Extending propositional separation logic for robustness properties

F·R·I·E·N·D·S of separation logic

Alessio Mansutti

LSV, CNRS, ENS Paris-Saclay

Paris - April 2019

What we will see

An extension of propositional separation logic that

- can express some interesting properties for program verification,
- is PSpace-complete,
- has very weak extensions that are Tower-hard.

A modal logic on trees that

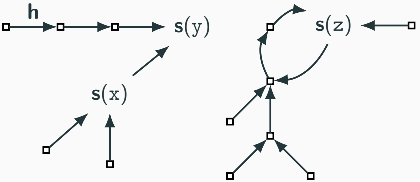
- is Tower-complete,
- it is very easily captured by logics that were independently found to be Tower-complete.

Memory states

Separation Logic is interpreted over **memory states** $(\mathbf{s}, \mathbf{h})$ where:

- **store**, $\mathbf{s} : \mathtt{VAR} \rightarrow \mathtt{LOC}$
- **heap**, $\mathbf{h} : \mathtt{LOC} \rightarrow_{\mathbf{fin}} \mathtt{LOC}$

where $\mathtt{VAR} = \{\mathrm{x}, \mathrm{y}, \mathrm{z}, \dots\}$ set of (program) variables,
$\mathtt{LOC}$ set of locations. $\mathtt{VAR}$ and $\mathtt{LOC}$ are countably infinite sets.



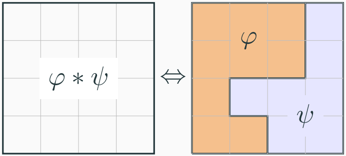
here, $\mathbf{h}(\mathbf{s}(\mathrm{x})) = \mathbf{s}(\mathrm{y})$

- Disjoint heaps: $\mathrm{dom}(\mathbf{h}_1) \cap \mathrm{dom}(\mathbf{h}_2) = \emptyset$
- Union of disjoint heaps $(\mathbf{h}_1 + \mathbf{h}_2)$: union of partial functions.

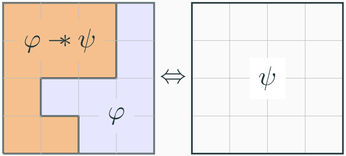
Propositional Separation Logic $\mathtt{SL}(*, \mathrel{-\!\!*})$

$$\varphi := \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathtt{emp} \mid \mathtt{x} = \mathtt{y} \mid \mathtt{x} \hookrightarrow \mathtt{y} \mid \varphi_1 * \varphi_2 \mid \varphi_1 \mathrel{-\!\!*} \varphi_2$$

$(\mathbf{s}, \mathbf{h}) \models \varphi * \psi$



$(\mathbf{s}, \mathbf{h}) \models \varphi \mathrel{-\!\!*} \psi$



Note: the satisfiability problem SAT($\mathtt{SL}(*, \mathrel{-\!\!*})$) is $\mathtt{PSpace}$-complete.

From where it started

Theorem (Demri, Lozes, M. – 2018, Fossacs)

$\mathtt{SL}(*, \mathrel{-\!\!*})$ *enriched with* $\mathtt{reach}(\mathtt{x}, \mathtt{y}) = 2$ *and* $\mathtt{reach}(\mathtt{x}, \mathtt{y}) = 3$ *is undecidable.*

- reduction from $\mathtt{SL}(\forall, \mathrel{-\!\!*})$ (Brochenin et al.'12)
- $\mathtt{SL}(*, \mathrel{-\!\!*}) + \mathtt{reach}(\mathtt{x}, \mathtt{y}) = 2$ is $\mathtt{PSpace}$-complete (Demri et al.'14)

Robustness Properties (Jansen, et al. – ESOP'17)

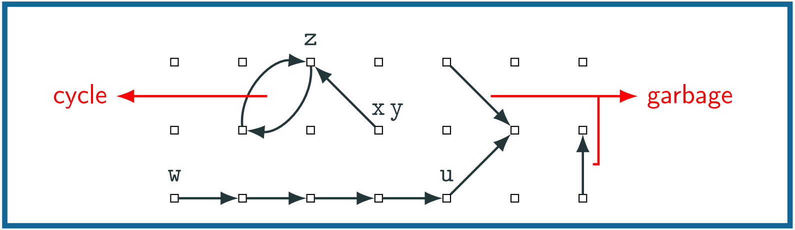
- φ comply with the **acyclicity** property iff every model of φ is acyclic.
- φ comply with the **garbage freedom** property iff in every model $(\mathbf{s}, \mathbf{h}) \models \varphi$, for each $\ell \in \mathrm{dom}(\mathbf{h})$ there is $\mathtt{x} \in \mathsf{v}(\varphi)$ s.t. $\mathbf{s}(\mathtt{x})$ reaches ℓ.

Checking for robustness properties is `ExpTime`-complete for Symbolic Heaps with Inductive Predicates (IP).

Our Goal

Provide a similar result for **propositional** separation logic.

Robustness Properties (Jansen, et al. – ESOP'17)



ic.

ℓ.

Checking for robustness properties is `ExpTime`-complete for Symbolic Heaps with Inductive Predicates (IP).

Our Goal

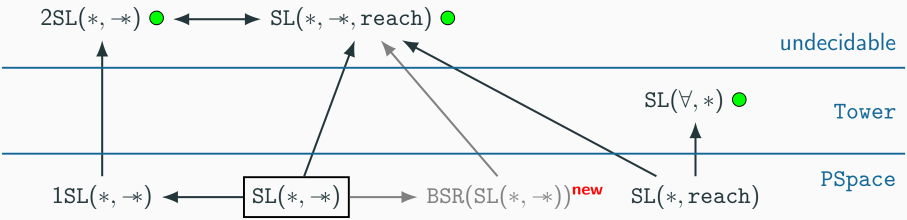
Provide a similar result for **propositional** separation logic.

Desiderata

We aim to an extension of propositional separation logic where

- satisfiability/entailment are decidable in PSpace (as $\mathtt{SL}(*, \mathrel{-\!\!*})$)
- robustness properties reduce to one of these classical problems

Known extensions



Let's start with reachability + 1 quantified variable

- $(\mathbf{s}, \mathbf{h}) \models \mathtt{reach}^+(\mathtt{x}, \mathtt{y}) \iff \mathbf{h}^{\mathbf{L}}(\mathbf{s}(\mathtt{x})) = \mathbf{s}(\mathtt{y})$ for some $\mathbf{L} \geq 1$
- $(\mathbf{s}, \mathbf{h}) \models \exists \mathtt{u}\ \varphi \iff$ there is $\ell \in \mathtt{LOC}$ s.t. $(\mathbf{s}[\mathtt{u} \leftarrow \ell], \mathbf{h}) \models \varphi$

It is only possible to quantify over the variable name $\mathtt{u}$.

Robustness properties reduce to entailment

- **Acyclicity**: $\varphi \models \neg \exists \mathtt{u}\ \mathtt{reach}^+(\mathtt{u}, \mathtt{u})$
- **Garbage freedom**: $\varphi \models \forall \mathtt{u}\ (\mathtt{alloc}(\mathtt{u}) \Rightarrow \bigvee_{\mathtt{x} \in \mathbf{fv}(\varphi)} \mathtt{reach}(\mathtt{x}, \mathtt{u}))$

where $\mathtt{u} \notin \mathbf{fv}(\varphi)$ and

- $\mathtt{alloc}(\mathtt{x}) \stackrel{\text{def}}{=} (\mathtt{x} \hookrightarrow \mathtt{x}) \mathrel{-\!\!*} \bot$
- $\mathtt{reach}(\mathtt{x}, \mathtt{y}) \stackrel{\text{def}}{=} \mathtt{x} = \mathtt{y} \vee \mathtt{reach}^+(\mathtt{x}, \mathtt{y})$

Undecidability and Restrictions

Theorem (Demri, Lozes, M. – 2018, Fossacs)

$\mathtt{SL}(*, -\!\!*)$ *enriched with* $\mathtt{reach}(\mathtt{x}, \mathtt{y}) = 2$ *and* $\mathtt{reach}(\mathtt{x}, \mathtt{y}) = 3$ *is undecidable.*

$\Longrightarrow$ $\mathsf{SAT}(1\mathtt{SL}(*, -\!\!*, \mathtt{reach}^+))$ is undecidable.

We syntactically restrict the logic so that $\mathtt{reach}^+(\mathtt{x}, \mathtt{y})$ is s.t.

R1: it does not appear on the right side of its first $-\!\!*$ ancestor (seeing the formula as a tree)

- $\varphi -\!\!* (\psi * \mathtt{reach}^+(\mathtt{u}, \mathtt{u}))$ violates R1

R2: if $\mathtt{x} = \mathtt{u}$ then $\mathtt{y} = \mathtt{u}$ (syntactically)

- $\mathtt{reach}^+(\mathtt{u}, \mathtt{x})$ violates R2

Note: robustness properties are still expressible (formulae as before)!

Results

1. SAT($\mathtt{1SL}^{\mathtt{R2}}_{\mathtt{R1}}(*, \mathrel{-\!\!*}, \mathtt{reach}^+)$) is $\mathtt{PSpace}$-complete
   - strictly subsumes $\mathtt{1SL}(*, \mathrel{-\!\!*})$ and $\mathtt{SL}(*, \mathtt{reach}^+)$.

2. SAT($\mathtt{1SL}_{\mathtt{R1}}(*, \mathrel{-\!\!*}, \mathtt{reach}^+)$) is $\mathtt{Tower}$-hard.

Proof Techniques

(1) extend the *core formulae technique* used for $\mathtt{SL}(*, \mathrel{-\!\!*})$.

(2) reduction from "an auxiliary logic on trees".

Core formulae technique

(and a bit of $\mathtt{1SL}^{\mathtt{R2}}_{\mathtt{R1}}(*, \mathrel{-\!\!*}, \mathtt{reach}^{+})$)

First order theories: Gaifman Locality Theorem

Theorem (Gaifman – 1982, Herbrand Symposium)

Every FO sentence is logically equivalent to a Boolean combination of ***local formulae****.*

- application of Ehrenfeucht-Fraïssé games

Relation between models		EF-games		Semantics of logic
$\mathfrak{M} \leftrightarrow_n \mathfrak{M}'$	$\rightleftarrows$	Duplicator has a winning strategy	$\rightleftarrows$	$\mathfrak{M} \approx_n \mathfrak{M}'$
(partial iso. up to n)		(n round game)		(n nested quantifiers)

First order theories: Gaifman Locality Theorem

Theorem (Gaifman – 1982, Herbrand Symposium)

Every FO sentence is logically equivalent to a Boolean combination of ***local formulae****.*

- application of Ehrenfeucht-Fraïssé games

Relation between models

$\mathfrak{M} \approx_n \mathfrak{M}'$

eq.sat. local formulae

$\rightleftarrows$

EF-games

Duplicator has a winning strategy

(n round game)

$\rightleftarrows$

Semantics of logic

$\mathfrak{M} \approx_n \mathfrak{M}'$

(n nested quantifiers)

"Locality theorem" for $\mathtt{SL}(*, \mathrel{-\!\!*})$

Theorem (Lozes, 2004 – Space)

Every formula of $\mathtt{SL}(*, \mathrel{-\!\!*})$ *is logically equivalent to a Boolean combination of* ***core formulae****.*

From this theorem we can get:

- expressive power results
- complexity result (small model property)
- axiomatisation

When considering extensions of the logic, we need to derive new core formulae and reprove the theorem.
$\Longrightarrow$ It does not work (at all) for $1\mathtt{SL}^{\mathtt{R2}}_{\mathtt{R1}}(*, \mathrel{-\!\!*}, \mathtt{reach}^+)$.

Core formulae for $\mathtt{SL}(*, -\!*)$

Fix $\mathtt{X} \subseteq \mathtt{VAR}$ and $\alpha \in \mathbb{N}^+$

$$\mathbf{Core}(\mathtt{X}, \alpha) \stackrel{\text{def}}{=} \left\{ \begin{array}{cc} \mathtt{x} = \mathtt{y}, & \mathtt{x} \hookrightarrow \mathtt{y}, \\ \mathtt{alloc}(\mathtt{x}), & \mathtt{size} \geq \beta \end{array} \middle| \begin{array}{l} \beta \in [0, \alpha], \\ \mathtt{x}, \mathtt{y} \in \mathtt{X} \end{array} \right\}$$

where $(\mathsf{s}, \mathbf{h}) \models \mathtt{size} \geq \beta$ iff $\mathrm{card}(\mathrm{dom}(\mathbf{h})) \geq \beta$.

- indistinguishability $\boxed{\text{Relation}}$:

  $$(\mathsf{s}, \mathbf{h}) \leftrightarrow^{\mathtt{X}}_{\alpha} (\mathsf{s}', \mathbf{h}') \ \text{ iff } \ \forall \varphi \in \mathbf{Core}(\mathtt{X}, \alpha),\ (\mathsf{s}, \mathbf{h}) \models \varphi \text{ iff } (\mathsf{s}', \mathbf{h}') \models \varphi$$

- Both EF-game and winning strategy for Duplicator are hidden inside two (technical) elimination lemmas.

Core formulae: $*$ elimination lemma

Lemma

Suppose $(\boldsymbol{s}, \boldsymbol{h}) \leftrightarrow^{\mathtt{X}}_{\alpha} (\boldsymbol{s}', \boldsymbol{h}')$. *Then,*

for every $\alpha_1 + \alpha_2 = \alpha$ $(\alpha_1, \alpha_2 \in \mathbb{N}^+)$, *and every* $\boldsymbol{h}_1 + \boldsymbol{h}_2 = \boldsymbol{h}$, *(Spoiler)*

there are $\boldsymbol{h}'_1 + \boldsymbol{h}'_2 = \boldsymbol{h}'$ *such that* *(Duplicator)*

$$(\boldsymbol{s}, \boldsymbol{h}_1) \leftrightarrow^{\mathtt{X}}_{\alpha_1} (\boldsymbol{s}', \boldsymbol{h}'_1) \text{ and } (\boldsymbol{s}, \boldsymbol{h}_2) \leftrightarrow^{\mathtt{X}}_{\alpha_2} (\boldsymbol{s}', \boldsymbol{h}'_2).$$

- necessary to obtain a winning strategy for Duplicator

Core formulae: $*$ elimination lemma

Lemma

Suppose $(\boldsymbol{s}, \boldsymbol{h}) \leftrightarrow^{\mathtt{X}}_{\alpha} (\boldsymbol{s}', \boldsymbol{h}')$. *Then,*

for every $\alpha_1 + \alpha_2 = \alpha$ $(\alpha_1, \alpha_2 \in \mathbb{N}^+)$, *and every* $\boldsymbol{h}_1 + \boldsymbol{h}_2 = \boldsymbol{h}$, *(Spoiler)*

there are $\boldsymbol{h}'_1 + \boldsymbol{h}'_2 = \boldsymbol{h}'$ *such that* *(Duplicator)*

$$(\boldsymbol{s}, \boldsymbol{h}_1) \leftrightarrow^{\mathtt{X}}_{\alpha_1} (\boldsymbol{s}', \boldsymbol{h}'_1) \text{ and } (\boldsymbol{s}, \boldsymbol{h}_2) \leftrightarrow^{\mathtt{X}}_{\alpha_2} (\boldsymbol{s}', \boldsymbol{h}'_2).$$

- necessary to obtain a winning strategy for Duplicator

By Relation $\leftrightarrows$ EF-games $\leftrightarrows$ Semantics it leads to:

For every $\varphi \in \textbf{Bool}(\textbf{Core}(\mathtt{X}, \alpha_1))$ and $\psi \in \textbf{Bool}(\textbf{Core}(\mathtt{X}, \alpha_2))$
there is $\chi \in \textbf{Bool}(\textbf{Core}(\mathtt{X}, \alpha_1 + \alpha_2))$ such that

$$\varphi * \psi \iff \chi$$

Note: similar elimination lemma for $\mathrel{-\!\!*}$.

Core formulae: after $*$ and $-\!*$ elimination

Theorem

For every φ in $\mathtt{SL}(*, -\!*)$:

1. *there is en equivalent Boolean combination of core formulae.*
2. *for every $\alpha \geq |\varphi|$, $\mathtt{X} \supseteq \mathrm{v}(\varphi)$ and $(\boldsymbol{s}, \boldsymbol{h}) \leftrightarrow_{\alpha}^{\mathtt{X}} (\boldsymbol{s}', \boldsymbol{h}')$,*

$$(\boldsymbol{s}, \boldsymbol{h}) \models \varphi \text{ iff } (\boldsymbol{s}', \boldsymbol{h}') \models \varphi.$$

[2] allows to derive a small-model property which leads to a proof that SAT($\mathtt{SL}(*, -\!*)$) is in $\mathtt{PSpace}$.

$1\mathtt{SL}^{\mathtt{R2}}_{\mathtt{R1}}(*, \mathrel{-\!\!*}, \mathtt{reach}^+)$ is in $\mathtt{PSpace}$: Not so easy...

$$\pi \coloneqq \mathtt{x} = \mathtt{y} \mid \mathtt{x} \hookrightarrow \mathtt{y} \mid \mathtt{emp} \mid \underline{\mathcal{A} \mathrel{-\!\!*} \mathcal{C}\ (\mathtt{R1})}$$

$$\mathcal{C} \coloneqq \pi \mid \mathcal{C} \wedge \mathcal{C} \mid \neg\mathcal{C} \mid \exists \mathtt{u}\, \mathcal{C} \mid \mathcal{C} * \mathcal{C}$$

$$\mathcal{A} \coloneqq \pi \mid \underline{\mathtt{reach}^+(v_1, v_2)} \mid \mathcal{A} \wedge \mathcal{A} \mid \neg\mathcal{A} \mid \exists \mathtt{u}\, \mathcal{A} \mid \mathcal{A} * \mathcal{A}$$

where if $v_1 = \mathtt{u}$ then $v_2 = \mathtt{u}$ ($\mathtt{R2}$).

- Asymmetric $\mathcal{A} \mathrel{-\!\!*} \mathcal{C}$: design two sets of core formulae against
  - two $*$ and two $\exists$ elimination lemmas;
  - one $\mathrel{-\!\!*}$ elimination lemma that glues the two set of core formulae.
- instead of "$\mathtt{size} \geq \beta$ s.t. $\beta \in [1, \alpha]$", the βs of new core formulae are bounded by functions on α, e.g.

  $$\#\mathtt{loop}(\beta) \geq \gamma \qquad \gamma \in [1, \tfrac{1}{2}\alpha(\alpha + 3) - 1]$$

  bounds are found by solving a set of recurrence equations.

Core formulae: Example on a toy logic

$$\varphi := \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 * \varphi_2 \mid \exists \mathtt{u}\ \varphi \mid \mathtt{alloc}(\mathtt{u}) \mid \mathtt{reach}^+(\mathtt{u},\mathtt{u})$$

Some formulae expressible in this logic:

- $\mathtt{size} \geq 0 \stackrel{\text{def}}{=} \top \qquad \mathtt{size} \geq \beta + 1 \stackrel{\text{def}}{=} \exists \mathtt{u}\ (\mathtt{alloc}(\mathtt{u}) * \mathtt{size} \geq \beta)$
- $\mathtt{reach}^+(\mathtt{u},\mathtt{u}){=}\beta$ iff there is a loop of size exactly β involving $\mathbf{s}(\mathtt{u})$.
- $\#\mathtt{loops}(\beta) \geq \gamma \stackrel{\text{def}}{=} \overbrace{\exists \mathtt{u}\, \mathtt{reach}^+(\mathtt{u},\mathtt{u}){=}\beta * \ldots * \exists \mathtt{u}\, \mathtt{reach}^+(\mathtt{u},\mathtt{u}){=}\beta}^{\gamma-1 \text{ times } *}$
- $\mathtt{rem} \geq \beta$ iff there are at least β memory cells not in a loop.

Designing Core Formulae

- Fix $\alpha \in \mathbb{N}^+$
- Let **Core**(α) be the **finite** set of predicates:

$$\left\{ \begin{matrix} \texttt{rem} \geq \beta, \\ \#\texttt{loops}(\beta) \geq \gamma, \\ \#\texttt{loops}_{>\mathcal{R}(\alpha)} \geq \gamma, \end{matrix} \;\middle|\; \begin{matrix} \beta \in [1, \mathcal{R}(\alpha)], \\ \gamma \in [1, \mathcal{L}(\alpha)] \end{matrix} \right\}$$

for some functions $\mathcal{L}$ and $\mathcal{R}$ in $[\mathbb{N} \to \mathbb{N}]$.

$\#\texttt{loops}_{>\beta} \geq \gamma \;=\; \exists \mathtt{u}\, \mathtt{reach}^+(\mathtt{u},\mathtt{u}) \geq \beta + 1 * \ldots * \exists \mathtt{u}\, \mathtt{reach}^+(\mathtt{u},\mathtt{u}) \geq \beta + 1$

Designing Core Formulae

- Fix $\alpha \in \mathbb{N}^+$
- Let **Core**(α) be the **finite** set of predicates:

$$\left\{ \begin{array}{l} \mathtt{rem} \geq \beta, \\ \#\mathtt{loops}(\beta) \geq \gamma, \\ \#\mathtt{loops}_{>\mathcal{R}(\alpha)} \geq \gamma, \end{array} \middle| \begin{array}{l} \beta \in [1, \mathcal{R}(\alpha)], \\ \gamma \in [1, \mathcal{L}(\alpha)] \end{array} \right\}$$

  for some functions $\mathcal{L}$ and $\mathcal{R}$ in $[\mathbb{N} \to \mathbb{N}]$.

These formulae induce a partition on the heap:

- $\mathtt{rem} \geq \beta$ speaks about memory cells not in a loop
- $\#\mathtt{loops}(\beta) \geq \gamma$ speaks about locations in loops of size $\beta \in [1, \mathcal{R}(\alpha)]$
- $\#\mathtt{loops}_{>\mathcal{R}(\alpha)} \geq \gamma$ speaks about locations in loops of size $> \mathcal{R}(\alpha)$.

$\#\mathtt{loops}_{>\beta} \geq \gamma \ = \ \exists \mathtt{u}\, \mathtt{reach}^+(\mathtt{u},\mathtt{u}) \geq \beta + 1 * \ldots * \exists \mathtt{u}\, \mathtt{reach}^+(\mathtt{u},\mathtt{u}) \geq \beta + 1$

Find $\mathcal{R}$ and $\mathcal{L}$

Lemma

Suppose $(\boldsymbol{s}, \boldsymbol{h}) \leftrightarrow_{\alpha}^{\mathtt{X}} (\boldsymbol{s}', \boldsymbol{h}')$. *Then,*

for every $\alpha_1 + \alpha_2 = \alpha$ $(\alpha_1, \alpha_2 \in \mathbb{N}^+)$*, and every* $\boldsymbol{h}_1 + \boldsymbol{h}_2 = \boldsymbol{h}$*,* *(Spoiler)*

...

- Test the core formulae against the $*$ elimination lemma.
- standard-ish way of doing things in EF-games.

Find $\mathcal{R}$ and $\mathcal{L}$

Lemma

Suppose $(\boldsymbol{s}, \boldsymbol{h}) \leftrightarrow_{\alpha}^{\mathtt{X}} (\boldsymbol{s}', \boldsymbol{h}')$. *Then,*

for every $\alpha_1 + \alpha_2 = \alpha$ $(\alpha_1, \alpha_2 \in \mathbb{N}^+)$, *and every* $\boldsymbol{h}_1 + \boldsymbol{h}_2 = \boldsymbol{h}$, *(Spoiler)*

...

- Test the core formulae against the $*$ elimination lemma.
- standard-ish way of doing things in EF-games.

What happens to the locations corresponding to $\mathtt{rem} \geq \beta$,
when we split a heap?

Find $\mathcal{R}$ and $\mathcal{L}$

Lemma

Suppose $(\boldsymbol{s}, \boldsymbol{h}) \leftrightarrow^{\mathtt{X}}_{\alpha} (\boldsymbol{s}', \boldsymbol{h}')$. *Then,*

for every $\alpha_1 + \alpha_2 = \alpha$ $(\alpha_1, \alpha_2 \in \mathbb{N}^+)$*, and every* $\boldsymbol{h}_1 + \boldsymbol{h}_2 = \boldsymbol{h}$*,* *(Spoiler)*

...

- Test the core formulae against the $*$ elimination lemma.
- standard-ish way of doing things in EF-games.

What happens to the locations corresponding to $\mathtt{rem} \geq \beta$,
when we split a heap?

They correspond to $\mathtt{rem} \geq \beta$, also in the subheaps.

Find $\mathcal{R}$ and $\mathcal{L}$

Lemma

Suppose $(\boldsymbol{s}, \boldsymbol{h}) \leftrightarrow_{\alpha}^{\mathtt{X}} (\boldsymbol{s}', \boldsymbol{h}')$. *Then,*

for every $\alpha_1 + \alpha_2 = \alpha$ $(\alpha_1, \alpha_2 \in \mathbb{N}^+)$, *and every* $\boldsymbol{h}_1 + \boldsymbol{h}_2 = \boldsymbol{h}$, *(Spoiler)*

...

- Te
- st

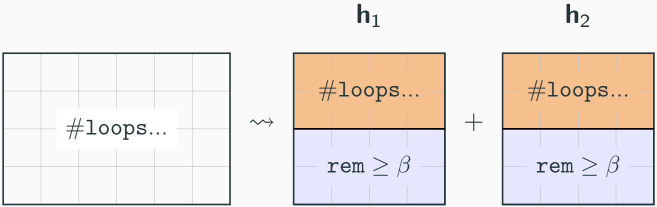
$\mathcal{R}$ $+$

$$\mathcal{R}(\alpha) \geq \max_{\substack{\alpha_1, \alpha_2 \in \mathbb{N}^+ \\ \alpha_1 + \alpha_2 = \alpha}} \left(\mathcal{R}(\alpha_1) + \mathcal{R}(\alpha_2)\right)$$

They correspond to $\texttt{rem} \geq \beta$, also in the subheaps.

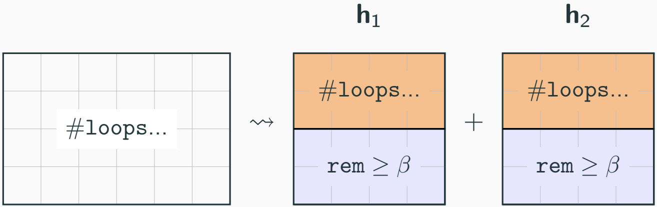
Find $\mathcal{R}$ and $\mathcal{L}$

For $\mathcal{L}$, roughly speaking...



Find $\mathcal{R}$ and $\mathcal{L}$

For $\mathcal{L}$, roughly speaking...



$$\mathcal{L}(\alpha) \geq \max_{\substack{\alpha_1, \alpha_2 \in \mathbb{N}^+ \\ \alpha_1 + \alpha_2 = \alpha}} \left(\mathcal{L}(\alpha_1) + \mathcal{L}(\alpha_2) + \mathcal{R}(\max(\alpha_1, \alpha_2))\right)$$

Find $\mathcal{R}$ and $\mathcal{L}$

We have the inequalities

$$\mathcal{R}(1) \geq 1 \qquad \mathcal{R}(\alpha) \geq \max_{\substack{\alpha_1, \alpha_2 \in \mathbb{N}^+ \\ \alpha_1 + \alpha_2 = \alpha}} (\mathcal{R}(\alpha_1) + \mathcal{R}(\alpha_2))$$

$$\mathcal{L}(1) \geq 1 \qquad \mathcal{L}(\alpha) \geq \max_{\substack{\alpha_1, \alpha_2 \in \mathbb{N}^+ \\ \alpha_1 + \alpha_2 = \alpha}} (\mathcal{L}(\alpha_1) + \mathcal{L}(\alpha_2) + \mathcal{R}(\alpha_1) + \mathcal{R}(\alpha_2))$$

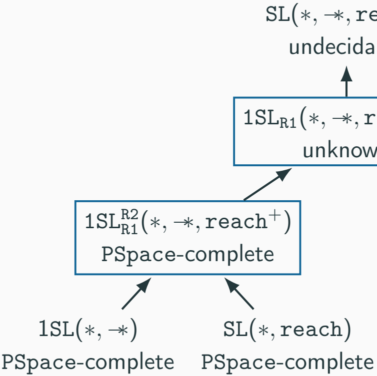
Which admit $\mathcal{R}(\alpha) = \alpha$ and $\mathcal{L}(\alpha) = \frac{1}{2}\alpha(\alpha+1)$ as a solution.

To satisfy the $*$ elimination lemma, build $\leftrightarrow^{\mathtt{X}}_{\alpha}$ w.r.t.

$$\left\{ \begin{cases} \mathtt{rem} \geq \beta, \\ \#\mathtt{loops}(\beta) \geq \gamma, \\ \#\mathtt{loops}_{>\alpha} \geq \gamma, \end{cases} \;\middle|\; \begin{array}{l} \beta \in [1, \alpha], \\ \\ \gamma \in [1, \dfrac{1}{2}\alpha(\alpha+1)] \end{array} \right\}$$

(it is not a solution for the toy logic, we forgot the variable $\mathtt{u}$!)

First recap



- $1\mathsf{SL}^{\mathtt{R2}}_{\mathtt{R1}}(*, -\!\!*, \mathtt{reach}^+)$ strictly generalise other PSpace-complete extensions of propositional separation logic.
- It can be used to check for robustness properties.

ALT: An auxiliary logic on trees

(or, what happens if we allow $\mathtt{reach}^+(\mathtt{u}, \mathtt{x})$)

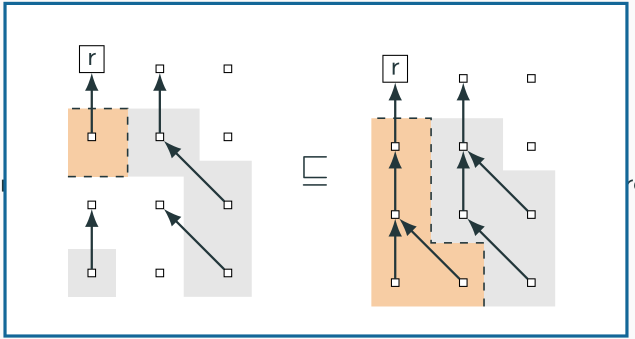
Auxiliary logic on trees (ALT)

$$\varphi \coloneqq \varphi_1 \wedge \varphi_2 \ \mid \ \neg\varphi \ \mid \ \langle \mathrm{U} \rangle \varphi \ \mid \ \blacklozenge\, \varphi \ \mid \ \blacklozenge^{*} \varphi \ \mid \ \triangle \ \mid \ \oslash$$

- interpreted on *acyclic heaps* (finite forests, encoding parent relation)
- one *current node* $\mathsf{n} \in \mathtt{LOC}$, one fixed *target node* $\mathsf{r} \in \mathtt{LOC}$
- $\mathbf{h}, \mathsf{n} \models_{\mathsf{r}} \langle \mathrm{U} \rangle \varphi$ iff there is $\mathsf{n}' \in \mathtt{LOC}$ s.t. $\mathbf{h}, \mathsf{n}' \models_{\mathsf{r}} \varphi$
- $\mathbf{h}, \mathsf{n} \models_{\mathsf{r}} \triangle$ iff $\mathsf{n} \in \mathrm{dom}(\mathbf{h})$ and n reaches r in at least one step
- $\mathbf{h}, \mathsf{n} \models_{\mathsf{r}} \oslash$ iff $\mathsf{n} \in \mathrm{dom}(\mathbf{h})$ and n **does not** reach r in at least one step
- $\blacklozenge\, \varphi \equiv (\mathtt{size} = 1) * \varphi$, $\qquad \blacklozenge^{*} \varphi \equiv \top * \varphi$

We prove that SAT(ALT) is a $\mathtt{Tower}$-complete problem.

Auxiliary logic on trees (ALT)



- inter… …relation)
- one…
- $\mathbf{h}, \mathsf{n}$ …
- $\mathbf{h}, \mathsf{n} \models_{\mathsf{r}} \triangle$ iff $\mathsf{n} \in \mathrm{dom}(\mathbf{h})$ and n reaches r in at least one step
- $\mathbf{h}, \mathsf{n} \models_{\mathsf{r}} \oslash$ iff $\mathsf{n} \in \mathrm{dom}(\mathbf{h})$ and n **does not** reach r in at least one step
- $\blacklozenge \varphi \equiv (\mathtt{size} = 1) * \varphi$, $\qquad \blacklozenge^{*} \varphi \equiv \top * \varphi$

We prove that $\mathrm{SAT}(\mathrm{ALT})$ is a $\mathtt{Tower}$-complete problem.

What can ALT do?

Given a pointed model $(\mathbf{h}, \mathsf{n})$ and a target node r:

If we consider a portion of $\mathbf{h}$ with domain in $\{\mathsf{n}' \in \mathtt{LOC} \mid \mathbf{h}, \mathsf{n}' \models \oslash\}$, ALT **can only express** size bounds.

- Proof done with EF-games for ALT.

$$\begin{aligned} \mathtt{size}(\oslash) \geq 0 \quad &\stackrel{\mathsf{def}}{=} \top \\ \mathtt{size}(\oslash) \geq \beta{+}1 \quad &\stackrel{\mathsf{def}}{=} \langle \mathrm{U} \rangle \big(\oslash \land \blacklozenge(\neg \mathtt{alloc} \land \mathtt{size}(\oslash) \geq \beta)\big) \end{aligned}$$

where $\mathtt{alloc} \stackrel{\mathsf{def}}{=} \oslash \lor \triangle$.

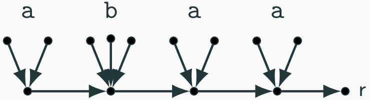
What can ALT do?

- If $\mathbf{h}, \mathsf{n} \models_{\mathsf{r}} \triangle$, ALT can check bounds on the number of descendants and children of n:

$$
\begin{aligned}
\#\texttt{desc} \geq \beta &\stackrel{\textsf{def}}{=} \blacklozenge^{*}\big([\mathrm{U}]\neg\oslash \land \triangle \land \blacklozenge(\neg\texttt{alloc} \land \texttt{size}(\oslash) \geq \beta)\big)\\
\#\texttt{child} \geq 0 &\stackrel{\textsf{def}}{=} \top\\
\#\texttt{child} \geq \beta{+}1 &\stackrel{\textsf{def}}{=} \#\texttt{desc} \geq \beta{+}1 \land \neg\blacklozenge^{\beta}(\triangle \land \neg\#\texttt{desc} \geq 1)
\end{aligned}
$$

- Easy to encode words as acyclic memory states

$\boxed{\texttt{abaa}} \rightsquigarrow$



PITL (Moszkowski'83)

$$\varphi \coloneqq \mathtt{pt} \ \mid\ \mathtt{a} \ \mid\ \varphi_1 \mathbin{\boldsymbol{|}} \varphi_2 \ \mid\ \neg\varphi \ \mid\ \varphi_1 \wedge \varphi_2$$

- interpreted on finite non-empty words over a finite alphabet Σ
- $\mathfrak{w} \models \mathtt{pt} \iff |\mathfrak{w}| = 1$
- $\mathfrak{w} \models \mathtt{a} \iff$ first letter of $\mathfrak{w}$ is $\mathtt{a} \in \Sigma$ (locality principle)
- $\mathfrak{w} \models \varphi_1 \mathbin{\boldsymbol{|}} \varphi_2 \iff \mathfrak{w}[1 : \mathrm{j}] \models \varphi_1$ and $\mathfrak{w}[\mathrm{j} : |\mathfrak{w}|] \models \varphi_2$ for some $\mathrm{j} \in [1, |\mathfrak{w}|]$

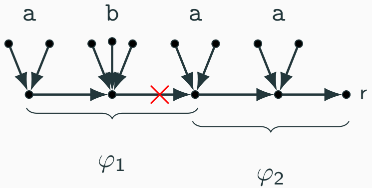
$$\boxed{\mathfrak{w}_1 \ldots \mathfrak{w}_{\mathrm{j}-1} \quad \mathfrak{w}_{\mathrm{j}} \quad \mathfrak{w}_{\mathrm{j}+1} \ldots \mathfrak{w}_{|\mathfrak{w}|}}$$

φ_1 φ_2

Note: SAT(PITL) is $\mathtt{Tower}$-complete.

Reducing PITL to ALT

- Set of models encoding words can be characterised in ALT
- However, difficult to translate $\varphi_1 \,|\, \varphi_2$!



After the cut, left side does not reach r anymore.

$\Longrightarrow$ nodes on the left side satisfy $\oslash$

$\Longrightarrow$ We cannot express the satisfaction of φ_1.

PITL to ALT: alternative semantics for PITL

- $\boxed{\mathsf{a}}$ marked representation of $\mathsf{a} \in \Sigma$

$$\boxed{\mathfrak{w}_1 \ldots \mathfrak{w}_{j-1}\ \mathfrak{w}_j\ \mathfrak{w}_{j+1} \ldots \boxed{\mathfrak{w}_{|\mathfrak{w}|}}}$$

- $\varphi \boldsymbol{|} \psi$ on standard semantics:

$$\underbrace{\boxed{\mathfrak{w}_1 \ldots \mathfrak{w}_{j-1}\ \mathfrak{w}_j}}_{\varphi_1} \qquad \underbrace{\boxed{\mathfrak{w}_j\ \mathfrak{w}_{j+1} \ldots \mathfrak{w}_{|\mathfrak{w}|}}}_{\varphi_2}$$

- $\varphi \boldsymbol{|} \psi$ on marked semantics

$$\boxed{\underbrace{\mathfrak{w}_1 \ldots \mathfrak{w}_{j-1} \boxed{\mathfrak{w}_j}}_{\varphi_1}\ \mathfrak{w}_{j+1} \ldots \boxed{\mathfrak{w}_{|\mathfrak{w}|}}} \qquad \underbrace{\boxed{\mathfrak{w}_j \mathfrak{w}_{j+1} \ldots \boxed{\mathfrak{w}_{|\mathfrak{w}|}}}}_{\varphi_2}$$

- alternative semantics is equivalent to the original one.

ALT, marking an element

- Given an alphabet $\Sigma = \{\mathtt{a}_1, \ldots, \mathtt{a}_n\}$, $\mathtt{a}_i$ and $\boxed{\mathtt{a}_i}$ are encoded as

$\implies$ marking a character $\sim$ removing a single child.

- SAT(PITL) can be reduced to SAT(ALT),
(translated formula is in $\mathtt{2ExpSpace}$ if Σ is coded in binary)

$\implies$ ALT is $\mathtt{Tower}$-complete (upper-bound from $\mathtt{MSO}$).

Some logics that are $\texttt{Tower}$-hard

- It is easy to see that ALT is a fragment of $\texttt{1SL}_{\texttt{R1}}(*, \mathrel{-\!\!*}, \texttt{reach}^+)$:

  fix $\texttt{x} \in \texttt{VAR}$ to play the role of the target node r,

  $$\langle \mathrm{U} \rangle \varphi \equiv \exists \texttt{u}\ \varphi \qquad \triangle \equiv \texttt{reach}^+(\texttt{u}, \texttt{x}) \qquad \oslash \equiv \texttt{alloc}(\texttt{u}) \wedge \neg \triangle$$

  + impose acyclic heaps: $\neg \exists \texttt{u}\ \texttt{reach}^+(\texttt{u}, \texttt{u})$.

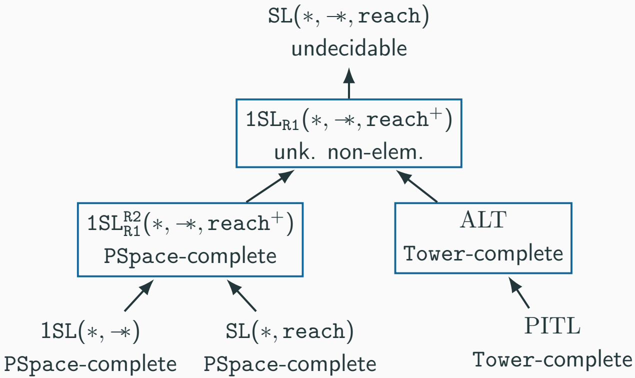
- ALT is a fragment of $\texttt{MSL}(*, \Diamond, \langle \mathrm{U} \rangle)$

- $\mathrm{ALT} \preceq_{\mathsf{SAT}} \texttt{MLH}(*, \Diamond, \langle \mathrm{U} \rangle)$ with modal depth 2.
  (then $*$, $\exists \texttt{u}$, $\texttt{alloc}(\texttt{u})$, $\texttt{alloc}^2(\texttt{u})$ is $\texttt{Tower}$-c.)

- $\mathrm{ALT} \preceq_{\mathsf{SAT}} \texttt{QCTL}(\texttt{U})$ without imbricated until operators $\texttt{U}$
  (or $\texttt{QCTL}(\texttt{EF})$ with 2 imbrication of $\texttt{EF}$)

Note: in these results $*$ can always be replaced with $\blacklozenge$ and $\blacklozenge^*$.

Second Recap



- ALT improves the understanding of some Tower-complete logics.
- It seems to be an interesting tool to prove Tower-hardness.