On Temporal and Separation Logics

Stéphane Demri

CNRS, LSV, ENS Paris-Saclay

TIME'18

Warsaw, October 2018

The blossom of separation logics

- Separation logic: extension of Hoare-Floyd logic for (concurrent) programs with mutable data structures.

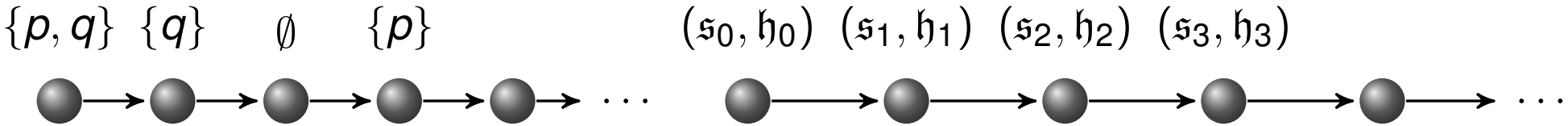
- A family of logical formalisms:
  - symbolic heap fragment,
  - negation-closed standard propositional SL($*$, $-\!\!*$),
  - first-order separation logics,
  - user-defined inductive predicates,
  - reasoning about data values, etc.

- Provers handling SL, translations into SMT solvers, separation logics verified in Coq,

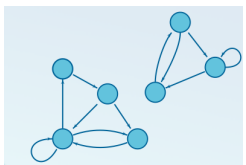
- Prestigious awards.
  - CAV award 2016 (Berdine, Calcagno, Distefano, Ishtiaq, O'Hearn, Reynolds, Yang)
  - Gödel prize 2016 for concurrent separation logic (O'Hearn, Brookes)

Relating temporal logics with separation logics

- Tree-like models vs. heaps as finite “forests”.
- LTL models vs. sequences of memory states



- Model-checking vs. deductive verification.



$\models \quad \phi$?

$$
\begin{array}{c}
\{\mathtt{emp}\} \\
\mathtt{x = new()} \\
\{x \mapsto -\} \\
\mathtt{y = new()} \\
\{(x \mapsto -) * (y \mapsto -)\} \\
\{x \mapsto -\} \quad \| \quad \{y \mapsto -\} \\
\mathtt{free(x)} \quad \| \quad \mathtt{free(y)} \\
\{\mathtt{emp}\} \quad \| \quad \{\mathtt{emp}\} \\
\{\mathtt{emp} * \mathtt{emp}\} \\
\{\mathtt{emp}\}
\end{array}
$$

Overview

1. Separation logic(s) in a nutshell
2. Relationships with temporal logics
3. Encoding linear structures
4. Modalities with separating connectives
5. Conclusion

Separation logic(s) in a nutshell

Floyd-Hoare logic

- Hoare triple: $\{\phi\}\ \mathtt{C}\ \{\psi\}$ (partial correctness).

  [Hoare, C. ACM 69; Floyd, 1967]

  - Precondition ϕ. — Assertion language
  - Postcondition ψ. — Assertion language
  - Command/program $\mathtt{C}$. — Programming language

- If we start in a state where ϕ holds true and the command $\mathtt{C}$ terminates, then it yields a state in which ψ holds.

- Proof system with axioms and deduction rules to derive new triples.

- Strengthening preconditions / weakening postconditions:

$$\frac{\phi \Rightarrow \phi' \quad \{\phi'\}\ \mathtt{C}\ \{\psi\} \quad \psi \Rightarrow \psi'}{\{\phi\}\ \mathtt{C}\ \{\psi'\}}$$

- Hoare's assignment axiom:

$$\frac{}{\{\phi[\mathtt{e}/\mathtt{x}]\}\ \mathtt{x} := \mathtt{e}\ \{\phi\}}$$

The rule of constancy

$$\frac{\{\phi\}\ \mathtt{C}\ \{\psi\}}{\{\phi \wedge \psi'\}\ \mathtt{C}\ \{\psi \wedge \psi'\}}$$

where $\mathtt{C}$ does not mess with ψ'

$$\frac{\{\mathtt{x} = 3\}\ \mathtt{x} := 4; \mathtt{z} := \mathtt{x}\ \{\mathtt{x} = 4\}}{\{\mathtt{x} = 3 \wedge \mathtt{y} = 8\}\ \mathtt{x} := 4; \mathtt{z} := \mathtt{x}\ \{\mathtt{x} = 4 \wedge \mathtt{y} = 8\}}$$

When separation logic enters into the play

$\mathtt{x} := \mathbf{cons}(\mathtt{e})/\mathbf{new}(\mathtt{e})$	allocation
$\mathtt{x} := [\mathtt{e}]$	lookup
$[\mathtt{e}] := \mathtt{e}'$	mutation
$\mathbf{dispose}(\mathtt{e})/\mathbf{free}(\mathtt{e})$	deallocation

Heap $\mathfrak{h}$: finite set of pairs made of a location and a value in $\mathtt{Val}$

$$(\mathfrak{s}, \mathfrak{h} \uplus \{[\![\mathtt{e}]\!] \mapsto n\}), [\mathtt{e}] := \mathtt{e}' \rightsquigarrow (\mathfrak{s}, \mathfrak{h} \uplus \{[\![\mathtt{e}]\!] \mapsto [\![\mathtt{e}']\!]\}), \mathtt{skip}$$

- Rule of constancy:

$$\frac{\{\phi\}\ \mathtt{C}\ \{\psi\}}{\{\phi \wedge \psi'\}\ \mathtt{C}\ \{\psi \wedge \psi'\}}$$

  where $\mathtt{C}$ does not mess with ψ'.

- Unsoundness of the rule of constancy with pointers:

$$\frac{\{\phi_1\}\ [\mathtt{x}] := 4\ \{\phi_2\}}{\{\phi_1 \wedge [\mathtt{y}] = 3\}\ [\mathtt{x}] := 4\ \{\phi_2 \wedge [\mathtt{y}] = 3\}} \quad \text{if } \mathtt{x} = \mathtt{y} \text{ then } [\mathtt{x}] = [\mathtt{y}]$$

Frame rule and separating conjunction

- Frame rule:

$$\frac{\{\phi\}\ \mathtt{C}\ \{\psi\}}{\{\phi * \psi'\}\ \mathtt{C}\ \{\psi * \psi'\}}$$

  where $\mathtt{C}$ does not mess with ψ'.

$$\frac{\{[\mathtt{x}] = 5\}\ [\mathtt{x}] := 4\ \{[\mathtt{x}] = 4\}}{\{[\mathtt{x}] = 5 * [\mathtt{y}] = 3\}\ [\mathtt{x}] := 4\ \{[\mathtt{x}] = 4 * [\mathtt{y}] = 3\}}$$

- $(\mathfrak{s}, \mathfrak{h}) \models [\mathtt{x}] = 5 * [\mathtt{y}] = 3$ implies $\mathtt{x} \neq \mathtt{y}$.
- $[\mathtt{z}] = \mathtt{z}'$ written $\mathtt{z} \hookrightarrow \mathtt{z}'$ in separation logic.

A taste of separation logic

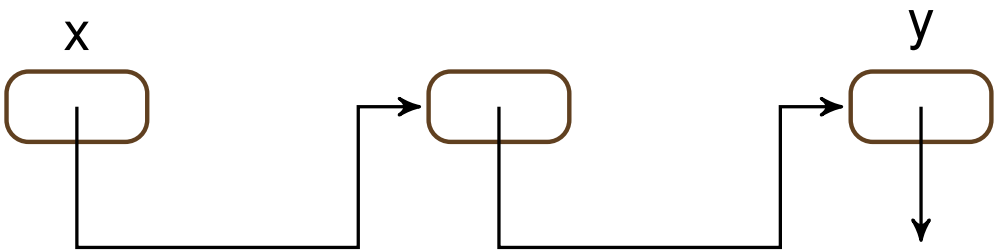
assertion logic + programming language + deduction rules

- Introduced by Ishtiaq, O'Hearn, Pym, Reynolds, Yang.
  circa 1998-2000, see also [Burstall, MI 72]
- Extension of Hoare logic with separating connectives.
  [O'Hearn, Reynolds & Yang, CSL'01; Reynolds, LICS'02]
- Separating conjunction $*$ and its adjunct $-\!\!*$.
- Automatic program analysis.
  Tools: Infer, Slayer, Space Invader, Smallfoot, etc.
- Separation logic competitions SL-COMP'14 & '18.

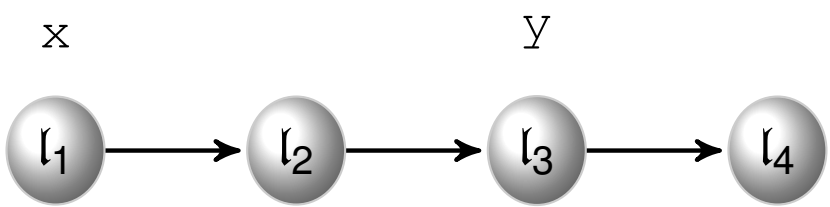
Memory states with one record field

- Program variables PVAR $= \{\mathtt{x}_1, \mathtt{x}_2, \mathtt{x}_3, \ldots\}$.
- $\mathtt{Loc}$: countably infinite set of locations
  $\mathtt{Val}$: countably infinite set of values with $\mathtt{Loc} \subseteq \mathtt{Val}$.
- Memory state $(\mathfrak{s}, \mathfrak{h})$:
  - Store $\mathfrak{s}$: PVAR $\to \mathtt{Val}$.
  - Heap $\mathfrak{h} : \mathtt{Loc} \rightharpoonup_{\mathit{fin}} \mathtt{Val}$ (finite domain).
    (richer models, e.g. with $\mathfrak{h} : \mathtt{Loc} \rightharpoonup_{\mathit{fin}} \mathtt{Val}^k$)
  - In this talk, we assume $\mathtt{Loc} = \mathtt{Val} = \mathbb{N}$.

Graphical representation

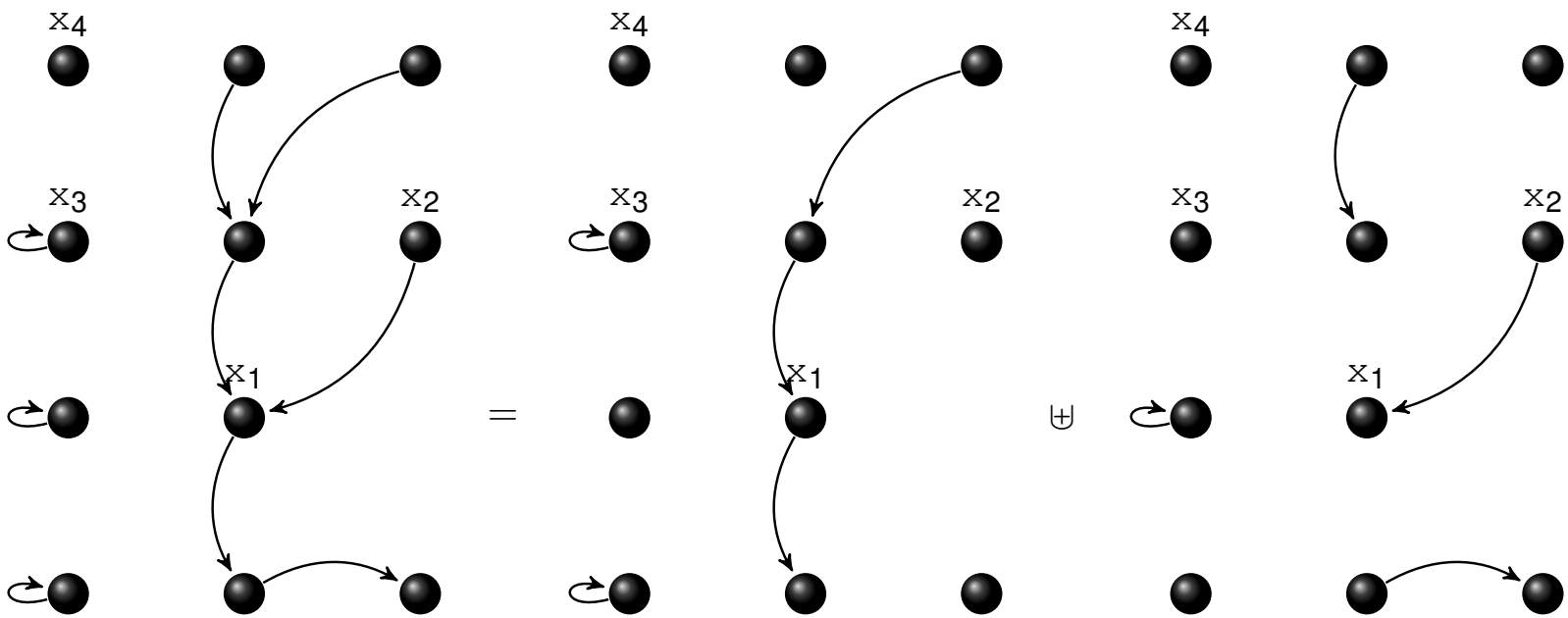


$$\mathfrak{s}(\mathtt{x}) = \mathfrak{l}_1$$
$$\mathfrak{s}(\mathtt{y}) = \mathfrak{l}_3$$
$$\mathrm{dom}(\mathfrak{h}) = \{\mathfrak{l}_1, \mathfrak{l}_2, \mathfrak{l}_3\}$$
$$\mathfrak{h}(\mathfrak{l}_1) = \mathfrak{l}_2$$
$$\mathfrak{h}(\mathfrak{l}_2) = \mathfrak{l}_3$$
$$\mathfrak{h}(\mathfrak{l}_3) = \mathfrak{l}_4$$



Disjoint heaps

- Disjoint heaps: $\mathrm{dom}(\mathfrak{h}_1) \cap \mathrm{dom}(\mathfrak{h}_2) = \emptyset$ (noted $\mathfrak{h}_1 \perp \mathfrak{h}_2$).
- When $\mathfrak{h}_1 \perp \mathfrak{h}_2$, disjoint heap $\mathfrak{h}_1 \uplus \mathfrak{h}_2$.



Syntax and semantics for 1SL

- Quantified variables $\mathtt{FVAR} = \{\mathtt{u}_1, \mathtt{u}_2, \mathtt{u}_3, \ldots\}$.
- Expressions and atomic formulae:

$$\mathtt{e} ::= \mathtt{x}_i \mid \mathtt{u}_j \qquad \pi ::= \mathtt{e} = \mathtt{e}' \mid \mathtt{e} \hookrightarrow \mathtt{e}' \mid \mathtt{emp}$$

- Formulae:

$$\phi ::= \pi \mid \phi \wedge \psi \mid \neg\phi \mid \phi * \psi \mid \phi \mathrel{-\!\!*} \psi \mid \exists\, \mathtt{u}\ \phi$$

- Models: memory states $(\mathfrak{s}, \mathfrak{h})$ + $\mathfrak{f} : \mathtt{FVAR} \to \mathtt{Val}$.
- $(\mathfrak{s}, \mathfrak{h}) \models_{\mathfrak{f}} \mathtt{emp} \overset{\text{def}}{\Leftrightarrow} \mathrm{dom}(\mathfrak{h}) = \emptyset$.
- $(\mathfrak{s}, \mathfrak{h}) \models_{\mathfrak{f}} \mathtt{e} = \mathtt{e}' \overset{\text{def}}{\Leftrightarrow} [\![\mathtt{e}]\!] = [\![\mathtt{e}']\!]$, with $[\![\mathtt{x}]\!] \overset{\text{def}}{=} \mathfrak{s}(\mathtt{x})$, $[\![\mathtt{u}]\!] \overset{\text{def}}{=} \mathfrak{f}(\mathtt{u})$.
- $(\mathfrak{s}, \mathfrak{h}) \models_{\mathfrak{f}} \mathtt{e} \hookrightarrow \mathtt{e}' \overset{\text{def}}{\Leftrightarrow} [\![\mathtt{e}]\!] \in \mathrm{dom}(\mathfrak{h})$ and $\mathfrak{h}([\![\mathtt{e}]\!]) = [\![\mathtt{e}']\!]$.

Binary modality: separating conjunction

$$(\mathfrak{s}, \mathfrak{h}) \models_{\mathfrak{f}} \phi_1 * \phi_2$$

$$\overset{\text{def}}{\Leftrightarrow}$$

for some $\mathfrak{h}_1, \mathfrak{h}_2$ such that $\mathfrak{h} = \mathfrak{h}_1 \uplus \mathfrak{h}_2$,

$(\mathfrak{s}, \mathfrak{h}_1) \models_{\mathfrak{f}} \phi_1$ and $(\mathfrak{s}, \mathfrak{h}_2) \models_{\mathfrak{f}} \phi_2$

$-\!\!*$ universally quantifies over an infinite set !

$$(\mathfrak{s}, \mathfrak{h}) \models_{\mathfrak{f}} \phi_1 -\!\!* \phi_2$$

$$\overset{\text{def}}{\Leftrightarrow}$$

for all $\mathfrak{h}'$,

if $\mathfrak{h} \perp \mathfrak{h}'$ and $(\mathfrak{s}, \mathfrak{h}') \models_{\mathfrak{f}} \phi_1$,

then $(\mathfrak{s}, \mathfrak{h} \uplus \mathfrak{h}') \models_{\mathfrak{f}} \phi_2$

- $*$ and $-\!\!*$ are adjunct operators:

$$\varphi_1 * \varphi_2 \models \varphi_3 \quad \text{iff} \quad \varphi_1 \models \varphi_2 -\!\!* \varphi_3$$

Simple properties stated in 1SL

- The value of $\mathtt{x}$ is in the domain of the heap:
  $\mathtt{alloc(x)} \stackrel{\text{def}}{=} \exists \mathtt{u}\ \mathtt{x} \hookrightarrow \mathtt{u}$ (variant of $(\mathtt{x} \hookrightarrow \mathtt{x}) \mathrel{-\!\!*} \bot$)

- The heap has a unique cell $\mathtt{x} \mapsto \mathtt{x}'$:

$$\mathtt{x} \mapsto \mathtt{x}' \stackrel{\text{def}}{=} \mathtt{x} \hookrightarrow \mathtt{x}' \wedge \neg \exists \mathtt{u}' \, (\mathtt{u}' \neq \mathtt{x} \wedge \mathtt{alloc}(\mathtt{u}'))$$

- The domain of the heap is empty: $\mathtt{emp} \stackrel{\text{def}}{=} \neg \exists\, \mathtt{u}\ \mathtt{alloc}(\mathtt{u})$

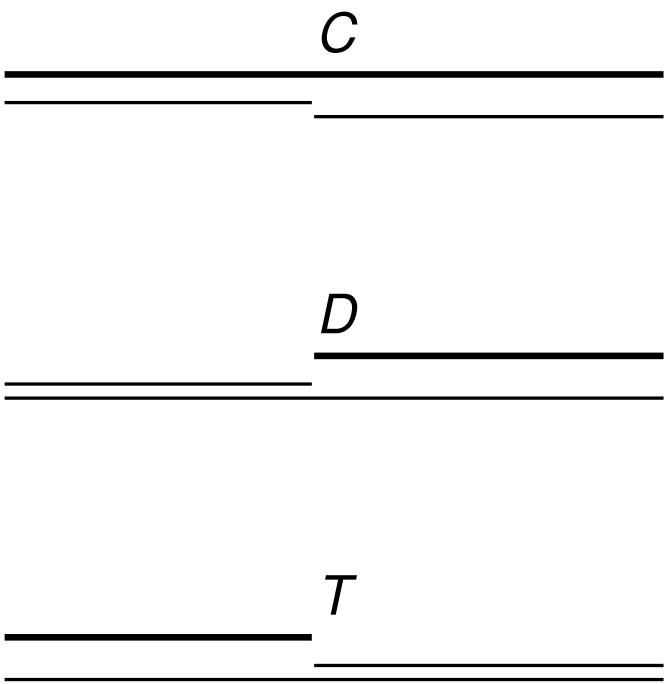
- $\mathtt{x}$ has at least *n* predecessors:

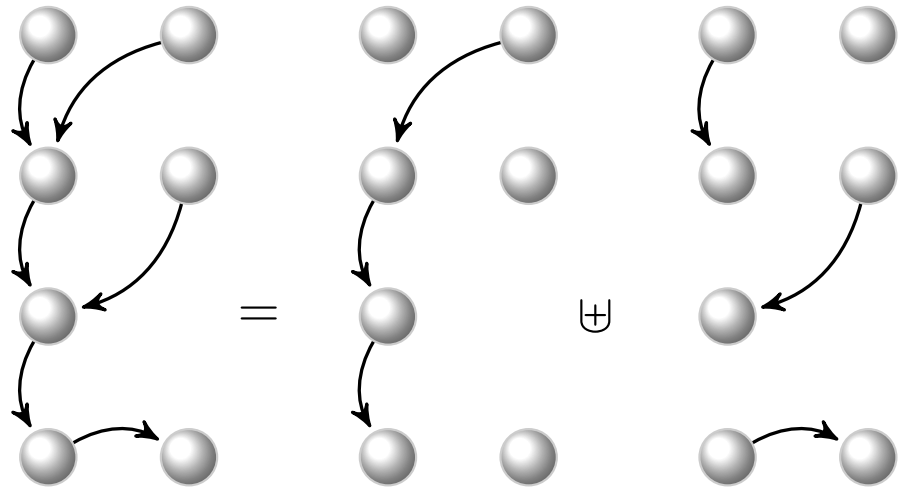
$$\overbrace{(\exists\, \mathtt{u}\ (\mathtt{u} \hookrightarrow \mathtt{x})) * \cdots * (\exists\, \mathtt{u}\ (\mathtt{u} \hookrightarrow \mathtt{x}))}^{n \text{ times}}$$

Relationships with temporal logics

Relating operators C, D, and T with $*$ and $-\!\circledast$

- Interval temporal logics with C, D, and T on linear orders.
  See e.g. [Venema, JLC 1991; Hodkinson et al., CSL'08]



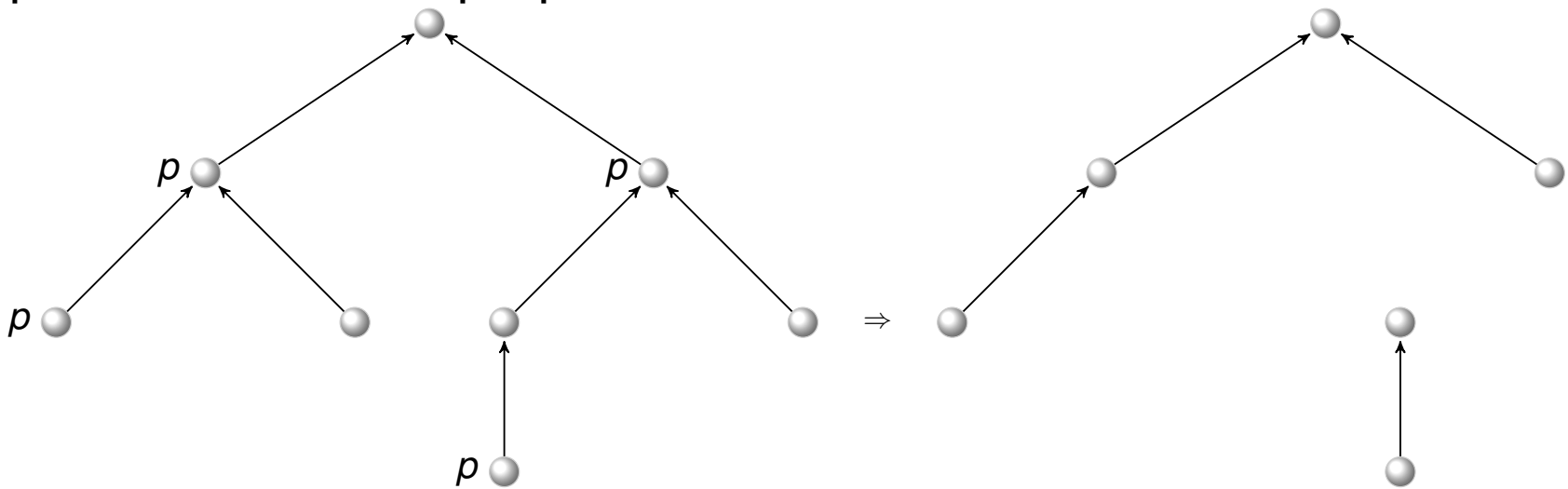


Septraction: $\phi -\!\circledast \psi \stackrel{\text{def}}{=} \neg(\phi \mathrel{-\!\!*} \neg\psi)$

+ π and propositional variables atomic formulae: $\mathtt{emp}$, $\mathtt{x} = \mathtt{y}$, $\mathtt{x} \hookrightarrow \mathtt{y}$.

Separating conjunction and prop. quantification

- The separating connectives $*$ and $-\!\!*$ correspond to second-order quantifications.
- Separating conjunction is strongly related to second-order quantification over propositions.



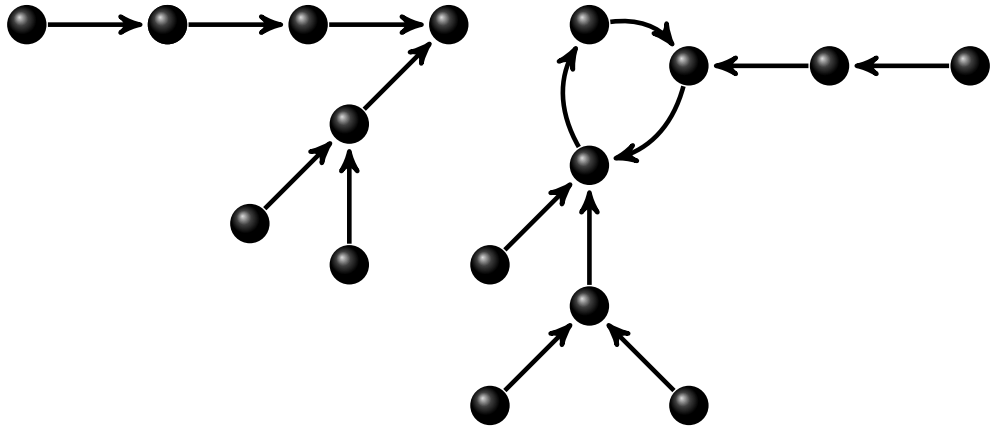
- Quantified CTL (QCTL) with tree semantics is decidable with non-elementary satisfiability problem.

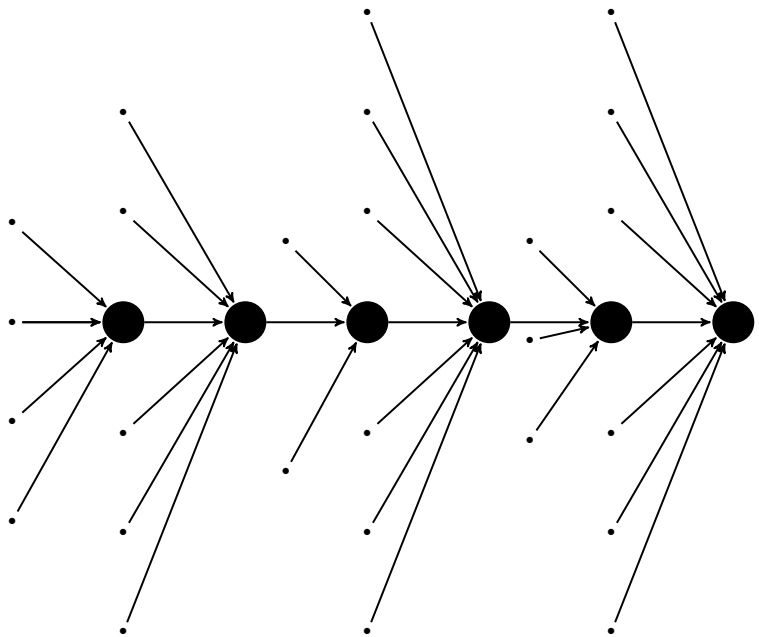
[Laroussinie & Markey, LMCS 2014]

- Restriction to QCTL(EX) is still TOWER-hard.

(work in progress with B. Bednarczyk)

Encoding linear/tree-like structures





Encoding linear structures

Reachability predicate in 1SL2(∗)

- $\mathtt{u}$ has a successor: $\mathtt{alloc}(\mathtt{u}) \stackrel{\text{def}}{=} \exists\, \overline{\mathtt{u}}\ \mathtt{u} \hookrightarrow \overline{\mathtt{u}}$
- $\mathtt{u}$ has at least k predecessors:

$$\sharp \mathtt{u} \geq k \stackrel{\text{def}}{=} \overbrace{(\exists\, \overline{\mathtt{u}}\ (\overline{\mathtt{u}} \hookrightarrow \mathtt{u})) * \cdots * (\exists\, \overline{\mathtt{u}}\ (\overline{\mathtt{u}} \hookrightarrow \mathtt{u}))}^{k \text{ times}}$$

- Non-empty path from $\mathtt{u}$ to $\overline{\mathtt{u}}$ and nothing else except loops:

$$\begin{aligned}\mathtt{reach}'(\mathtt{u}, \overline{\mathtt{u}}) \stackrel{\text{def}}{=}\ & \sharp \mathtt{u} = 0 \wedge \mathtt{alloc}(\mathtt{u}) \wedge \neg \mathtt{alloc}(\overline{\mathtt{u}}) \wedge \\ & \forall\, \overline{\mathtt{u}}\ ((\mathtt{alloc}(\overline{\mathtt{u}}) \wedge \sharp \overline{\mathtt{u}} = 0) \Rightarrow \overline{\mathtt{u}} = \mathtt{u}) \wedge \\ & \forall\, \mathtt{u}\ ((\sharp \mathtt{u} \neq 0 \wedge \mathtt{u} \neq \overline{\mathtt{u}}) \Rightarrow (\sharp \mathtt{u} = 1 \wedge \mathtt{alloc}(\mathtt{u})))\end{aligned}$$

- There is a path from $\mathtt{u}$ to $\overline{\mathtt{u}}$:

$$\mathtt{reach}(\mathtt{u}, \overline{\mathtt{u}}) \stackrel{\text{def}}{=} \mathtt{u} = \overline{\mathtt{u}} \vee (\top * \mathtt{reach}'(\mathtt{u}, \overline{\mathtt{u}}))$$

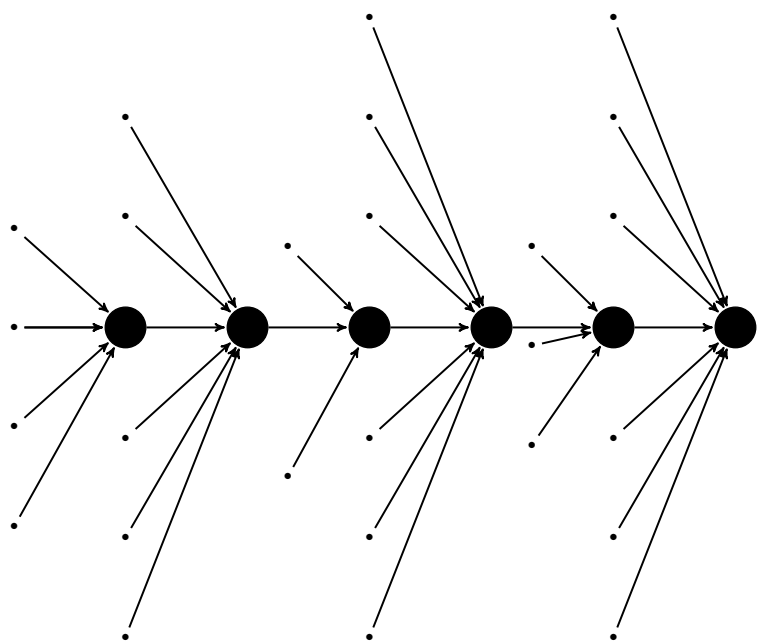
Fishbone heaps

- $\mathfrak{h}$ is a fishbone heap

  (fb1) $\mathrm{dom}(\mathfrak{h}) \neq \emptyset$.

  (fb2) There is a location reachable from all the locations of $\mathrm{dom}(\mathfrak{h})$ that is not in $\mathrm{dom}(\mathfrak{h})$.

  (fb3) there are no distinct locations $\mathfrak{l}_1, \mathfrak{l}_2, \mathfrak{l}_3, \mathfrak{l}_4, \mathfrak{l}_5$ such that $\mathfrak{l}_1 \to \mathfrak{l}_2 \to \mathfrak{l}_3 \leftarrow \mathfrak{l}_4 \leftarrow \mathfrak{l}_5$ in the heap $\mathfrak{h}$.

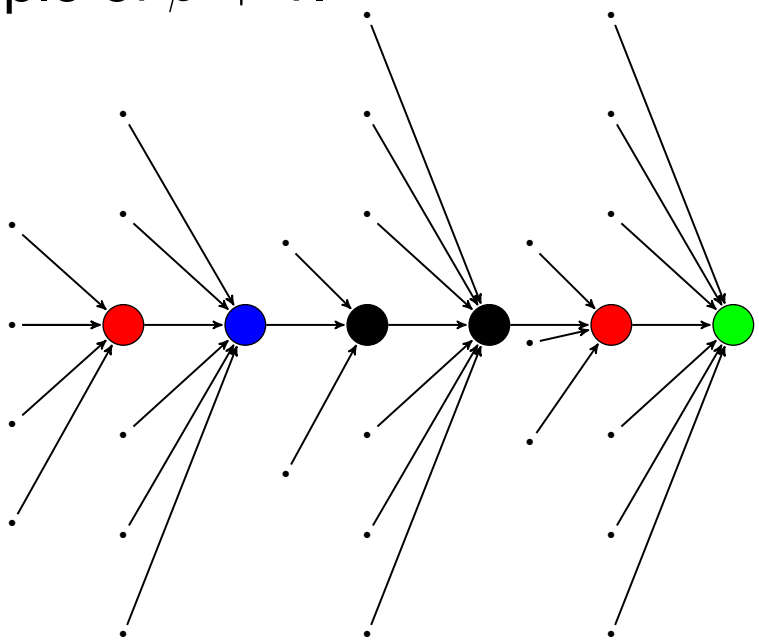


(α, β)-fishbone heap

(C1) the first location on the main path has a number of predecessors in $[1 + 2, \alpha + 2]$.

(C2) on the main path, a location with a number of predecessors in $[3, \alpha + 2]$, is followed by β locations with at least $\alpha + 3$ predecessors, and

(C3) the number of locations on the main path is a multiple of $\beta + 1$.



Encoding data words by fishbone heaps

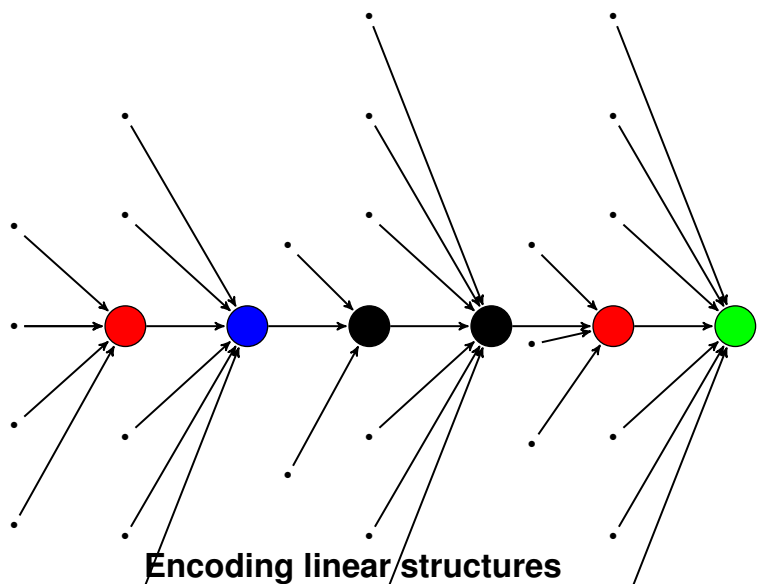
- Data word: $\begin{matrix} a_1 & a_2 & a_3 & \cdots \\ d_1 & d_2 & d_3 & \cdots \end{matrix}$
  - Each a_i belongs to a finite alphabet Σ.
  - Each d_i belongs to an infinite domain D.

- Data word

$$\mathfrak{dw} = (a^1, \mathfrak{d}^1_1, \ldots, \mathfrak{d}^1_\beta) \cdots (a^L, \mathfrak{d}^L_1, \ldots, \mathfrak{d}^L_\beta) \in ([1, \alpha] \times \mathbb{N}^\beta)^+$$

$$(2,1)(1,2)(2,2) \in ([1,2] \times \mathbb{N})^+$$

$$\approx$$



PITL and the fragment 1SL2($*$)

- Propositional Interval Temporal Logic (PITL).

  [Moszkowski, PhD 83]

- Models: non-empty finite words over alphabet $\Sigma = [1, \alpha]$. (data words with $\beta = 0$)

- $\phi \quad ::= \quad a \mid \mathtt{pt} \mid \neg\phi \mid \phi \wedge \phi \mid \phi \,\mathbf{C}\, \phi$

- Chop (with overlap):

$$\mathtt{chops} \stackrel{\text{def}}{=} \{(\mathfrak{w}_1, \mathfrak{w}_2, \mathfrak{w}_3) \in (\Sigma^+)^3 \mid \exists\, a, \mathfrak{w}', \mathfrak{w}'' \text{ such that}$$

$$\mathfrak{w}_1 = \mathfrak{w}'\, a\, \mathfrak{w}'', \mathfrak{w}_2 = \mathfrak{w}'\, a, \mathfrak{w}_3 = a\, \mathfrak{w}''\}$$

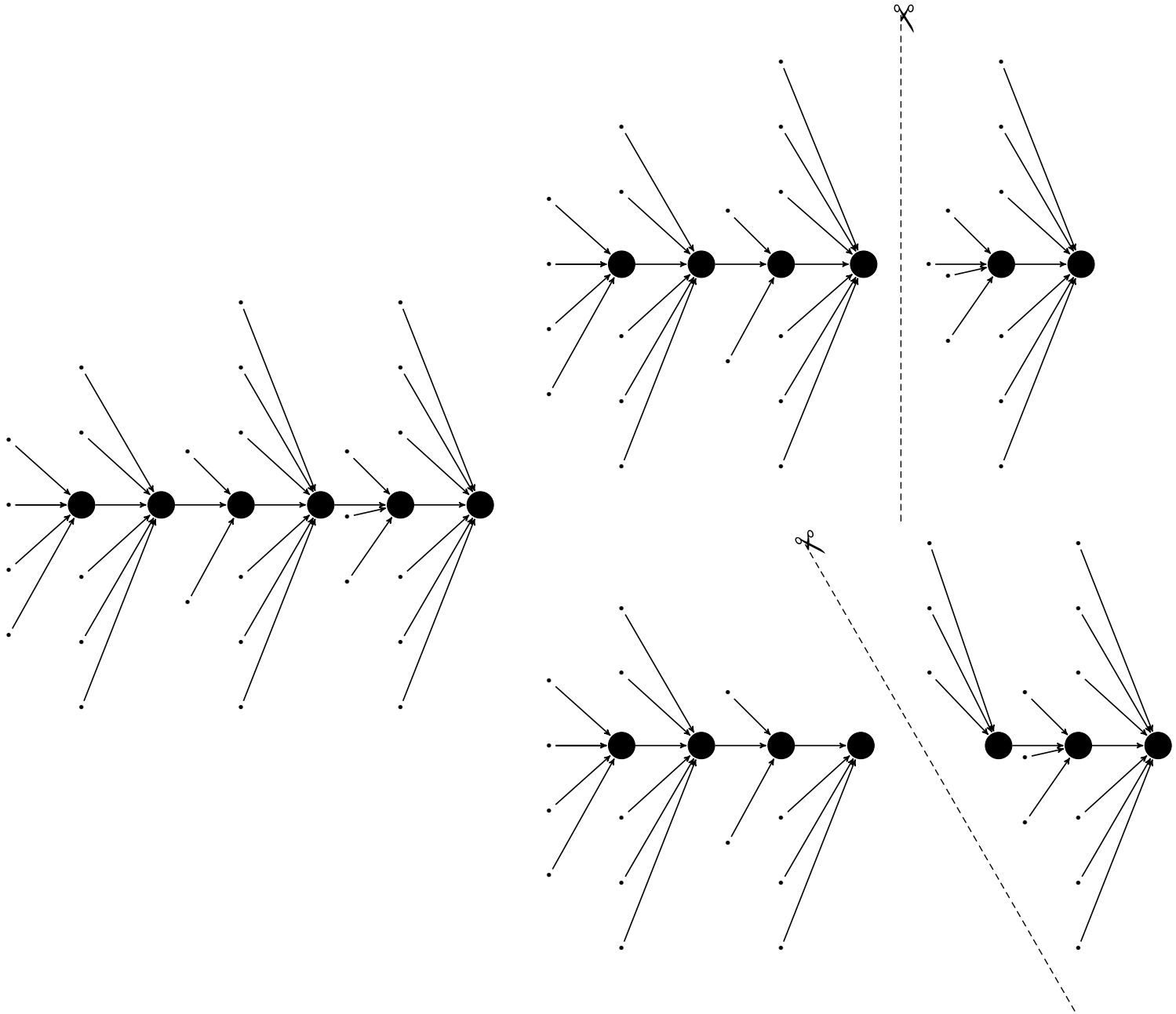
Semantics for PITL

- $\mathfrak{w} \models a \stackrel{\text{def}}{\Leftrightarrow} \mathfrak{w}[1] = a$ (locality cond.) ; $\mathfrak{w} \models \mathtt{pt} \stackrel{\text{def}}{\Leftrightarrow} |\mathfrak{w}| = 1$.
- $\mathfrak{w} \models \phi \mathbf{C} \psi \stackrel{\text{def}}{\Leftrightarrow}$ there are $\mathfrak{w}_1, \mathfrak{w}_2$ s.t. $(\mathfrak{w}, \mathfrak{w}_1, \mathfrak{w}_2) \in \mathtt{chops}$, $\mathfrak{w}_1 \models \phi$ and $\mathfrak{w}_2 \models \psi$.
- Given $\alpha \geq 1$ and $\Sigma = [1, \alpha]$, the problem SAT(PITL$_\Sigma$) is decidable, but with $\alpha \geq 2$ is not elementary recursive.

  [Halpern, Kozen, Moszkowski, 80's]
- Chop in PITL can be encoded with separating conjunction in 1SL2($*$).
- The satisfiability problem for 1SL2($*$) is TOWER-complete.

  [Demri & Deters, ToCL 15; Schmitz, ToCT 16]

Clean cuts are needed!



Pointer to a pointer about separation logic

Concurrent Separation Logic – Temporal Separation

Tony Hoare, Microsoft Research, Cambridge
Peter O'Hearn, Queen Mary, University of London

Separation Logic already has a notion of **spatial separation**.

Hoare and O'Hearn consider **separation in time** in the article

Separation Logic Semantics for Communicating Processes.

They use state sequences and sequential operator similar to ITL's.

Quote from Zhou Chaochen in recent book:

> Moreover, in a personal communication, Tony Hoare drew my attention to the possible correspondence between the chop operator in **Interval Logic** and the separating (spatial) conjunction in **Separation Logic**. I believe that **Interval Logic** deserves an important role in computing science.

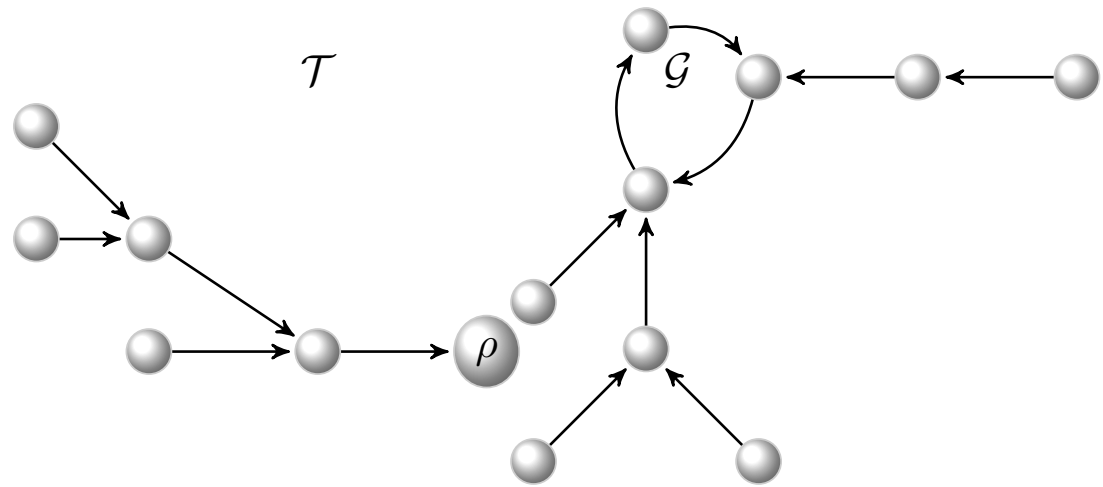
9

Another TOWER-hard variant [Mansutti, FST&TCS'18]

- Formulae:

$$\phi ::= T(\mathtt{u}) \mid G(\mathtt{u}) \mid \phi \wedge \phi \mid \neg\phi \mid \exists \mathtt{u}\ \phi \mid \phi * \phi$$

- Models $(\mathcal{T}, \mathcal{G})$ where $\mathcal{T}$ is a tree with root ρ and $\mathcal{G}$ is the garbage heap.



- $T(\mathtt{u})$ holds true when $\mathtt{u}$ is on tree whereas $G(\mathtt{u})$ holds true when $\mathtt{u}$ is in the garbage heap.

- TOWER-hardness by reduction from PITL.

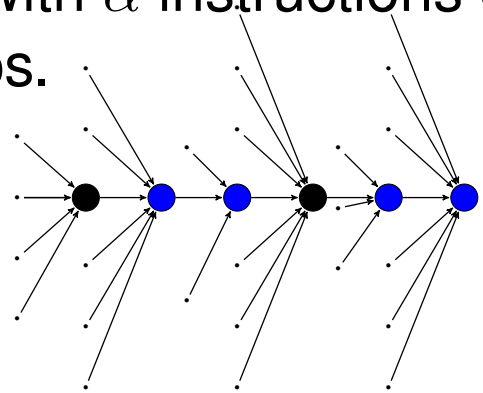
Undecidability of 1SL2 [Demri & Deters, ToCL 2015]

- 1SL2 formulae:

$$\mathtt{e} ::= \mathtt{x}_i \mid \mathtt{u}_0 \mid \mathtt{u}_1 \qquad \pi ::= \mathtt{e} = \mathtt{e}' \mid \mathtt{e} \hookrightarrow \mathtt{e}'$$

$$\phi ::= \pi \mid \phi \wedge \psi \mid \neg\phi \mid \phi * \psi \mid \phi \mathrel{-\!\!*} \psi \mid \exists\, \mathtt{u}_0\ \phi \mid \exists\, \mathtt{u}_1\ \phi$$

- Reduction from the halting problem for Minsky machines.
- Runs of a machine with α instructions can be encoded as $(\alpha, 2)$-fishbone heaps.



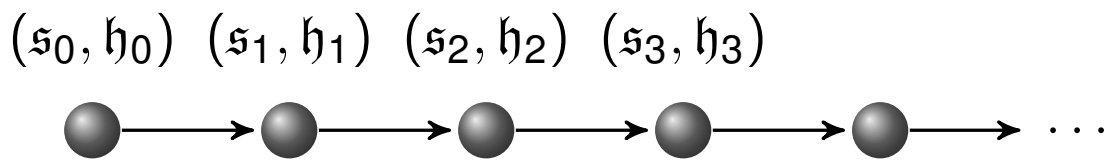
- Limit conditions and instructions are easy to take care of.
- The main difficulty is to encode in 1SL2 comparisons between number of predecessors:

$$\widetilde{\sharp}\mathfrak{l} = \widetilde{\sharp}\mathfrak{l}' \quad \text{or} \quad \widetilde{\sharp}\mathfrak{l} = \widetilde{\sharp}\mathfrak{l}' + 1.$$

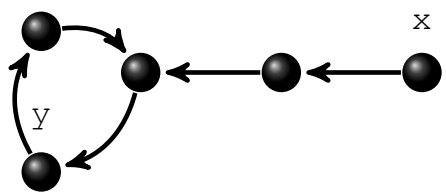
Modalities with separating connectives

Two-dimensional models or not ?

- Two-dimensional models:
  - To combine an assertion language from separation logic with linear-time/branching-time temporal logics.
  - Recently, interval temporal logics for memory states in [Lu & Tian & Duan, IJCAI'17].



- Uniform framework
  - Modal/temporal separation logics: Kripke-style semantics with modalities and separating connectives.



E.g.: $(\mathtt{ls}(\mathtt{x},\mathtt{y}) * \top) \wedge @_{\mathtt{x}}\mathsf{EF}\mathtt{y}$

  - To design modal/temporal logics with separating connectives as an alternative to FO separation logics.

Modal separation logics

Modal separation logic $\mathrm{MSL}(*, \Diamond, \langle\neq\rangle)$

- Fascinating realm of logics updating models:
  - sabotage modal logics [van Benthem, 2002]
  - relation-changing modal logics [Fervari, PhD 2014]
  - modal separation logic DMBI [Courtault & Galmiche, JLC 2018]
  - logics with reactive Kripke semantics [Gabbay, AMAI, 2012]
  - etc.

- Formulae:

$$\phi ::= p \mid \mathtt{emp} \mid \neg\phi \mid \phi \vee \phi \mid \Diamond\phi \mid \langle\neq\rangle\phi \mid \phi * \phi$$

- Models $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$:
  - $\mathfrak{R} \subseteq \mathbb{N} \times \mathbb{N}$ is finite and weakly functional (deterministic),
  - $\mathfrak{V} : \mathrm{PROP} \to \mathcal{P}(\mathbb{N})$.

- Disjoint unions $\mathfrak{M}_1 \uplus \mathfrak{M}_2$.

Semantics

$$\mathfrak{M}, \mathfrak{l} \models p \quad \overset{\text{def}}{\Leftrightarrow} \quad \mathfrak{l} \in \mathfrak{V}(p)$$

$$\mathfrak{M}, \mathfrak{l} \models \Diamond\phi \quad \overset{\text{def}}{\Leftrightarrow} \quad \mathfrak{M}, \mathfrak{l}' \models \phi, \text{ for some } \mathfrak{l}' \in \mathbb{N} \text{ such that } (\mathfrak{l}, \mathfrak{l}') \in \mathfrak{R}$$

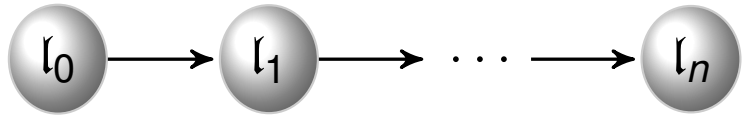
$$\mathfrak{M}, \mathfrak{l} \models \langle\neq\rangle\phi \quad \overset{\text{def}}{\Leftrightarrow} \quad \mathfrak{M}, \mathfrak{l}' \models \phi, \text{ for some } \mathfrak{l}' \in \mathbb{N} \text{ such that } \mathfrak{l}' \neq \mathfrak{l}$$

$$\mathfrak{M}, \mathfrak{l} \models \mathtt{emp} \quad \overset{\text{def}}{\Leftrightarrow} \quad \mathfrak{R} = \emptyset$$

$$\mathfrak{M}, \mathfrak{l} \models \phi_1 * \phi_2 \quad \overset{\text{def}}{\Leftrightarrow} \quad \langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, \mathfrak{l} \models \phi_1 \text{ and } \langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V} \rangle, \mathfrak{l} \models \phi_2,$$
for some partition $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ of $\mathfrak{R}$

Towards lower bounds: encoding linear structures

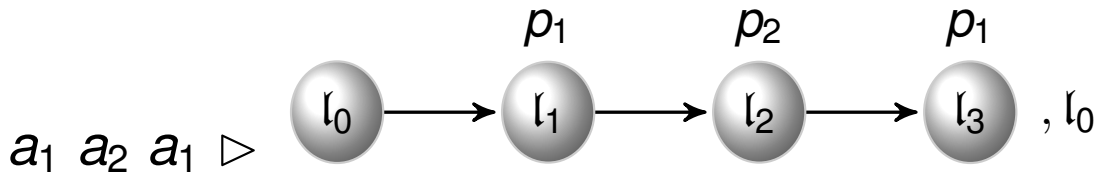
- Linear model:



- There is a formula $\phi_{\exists \mathtt{ls}}$ in $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ such that $\mathfrak{M} \models \phi_{\exists \mathtt{ls}}$ iff $\mathfrak{M}$ is linear.
- Star-free expressions

$$e ::= a \mid \varepsilon \mid e \cup e \mid ee \mid \sim e$$

  - Nonemptiness problem is TOWER-complete.
    [Meyer & Stockmeyer, STOC'73; Schmitz, ToCT 2016]
  - Encoding words by linear models.



- $\mathrm{MSL}(*, \Diamond, \langle \neq \rangle)$ satisfiability problem is TOWER-hard.
  [Demri & Fervari, AiML'18]

Two-dimensional models

Constrained LTL for memory states

- To design temporal languages to specify the behaviors of pointer programs.
- To combine an assertion language from separation logic with linear-time/branching-time temporal logics.
- To evaluate the borders for decidability.
- In the spirit of description/temporal logics over concrete domains.

  See e.g., [Lutz et al., TIME'08; Demri & D'Souza, IC 07]

The logic $\mathrm{LTL}^{\mathrm{mem}}$ [APAL 2009]

- Syntax

$$
\begin{array}{lll}
\mathtt{e} ::= & \mathtt{x} \mid \mathtt{null} \mid \mathsf{X}\mathtt{e} & \text{(expressions)} \\
\pi ::= & \mathtt{e} = \mathtt{e}' \mid \mathtt{e} + i \hookrightarrow \mathtt{e} & \text{(atomic formulae)} \\
\phi ::= & \pi \mid \phi \wedge \psi \mid \neg\phi \mid & \text{(classical fragment)} \\
& \phi * \psi \mid \phi \mathrel{-\!\!*} \psi \mid \mathtt{emp} & \text{(spatial fragment)} \\
\phi ::= & \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi' \mid \phi \wedge \phi' \mid \neg\phi & \text{(temporal formulae)}
\end{array}
$$

- Examples

$$\mathsf{G}\,(\mathtt{alloc(x)} \Rightarrow \mathsf{F}\,\mathtt{alloc(y)})$$

$$\mathsf{GF}(\mathtt{size} \geq 2) \quad (\mathsf{X}\mathtt{x} = \mathtt{x})\mathsf{U}(\mathtt{y} \hookrightarrow \mathtt{z})$$

Semantics

Models are sequences of memory states: $\mathfrak{M} = (\mathfrak{s}_t, \mathfrak{h}_t)_{t \geq 0}$.

$\mathfrak{M}, t \models \quad \mathsf{e} = \mathsf{e}'$ iff $[\![\mathsf{e}]\!]_{\mathfrak{M},t} = [\![\mathsf{e}']\!]_{\mathfrak{M},t}$ *with* $[\![X\mathsf{e}]\!]_{\mathfrak{M},t} = [\![\mathsf{e}]\!]_{\mathfrak{M},t+1}$

$\mathfrak{M}, t \models \quad \mathsf{e} + i \hookrightarrow \mathsf{e}'$ iff $\mathfrak{h}_t([\![\mathsf{e}]\!]_{\mathfrak{M},t} + i) = [\![\mathsf{e}']\!]_{\mathfrak{M},t}$

$\mathfrak{M}, t \models \quad \phi_1 * \phi_2$ iff $\exists\, \mathfrak{h}_1, \mathfrak{h}_2$ s.t. $\mathfrak{h}_t = \mathfrak{h}_1 * \mathfrak{h}_2$,
$\mathfrak{M}[\mathfrak{h}_t \leftarrow \mathfrak{h}_1], t \models \phi_1$,
and $\mathfrak{M}[\mathfrak{h}_t \leftarrow \mathfrak{h}_2], t \models \phi_2$.

$\mathfrak{M}, t \models \quad \phi_1 \mathrel{-\!\!*} \phi_2$ iff $\forall \mathfrak{h}'$, if $\mathfrak{h}_t \perp \mathfrak{h}'$ and $\mathfrak{M}[\mathfrak{h}_t \leftarrow \mathfrak{h}'], t \models \phi_1$
then $\mathfrak{M}[\mathfrak{h}_t \leftarrow \mathfrak{h} * \mathfrak{h}'], t \models \phi_2$.

$\mathfrak{M}, t \models \quad \mathsf{X}\phi$ iff $\mathfrak{M}, t+1 \models \phi$.

$\mathfrak{M}, t \models \quad \phi_1 \mathsf{U} \phi_2$ iff $\exists t' \geq t$ such that $\mathfrak{M}, t' \models \phi_2$,
and $\forall t'',\ t \leq t'' < t',\ \mathfrak{M}, t'' \models \phi_1$.

Satisfiability problems

- Satisfiability problem $\mathrm{SAT}(\mathrm{Frag})$: restriction to the fragment Frag.
- Satisfiability problem $\mathrm{SAT}^{ct}(\mathrm{Frag})$ with constant heap.
  $\rightarrow$ temporal language allows us to explore the heap.
- Satisfiability problem $\mathrm{SAT}_{init}(\mathrm{Frag})$ with a fixed initial heap.

A class of programs manipulating pointers

- Set of instructions

$$\begin{aligned}\texttt{instr} ::= \;& \texttt{x} := \texttt{y} \mid \texttt{skip} \\ & \mid \texttt{x} := [\texttt{y}] \mid [\texttt{x}] := \texttt{y} \\ & \mid \texttt{x} := \texttt{cons}(\texttt{x}_1) \mid \texttt{dispose}(\texttt{x}) \\ & \mid \texttt{x} := \texttt{y}[i] \mid \texttt{x}[i] := \texttt{y} \\ & \mid \texttt{x} = \texttt{malloc}(i) \mid \texttt{dispose}(\texttt{x} + i)\end{aligned}$$

- Programs are finite-state automata with transitions labelled by instructions and equality tests.

- A program without destructive update admits runs with constant heap.

Model-checking problems

- $\mathrm{MC}(\mathrm{Frag})$:

  input: formula in Frag, and program $\mathtt{PROG}$ of the associated fragment.

  question: is there an infinite computation $\mathfrak{M}$ of $\mathtt{PROG}$ such that $\mathfrak{M}, 0 \models \phi$?

- $\mathrm{MC}^{ct}_{init}(\mathrm{Frag})$: idem with fixed initial memory state and no destructive update.

Fragments with decidable temporal reasoning

- SL fragments:

Classical fragment (CL)

$$\phi ::= \quad \mathtt{e} = \mathtt{e}' \mid \mathtt{x} + i \hookrightarrow \mathtt{e} \mid \phi \wedge \phi \mid \neg\phi$$

Record fragment (RF)

$$\phi ::= \quad \mathtt{e} = \mathtt{e}' \mid \mathtt{x} \hookrightarrow \mathtt{e} \mid \phi * \phi \mid \phi \mathrel{-\!\!*} \phi \mid \mathtt{emp} \mid \phi \wedge \phi \mid \neg\phi$$

- The satisfiability problems for $\mathrm{LTL}^{\mathrm{mem}}$(CL) and $\mathrm{LTL}^{\mathrm{mem}}$(RF) are PSPACE-complete.

- Decidable model-checking problems.
  - $\mathrm{MC}^{ct}_{init}(\mathrm{RF})$ is PSPACE-complete.
    Proof by reduction into $\mathrm{SAT}^{ct}_{init}(\mathrm{RF})$.
  - $\mathrm{MC}^{ct}_{init}(\mathrm{SL})$ is PSPACE-complete.
    Proof by reduction into LTL model-checking.

Undecidability

- $\mathrm{SAT}(\mathrm{SL} \setminus \mathrel{-\!\!*})$ are Σ^1_1-complete.
- Reduction from the recurrence problem for non-deterministic Minsky machines.

[Alur & Henzinger, JACM 94]

- Incrementation is encoded thanks to

$$(\mathsf{X}\mathtt{x} \hookrightarrow \mathtt{y} \ \wedge \ \mathtt{x}+1 \hookrightarrow \mathtt{y}) \ \wedge \ \neg(\mathsf{X}\mathtt{x} \hookrightarrow \mathtt{y} \ * \ \mathtt{x}+1 \hookrightarrow \mathtt{y})$$

Concluding remarks

- Separation logics share many features with modal/temporal logics.
- Close relationships with interval temporal logics.
- See also relationships with ambient logic on trees.

  [Calcagno et al., TLDI'03; Calcagno et al., POPL'05]
- More to be done and to be understood.
  - Design of adequate temporal separation logics is still open.
  - How to add fixpoint operators in temporal separation logics? (for instance to tame reachability predicates such as $\langle * \rangle$)
  - Design of proof systems. (axiomatisation for $\mathrm{MSL}(*, \Diamond)$: work in progress with R. Fervari and A. Mansutti).