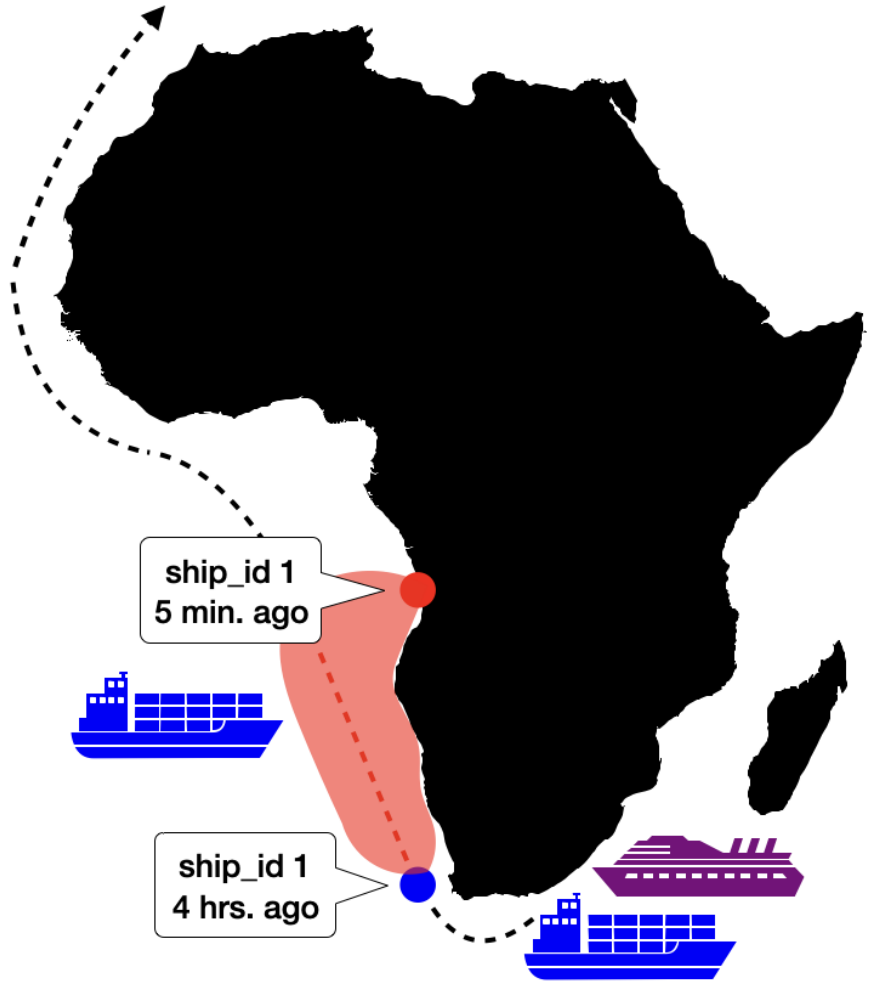


Motivation

$$\texttt{pirated} \stackrel{\text{def}}{=} (\texttt{off_route}\,[\mathbf{v}\,x])\;\mathrm{R}_{[0,n)}\;(\texttt{no_sign}\,[\mathbf{v}\,x])$$



time	ship_id 1	ship_id 2
0	no_sign	no_sign
1	no_sign	no_sign
2	no_sign	no_sign
3	off_route	no_sign
4	off_route	no_sign



$$\models_{\langle v,i\rangle} \alpha\,\mathrm{R}_I\,\beta \iff \left(\forall j \geq i.\ (\tau_j - \tau_i) \in I \implies \models_{\langle v,j\rangle} \beta \vee \left(\exists k \in [i,j).\ \models_{\langle v,i\rangle} \alpha\right)\right)$$

Motivation

$$\texttt{pirated} \equiv \neg\left(\left(\neg\,\texttt{off_route}\,[\mathbf{v}\,x]\right)\,\mathrm{U}_{[0,n)}\,\left(\neg\,\texttt{no_sign}\,[\mathbf{v}\,x]\right)\right)$$

```
pzx681@SCI1012595 monpoly % ./monpoly -sig examples/sigs/release.sig -formula
examples/mfotl/release.mfotl -log examples/logs/release.log -verbose
The analyzed formula is:
  NOT ((NOT offroute(a)) UNTIL[0,2] NOT nosign(a))
The sequence of free variables is: (a)
The analyzed formula is NOT monitorable, because of the subformula:
  NOT ((NOT offroute(a)) UNTIL[0,2] NOT nosign(a))
Subformulas of the form NOT psi should contain no free variables (except when
they are part of subformulas of the form phi AND NOT psi, NOT psi SINCE_I phi,
 or NOT psi UNTIL_I phi).
The analyzed formula is neither safe-range.
By the way, the analyzed formula is not TSF safe-range.
```

$$\models_{\langle v,i\rangle} \alpha\,\mathrm{R}_I\,\beta \iff \left(\forall j \geq i.\ (\tau_j - \tau_i) \in I \implies \models_{\langle v,j\rangle} \beta \vee \left(\exists k \in [i,j).\ \models_{\langle v,i\rangle} \alpha\right)\right)$$

Motivation

$$\texttt{pirated} \equiv \neg\left(\left(\neg\,\texttt{off_route}\,[\mathbf{v}\,x]\right)\,\mathrm{U}_{[0,n)}\,\left(\neg\,\texttt{no_sign}\,[\mathbf{v}\,x]\right)\right)$$

Other formulas we would like to monitor directly: $\mathrm{G}_I\,\alpha \quad \mathrm{H}_I\,\alpha \quad \alpha\,\mathrm{T}_I\,\beta$

```
pzx681@SCI1012595 monpoly % ./monpoly -sig examples/sigs/release.sig -formula
examples/mfotl/release.mfotl -log examples/logs/release.log -verbose
The analyzed formula is:
  NOT ((NOT offroute(a)) UNTIL[0,2] NOT nosign(a))
The sequence of free variables is: (a)
The analyzed formula is NOT monitorable, because of the subformula:
  NOT ((NOT offroute(a)) UNTIL[0,2] NOT nosign(a))
Subformulas of the form NOT psi should contain no free variables (except when
they are part of subformulas of the form phi AND NOT psi, NOT psi SINCE_I phi,
 or NOT psi UNTIL_I phi).
The analyzed formula is neither safe-range.
By the way, the analyzed formula is not TSF safe-range.
```

Can we make these formulas monitorable?

$$\models_{\langle v,i\rangle}\,\alpha\,\mathrm{R}_I\,\beta \iff \left(\forall j \geq i.\ (\tau_j - \tau_i) \in I \implies \models_{\langle v,j\rangle}\,\beta \vee \left(\exists k \in [i,j).\ \models_{\langle v,i\rangle}\,\alpha\right)\right)$$

Contributions

1. New fragment of metric first-order temporal logic (MFOTL) that guarantees monitorability
2. Addition of dual operators trigger and release to MFOTL-Verimon
3. Formal verification of correctness of our extensions in Isabelle/HOL

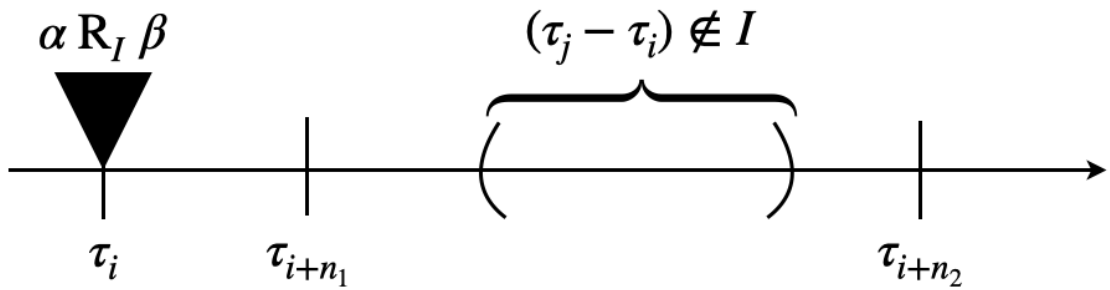


https://bitbucket.org/jshs/monpoly/src/b4b63034eca0ccd5783085dececddb6c47cf6f52/thys/Relax_Safety/MFOTL/

Why are they not monitorable?

- Intuitively, monitor outputs $[\![\alpha]\!]_i = \{v \mid \models_{\langle v,i\rangle} \alpha\}$ at each i.
- However, $[\![\alpha \,\mathrm{R}_I\, \beta]\!]_i$ might be infinite because:

$$\models_{\langle v,i\rangle} \alpha \,\mathrm{R}_I\, \beta \iff \left(\forall j \geq i.\ (\tau_j - \tau_i) \in I \implies \models_{\langle v,j\rangle} \beta \vee \left(\exists k \in [i,j)\,.\ \models_{\langle v,i\rangle} \alpha\right)\right)$$



- Other sources of non-monitorability (assuming infinite domain):
  - Negations: if $[\![\alpha]\!]_i$ is finite, then $[\![\neg\, \alpha]\!]_i$ is infinite.
  - Disjunctions: if $\mathsf{fv}\,\alpha \neq \mathsf{fv}\,\beta$, then $[\![\alpha \vee \beta]\!]_i$ is infinite.
  - (In)equalities: $[\![x = y]\!]_i$, $[\![x < y]\!]_i$ and $[\![x \leq y]\!]_i$ are infinite

Our solution: first part

Implement the dual-operators directly into the monitor.

$$\alpha \ ::= p\,[t_1, \ldots, t_n] \mid t = t \mid \neg\,\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \exists\,\alpha \mid$$
$$\mathrm{Y}_I\,\alpha \mid \mathrm{X}_I\,\alpha \mid \alpha\,\mathrm{S}_I\,\alpha \mid \alpha\,\mathrm{U}_I\,\alpha \mid \alpha\,\mathrm{T}_\mathbf{I}\,\alpha \mid \alpha\,\mathrm{R}_\mathbf{I}\,\alpha,$$

For example, the semantics of the release operator

$$\models_{\langle v,i\rangle} \alpha\,\mathrm{R}_I\,\beta \iff \left(\forall j \geq i.\ (\tau_j - \tau_i) \in I \implies \models_{\langle v,j\rangle} \beta \vee \left(\exists k \in [i,j)\,.\ \models_{\langle v,i\rangle} \alpha\right)\right),$$

correspond to the operations

$$[\![\alpha\,\mathrm{R}_I\,\beta]\!]_i = \bigcap_{j \in \uparrow_I i} [\![\beta]\!]_j \cup \left(\bigcup_{k \in [i,j)} \mathit{rel}_k \cap [\![\alpha]\!]_k\right),$$

where $\mathit{rel}_k \stackrel{\text{def}}{=} \text{if } 0 \in I \text{ then } [\![\beta]\!]_k \text{ else } [\![\alpha]\!]_k$, and
$\uparrow_I i \stackrel{\text{def}}{=} \{j \geq i \mid (\tau_j - \tau_i) \in I\}$.

The problem persists: we have to declare $\alpha\,\mathrm{R}_I\,\beta$ unsafe whenever $0 \notin I$

Observation

Monitor's outputs are tables: $[\![\alpha]\!]_i = \{v \mid \models_{\langle v,i\rangle} \alpha \wedge \texttt{wf_tuple}\ n\ X\ v\}$

$[\![\alpha]\!]_i$

true at i

x_1	x_2	$\cdots$	x_n
nan	*nan*	$\cdots$	*nan*

$= \texttt{unit_table}\ n$
$\texttt{table}\ n\ \emptyset\ (\texttt{unit_table}\ n)$

false at i

x_1	x_2	$\cdots$	x_n

$= \texttt{empty_table} = \emptyset$
$\texttt{table}\ n\ X\ (\texttt{empty_table})$

satisfiable & finite at i

x_1	x_2	$\cdots$	x_n
a_{11}	a_{12}	$\cdots$	a_{1n}
$\vdots$	$\vdots$	$\ddots$	$\vdots$
a_{m1}	a_{m2}		a_{mn}

$\texttt{table}\ n\ (\mathrm{fv}\,\alpha)\ R$ with
$\mathrm{fv}\,\alpha = \{x_1, \ldots, x_n\}$

satisfiable & infinite at i

x_1	x_2	$\cdots$	x_n
$\vdots$	$\vdots$	$\vdots$	$\vdots$

$\texttt{table}\ n\ X\ R \iff$ every $v \in R$ satisfies $\texttt{wf_tuple}\ n\ X\ v$

$\texttt{wf_tuple}\ n\ X\ v \iff$ if length $v = n$ and for every $k < n$, $v_k = \mathit{nan}$ iff $k \notin X$

Our solution: second part

Define the set of *dynamic free variables* (dfv_i variables that determine satisfiability of α at i) and the *safe sets of free variables* (ssfv set of all dfvs over time) of a formula α, such that

	$[\![\alpha]\!]_i$	$dfv_i\,\alpha$	$ssfv\,\alpha$
true at i	$\begin{array}{\|cccc\|}\hline x_1 & x_2 & \cdots & x_n \\ nan & nan & \cdots & nan \\ \hline\end{array}$	$\emptyset$	$\emptyset \in ssfv\,\alpha$ *
false at i	$\begin{array}{\|cccc\|}\hline x_1 & x_2 & \cdots & x_n \\ & & & \\ \hline\end{array}$	$\emptyset$ *or* $fv\,\alpha$	$\emptyset \in ssfv\,\alpha$ * *or* $fv\,\alpha \in ssfv\,\alpha$
satisfiable & finite at i	$\begin{array}{\|cccc\|}\hline x_1 & x_2 & \cdots & x_n \\ a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & & a_{mn} \\ \hline\end{array}$	$fv\,\alpha$	$fv\,\alpha \in ssfv\,\alpha$ *
satisfiable & infinite at i	$\begin{array}{\|cccc\|}\hline x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & \vdots & \vdots \\ \hline\end{array}$	$fv\,\alpha$	$ssfv\,\alpha = \emptyset$ * * $fv\,\alpha = \{x_1, \ldots, x_n\}$

* there is no algorithm deciding finiteness for any given formula

Our solution: second part

Define the set of *dynamic free variables* (dfv_i variables that determine satisfiability of α at i) and the *safe sets of free variables* (ssfv set of all dfvs over time) of a formula α, such that

- $\mathsf{dfv}_i\ \alpha \subseteq \mathsf{fv}\ \alpha$ for all time-points i
- $(\bigcup \mathsf{ssfv}\ \alpha) \subseteq \mathsf{fv}\ \alpha$
- $\mathsf{dfv}_i\ \alpha \in \mathsf{ssfv}\ \alpha$ for all time-points i, if *is-safe* α
- if for each $x \in \mathsf{dfv}_i\ \alpha$ it holds that $v!x = v'!x$, then $\models_{\langle v,i\rangle} \alpha$ iff $\models_{\langle v',i\rangle} \alpha$
- *is-safe* $\alpha \overset{\text{def}}{\iff} \mathsf{ssfv}\ \alpha \neq \emptyset$
- *Example*: if $\mathsf{dfv}_i\ (\alpha\ \mathrm{R}_I\ \beta) = \emptyset$ and $\mathsf{dfv}_j\ (\alpha\ \mathrm{R}_I\ \beta) = \mathsf{fv}\ (\alpha\ \mathrm{R}_I\ \beta)$ with $i \neq j$, then $\{\emptyset, \mathsf{fv}\ (\alpha\ \mathrm{R}_I\ \beta)\} \subseteq \mathsf{ssfv}(\alpha\ \mathrm{R}_I\ \beta)$

Our solution: second part

Define the set of *dynamic free variables* (dfv_i variables that determine satisfiability of α at i) and the *safe sets of free variables* (ssfv set of all dfvs over time) of a formula α, such that

$$
\begin{aligned}
&| \ \mathrm{dfv}_i\ (\alpha\ \mathbf{R}\ I\ \beta) = \\
&\quad (\mathit{let}\ \mathit{satisf\text{-}at} = \lambda v\, j.\ \langle \sigma, v, j\rangle \models \beta \vee (\exists\, k{\in}\{i..{<}j\}.\ \langle \sigma, v, k\rangle \models \alpha)\ \mathit{in} \\
&\qquad (\mathit{if} \uparrow_I\ i = \{\}\ \mathit{then}\ \{\} \\
&\qquad \mathit{else\ if}\ (\forall\, v.\ \exists\, j{\in}\uparrow_I\ i.\ \neg\ \mathit{satisf\text{-}at}\ v\ j)\ \mathit{then}\ \mathrm{fv}\ (\alpha\ \mathbf{R}\ I\ \beta) \\
&\qquad \mathit{else}\ (\mathit{let}\ J = \{j{\in}\uparrow_I\ i.\ \exists\, v.\ \langle \sigma, v, j\rangle \models \beta\}; \\
&\qquad\quad K = \{k.\ \exists\, v.\ \exists\, j{\in}\uparrow_I\ i.\ i \le k \wedge k < j \wedge \langle \sigma, v, k\rangle \models \alpha\}\ \mathit{in} \\
&\qquad\quad (\bigcup \{(\mathrm{dfv}_k\ \alpha) | k.\ k \in K\}) \cup (\bigcup \{(\mathrm{dfv}_j\ \beta) | j.\ j \in J\})))
\end{aligned}
$$

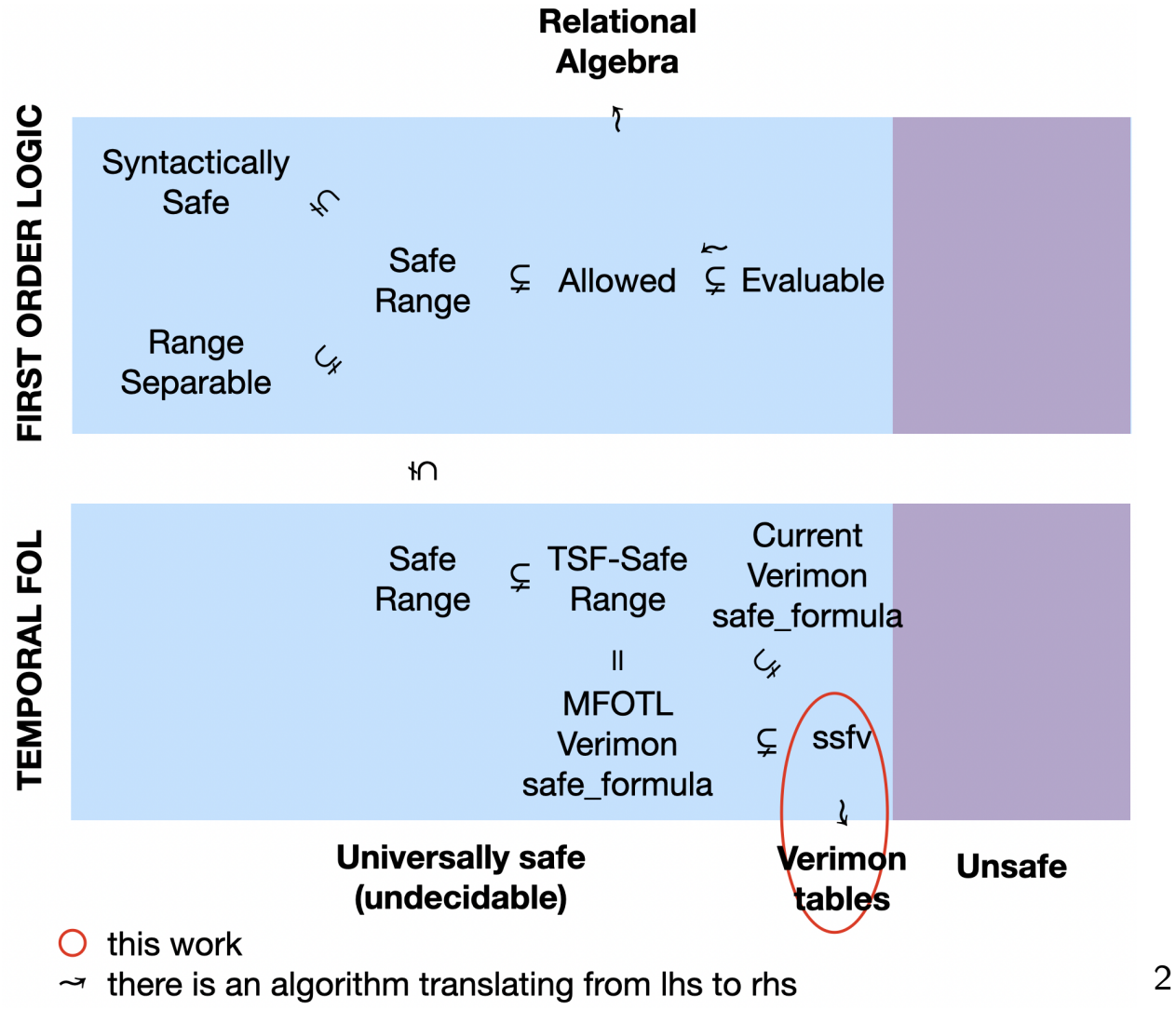
$$
\begin{aligned}
&| \ \mathit{ssfv}\ (\alpha\ \mathbf{R}\ I\ \beta) = (\mathit{let}\ \mathcal{A} = \mathit{ssfv}\ \alpha;\ \mathcal{B} = \mathit{ssfv}\ \beta;\ X = \mathrm{fv}\ \alpha;\ Y = \mathrm{fv}\ \beta\ \mathit{in} \\
&\quad \mathit{if}\ 0 \in I\ \mathit{then} \\
&\qquad \mathit{if}\ (\mathcal{B} = \{Y\})\ \mathit{then} \\
&\qquad\quad \mathit{if}\ \mathcal{A} \neq \{\} \wedge X \subseteq Y\ \mathit{then}\ \{Y\} \\
&\qquad\quad \mathit{else}\ (\mathit{case}\ \alpha\ \mathit{of} \\
&\qquad\qquad\quad \neg_F\ \alpha' \Rightarrow (\mathit{let}\ \mathcal{A}' = \mathit{ssfv}\ \alpha'\ \mathit{in}\ \mathit{if}\ \mathcal{A}' \neq \{\} \wedge X \subseteq Y\ \mathit{then}\ \{Y\}\ \mathit{else}\ \{\}) \\
&\qquad\qquad |\quad \text{-} \Rightarrow \{\}) \\
&\qquad \mathit{else}\ \{\} \\
&\quad \mathit{else} \\
&\qquad \mathit{if}\ X = Y \wedge \mathcal{A} = \{X\} \wedge \mathcal{B} = \{Y\}\ \mathit{then}\ \{\{\},X\}\ \mathit{else}\ \{\})
\end{aligned}
$$

Properties of our contribution

- Expressive monitor
  - Strictly larger safety fragment for MFOTL
    - Verimon's *safe-formula* α implies our *is-safe* α for all α, and
    - there is a formula β such that *is-safe* β but it is not true that *safe-formula* β
  - Before: $\texttt{pirated}' \stackrel{\text{def}}{\equiv} (\texttt{no_sign}\,[\mathbf{v}\,x]\ \mathrm{U}_{[n,2n-1)}\ (\mathrm{Y}_{[0,\infty)}\ \texttt{no_sign}\,[\mathbf{v}\,x])) \vee (\texttt{no_sign}\,[\mathbf{v}\,x]\ \mathrm{U}_{[0,b)}\ (\texttt{no_sign}\,[\mathbf{v}\,x] \wedge (\mathrm{X}_{[b,\infty)}\ \top))) \vee (\texttt{no_sign}\,[\mathbf{v}\,x])\ \mathrm{U}_{[0,\infty)}\ (\texttt{no_sign}\,[\mathbf{v}\,x] \wedge \texttt{off_route}\,[\mathbf{v}\,x])$[1]
    Now: $\texttt{pirated} \equiv (\texttt{off_route}\,[\mathbf{v}\,x])\ \mathrm{R}_{[0,n)}\ (\texttt{no_sign}\,[\mathbf{v}\,x])$
- Trustworthy monitor
  - Implemented and formally verified in Isabelle/HOL
  - Correctness: if *is-safe* α, monitor's output is equal to $[\![\alpha]\!]_i = \{v \mid \models_{\langle v,i\rangle} \alpha \wedge \texttt{wf_tuple}\ (\#\,\mathrm{fv}\,\alpha)\ (\mathrm{dfv}_i\ \alpha)\ v\}$ for all time-points i

[1] Hauser, "Safe Evaluation of MFOTL Dual Temporal Operators"

Context of the contribution



2

[2] Abiteboul, Hull, and Vianu, *Foundations of Databases*
Basin et al., "Monitoring Metric First-Order Temporal Properties"
Chomicki, Toman, and Böhlen, "Querying ATSQL databases with temporal logic"
Gelder and Topor, "Safety and Translation of Relational Calculus Queries"
Schneider et al., "A Formally Verified Monitor for MFOTL"

Future work

- Further relaxation of safety:
  - Address each special-case individually: $\mathrm{H}_I\,\alpha$, $\mathrm{G}_I\,\alpha$, $t = t$
  - Replace unions in algorithm $(\cup)$ with generalised version to be able to monitor nested temporal operators $\mathrm{P}_I(p\,x \wedge \mathrm{H}_J\,q\,y)$
- Extension of the approach to metric first-order dynamic logic (MFODL)
- Integration of Martin Raszyk's method for translating (even some "unsafe") formulas into a safety fragment[3].



https://bitbucket.org/jshs/monpoly/src/b4b63034eca0ccd5783085dececddb6c47cf6f52/thys/Relax_Safety/MFOTL/

jonathan.munive@di.ku.dk

[3] Raszyk et al., "Practical Relational Calculus Query Evaluation"

References I


Abiteboul, Serge, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. ISBN: 0-201-53771-0. URL: http://webdam.inria.fr/Alice/.


Basin, David A. et al. "Monitoring Metric First-Order Temporal Properties". In: *J. ACM* 62.2 (2015), 15:1–15:45.


Chomicki, Jan, David Toman, and Michael H. Böhlen. "Querying ATSQL databases with temporal logic". In: *ACM Trans. Database Syst.* 26.2 (2001), pp. 145–178.


Gelder, Allen Van and Rodney W. Topor. "Safety and Translation of Relational Calculus Queries". In: *ACM Trans. Database Syst.* 16.2 (1991), pp. 235–278.

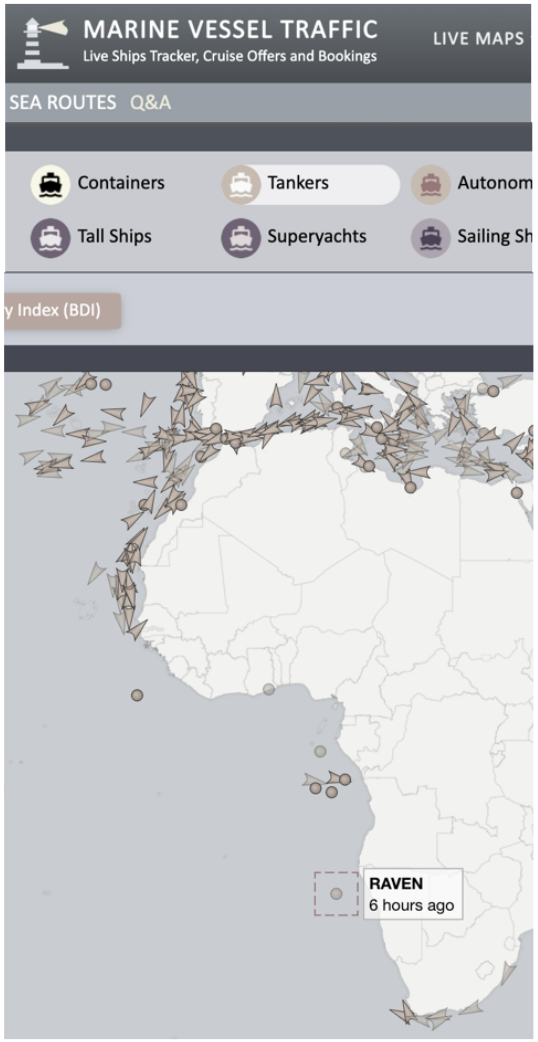

Hauser, Nico. "Safe Evaluation of MFOTL Dual Temporal Operators". ETH Zürich, 2021.


Raszyk, Martin et al. "Practical Relational Calculus Query Evaluation". In: *ICDT 2022*. Vol. 220. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 11:1–11:21.

References II

 Schneider, Joshua et al. “A Formally Verified Monitor for MFOTL”. In: *RV 2019*. Vol. 11757. LNCS. Springer, 2019, pp. 310–328.

Demo?





MarineVesselTraffic.com