Separation Logic and Logics with Team Semantics

LoDE 2020V

Erich Grädel, Darion Haase, Richard Wilke

10th of August 2020

Section 1

Separation Logic

Separation Logic – Introduction

Program verification using Hoare logic: first-order logic with standard Tarskian-semantics

$$\{\mathit{precondition}\}\mathit{code}\{\mathit{postcondition}\}$$

Separation Logic – Introduction

Program verification using Hoare logic: first-order logic with standard Tarskian-semantics

$$\{\mathit{precondition}\}\mathit{code}\{\mathit{postcondition}\}$$

- Problem: only static memory/datastructures supported.

Separation Logic – Introduction

Program verification using Hoare logic: first-order logic with standard Tarskian-semantics

$$\{\mathit{precondition}\}\mathit{code}\{\mathit{postcondition}\}$$

- Problem: only static memory/datastructures supported.

Solution: Separation Logic

Separation Logic – Introduction

Program verification using Hoare logic: first-order logic with standard Tarskian-semantics

$$\{\mathit{precondition}\}\mathit{code}\{\mathit{postcondition}\}$$

- Problem: only static memory/datastructures supported.

Solution: Separation Logic

- Extend model by a heap to reason about dynamic data structures.

Separation Logic – Introduction

Program verification using Hoare logic: first-order logic with standard Tarskian-semantics

$$\{\mathit{precondition}\}\mathit{code}\{\mathit{postcondition}\}$$

- Problem: only static memory/datastructures supported.

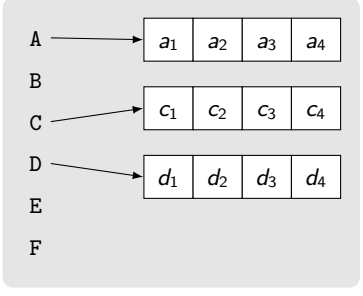
Solution: Separation Logic

- Extend model by a heap to reason about dynamic data structures.
- Extend logic by atoms and connectives from logic of Bunched Implications to reason about this heap.
  - Extending the heap
  - Splitting the heap

Heaps

Definition

Heap: A unary, finite, partial function $\mathfrak{h} \colon A \rightharpoonup A^k$. The domain of a heap is denoted by $\mathrm{dom}\,(\mathfrak{h})$.

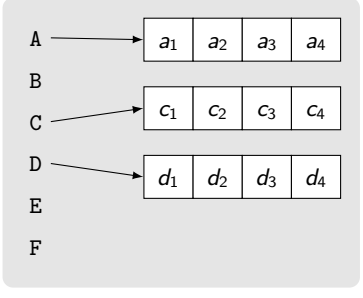


$$\mathrm{dom}\,(\mathfrak{h}) = \{\, A, C, D \,\}$$

Heaps

Definition

Heap: A unary, finite, partial function $\mathfrak{h} \colon A \rightharpoonup A^k$. The domain of a heap is denoted by $\mathrm{dom}\,(\mathfrak{h})$.



$$\mathrm{dom}\,(\mathfrak{h}) = \{\, A, C, D \,\}$$

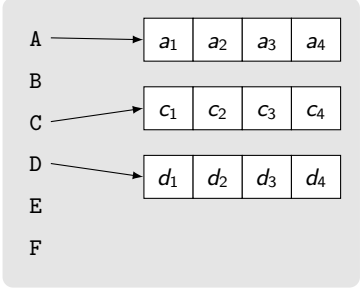
Definition

SL^k-interpretations consist of:

Heaps

Definition

Heap: A unary, finite, partial function $\mathfrak{h} \colon A \rightharpoonup A^k$. The domain of a heap is denoted by $\mathrm{dom}\,(\mathfrak{h})$.



$$\mathrm{dom}\,(\mathfrak{h}) = \{\, A, C, D \,\}$$

Definition

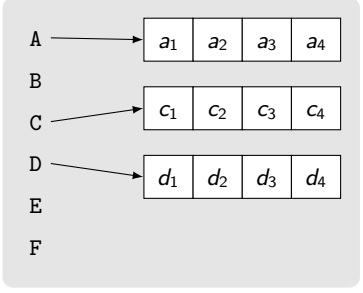
SL^k-interpretations consist of:

- A first-order structure $\mathfrak{A}$ with universe A,

Heaps

Definition

Heap: A unary, finite, partial function $\mathfrak{h} \colon A \rightharpoonup A^k$. The domain of a heap is denoted by $\mathrm{dom}\,(\mathfrak{h})$.



$$\mathrm{dom}\,(\mathfrak{h}) = \{\, A, C, D \,\}$$

Definition

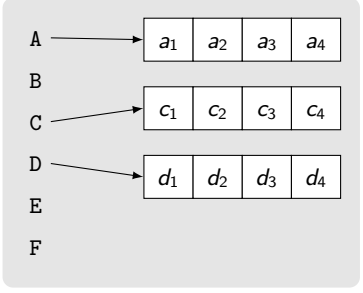
SL^k-interpretations consist of:

- A first-order structure $\mathfrak{A}$ with universe A,
- an assignment (store) $s \colon \mathtt{Var} \to A$, and

Heaps

Definition

Heap: A unary, finite, partial function $\mathfrak{h} \colon A \rightharpoonup A^k$. The domain of a heap is denoted by $\mathrm{dom}\,(\mathfrak{h})$.



$$\mathrm{dom}\,(\mathfrak{h}) = \{\, A, C, D \,\}$$

Definition

SL^k-interpretations consist of:

- A first-order structure $\mathfrak{A}$ with universe A,
- an assignment (store) $s \colon \mathtt{Var} \to A$, and
- a heap $\mathfrak{h}$.

Formulas of Separation Logic

SL^k extends first-order logic with new atoms and connectives to reason about a heap.

Formulas of Separation Logic

SL^k extends first-order logic with new atoms and connectives to reason about a heap.

Emptiness atom:

$$\mathfrak{A}, \mathfrak{h} \models_s \textbf{emp} \iff \mathfrak{h} = \emptyset$$

A

B

C

D

$$\mathfrak{A}, \mathfrak{h} \models \textbf{emp}$$

Formulas of Separation Logic

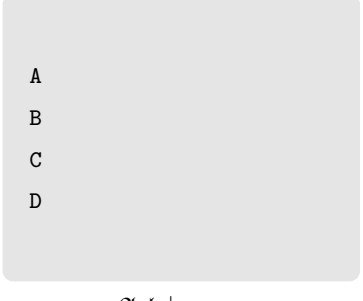
SL^k extends first-order logic with new atoms and connectives to reason about a heap.

Emptiness atom:

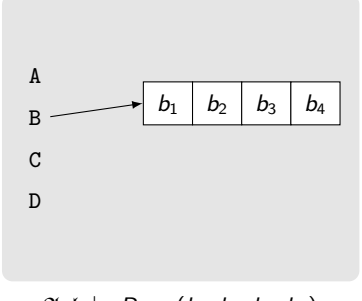
$$\mathfrak{A}, \mathfrak{h} \models_s \textbf{emp} \iff \mathfrak{h} = \emptyset$$

Points-to atom:

$$\mathfrak{A}, \mathfrak{h} \models_s x \mapsto (y_1, \ldots, y_k) \iff \mathfrak{h} = \{ (s(x), (s(y_1), \ldots, s(y_k))) \}$$



$\mathfrak{A}, \mathfrak{h} \models \textbf{emp}$



$\mathfrak{A}, \mathfrak{h} \models B \mapsto (b_1, b_2, b_3, b_4)$

Subsection 2

Splitting the Heap

Separating Conjunction and Separating Disjunction

Definition

- Two heaps $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are disjoint, if $\mathrm{dom}(\mathfrak{h}_1) \cap \mathrm{dom}(\mathfrak{h}_2) = \emptyset$.
- Disjoint heaps $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are a split of $\mathfrak{h}$, if $\mathfrak{h} = \mathfrak{h}_1 \dot{\cup} \mathfrak{h}_2$.

Separating Conjunction and Separating Disjunction

Definition

- Two heaps $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are disjoint, if $\mathrm{dom}(\mathfrak{h}_1) \cap \mathrm{dom}(\mathfrak{h}_2) = \emptyset$.
- Disjoint heaps $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are a split of $\mathfrak{h}$, if $\mathfrak{h} = \mathfrak{h}_1 \dot{\cup} \mathfrak{h}_2$.

Separating Conjunction:

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi * \psi :\Longleftrightarrow \text{there is a split } (\mathfrak{h}_1, \mathfrak{h}_2) \text{ of } \mathfrak{h} \text{ with } \mathfrak{A}, \mathfrak{h}_1 \models_s \varphi \text{ and } \mathfrak{A}, \mathfrak{h}_2 \models_s \psi.$$

Separating Conjunction and Separating Disjunction

Definition

- Two heaps $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are disjoint, if $\operatorname{dom}(\mathfrak{h}_1) \cap \operatorname{dom}(\mathfrak{h}_2) = \emptyset$.
- Disjoint heaps $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are a split of $\mathfrak{h}$, if $\mathfrak{h} = \mathfrak{h}_1 \dot\cup \mathfrak{h}_2$.

Separating Conjunction:

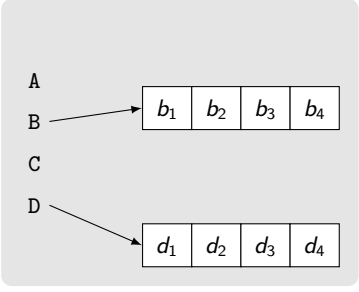
$$\mathfrak{A}, \mathfrak{h} \models_s \varphi * \psi :\Longleftrightarrow \text{there is a split } (\mathfrak{h}_1, \mathfrak{h}_2) \text{ of } \mathfrak{h} \text{ with } \mathfrak{A}, \mathfrak{h}_1 \models_s \varphi \text{ and } \mathfrak{A}, \mathfrak{h}_2 \models_s \psi.$$

Separating Disjunction:

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \circ \psi :\Longleftrightarrow \text{for all splits } (\mathfrak{h}_1, \mathfrak{h}_2) \text{ of } \mathfrak{h}\text{: } \mathfrak{A}, \mathfrak{h}_1 \models_s \varphi \text{ or } \mathfrak{A}, \mathfrak{h}_2 \models_s \psi.$$

Separating Conjunction – Example

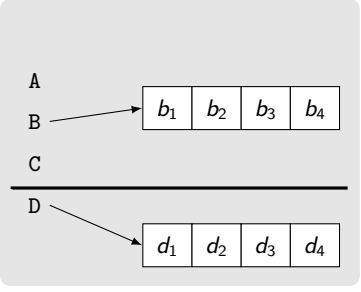
Heap $\mathfrak{h}$:



$$\mathfrak{A}, \mathfrak{h} \models B \mapsto (b_1, b_2, b_3, b_4) * D \mapsto (d_1, d_2, d_3, d_4)$$

Separating Conjunction – Example

Heap $\mathfrak{h}$:

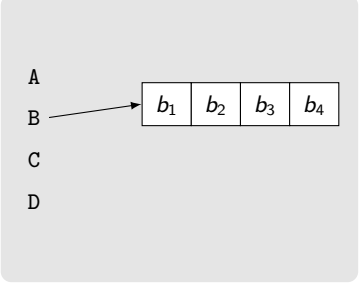


$$\mathfrak{A}, \mathfrak{h} \models B \mapsto (b_1, b_2, b_3, b_4) * D \mapsto (d_1, d_2, d_3, d_4)$$

Separating Conjunction – Example

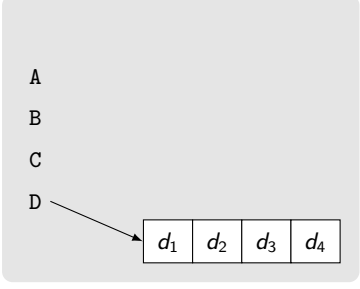
$$\mathfrak{h} = \mathfrak{h}_1 \dot{\cup} \mathfrak{h}_2$$

Heap $\mathfrak{h}_1$:



$\mathfrak{A}, \mathfrak{h}_1 \models B \mapsto (b_1, b_2, b_3, b_4)$

Heap $\mathfrak{h}_2$:



$\mathfrak{A}, \mathfrak{h}_2 \models D \mapsto (d_1, d_2, d_3, d_4)$

Subsection 3

Extending the Heap

Separating Implication and Septraction

Separating Implication (Magic Wand):

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \multimap\!\!* \psi :\Longleftrightarrow \text{for every disjoint heap } \mathfrak{h}_e \text{ with } \mathfrak{A}, \mathfrak{h}_e \models_s \varphi,$$
$$\text{also } \mathfrak{A}, (\mathfrak{h} \dot{\cup} \mathfrak{h}_e) \models_s \psi.$$

Separating Implication and Septraction

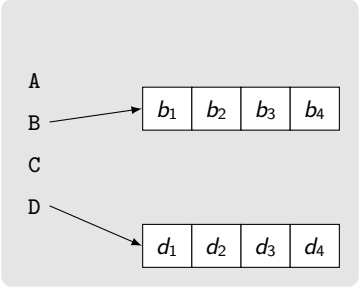
Separating Implication (Magic Wand):

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \mathrel{-\!\!*\!\!-} \psi :\Longleftrightarrow \text{for every disjoint heap } \mathfrak{h}_e \text{ with } \mathfrak{A}, \mathfrak{h}_e \models_s \varphi, \text{ also } \mathfrak{A}, (\mathfrak{h} \dot\cup \mathfrak{h}_e) \models_s \psi.$$

Septraction:

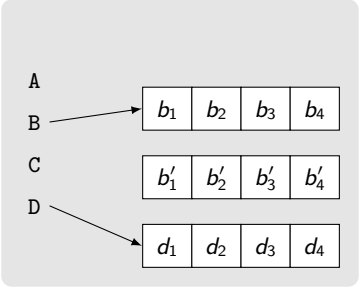
$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \multimap \psi :\Longleftrightarrow \text{there is a disjoint heap } \mathfrak{h}_e \text{ with } \mathfrak{A}, \mathfrak{h}_e \models_s \varphi \text{ and } \mathfrak{A}, (\mathfrak{h} \dot\cup \mathfrak{h}_e) \models_s \psi.$$

Separating Implication – Example



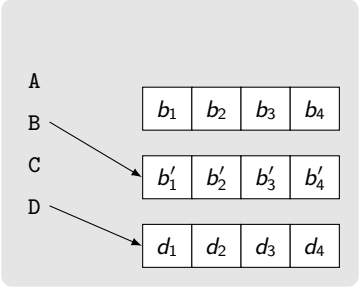
$$B \mapsto (b_1, b_2, b_3, b_4) * (B \mapsto (b'_1, b'_2, b'_3, b'_4) \mathrel{-\!\!*} \psi)$$

Separating Implication – Example



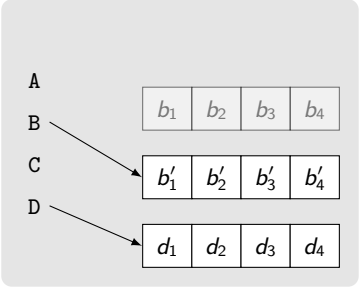
$$B \mapsto (b_1, b_2, b_3, b_4) * (B \mapsto (b'_1, b'_2, b'_3, b'_4) \mathrel{-\!\!*} \psi)$$

Separating Implication – Example



$$B \mapsto (b_1, b_2, b_3, b_4) * (B \mapsto (b'_1, b'_2, b'_3, b'_4) \multimap\!\!* \psi)$$

Separating Implication – Example



$$B \mapsto (b_1, b_2, b_3, b_4) * (B \mapsto (b'_1, b'_2, b'_3, b'_4) \mathrel{-\!\!*} \psi)$$

Section 2

Connections to Logics with Team Semantics

Team Semantics

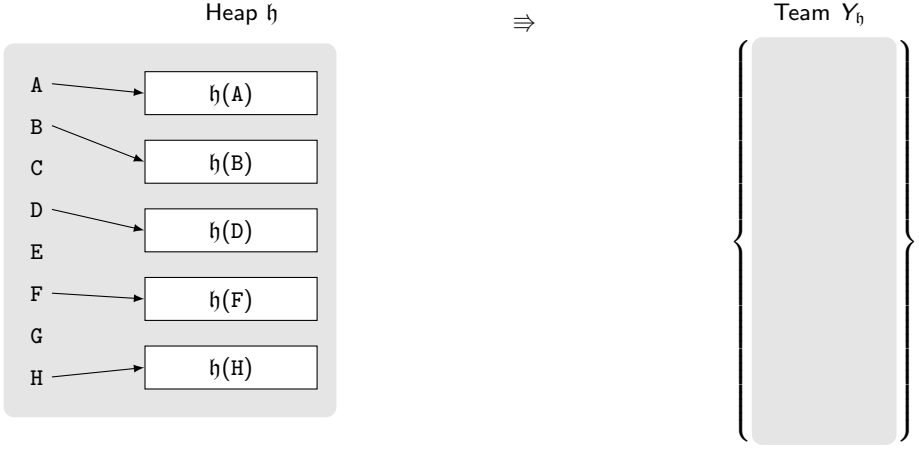
- Team Semantics for FO: Formulas are interpreted over teams: sets of assignments with a common domain.
- Variety of new atoms and connectives allow to reason about the team, e.g. dependence $\mathrm{dep}(\cdot;\cdot)$, independence $\perp$, ...

Definition

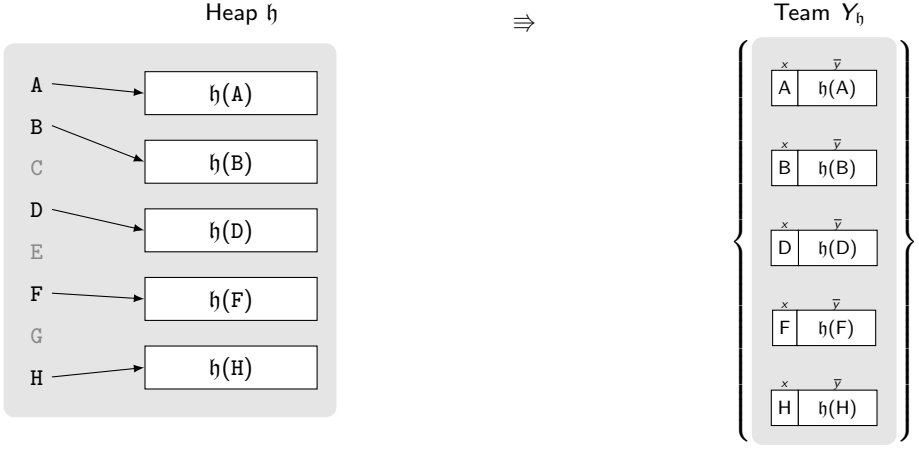
TS-interpretations consist of:

- A first-order structure $\mathfrak{A}$, and
- a team X with domain $\mathrm{dom}(X)$.

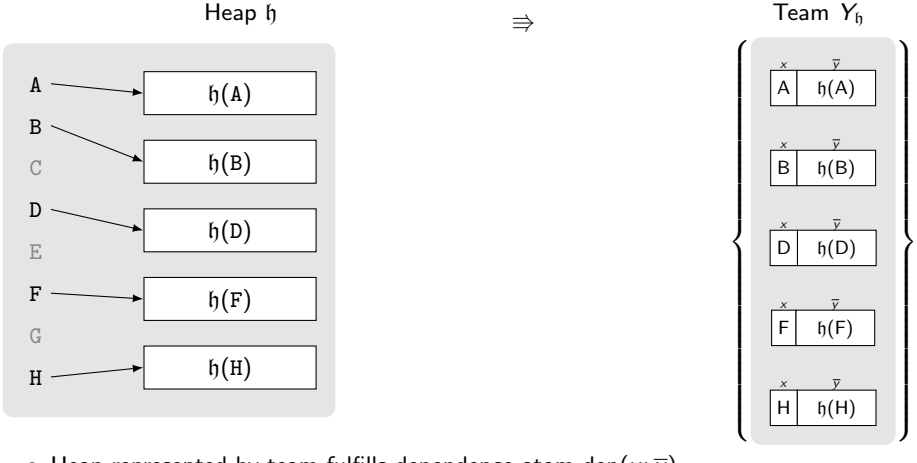
Connections and Differences between SL and TS



Connections and Differences between SL and TS

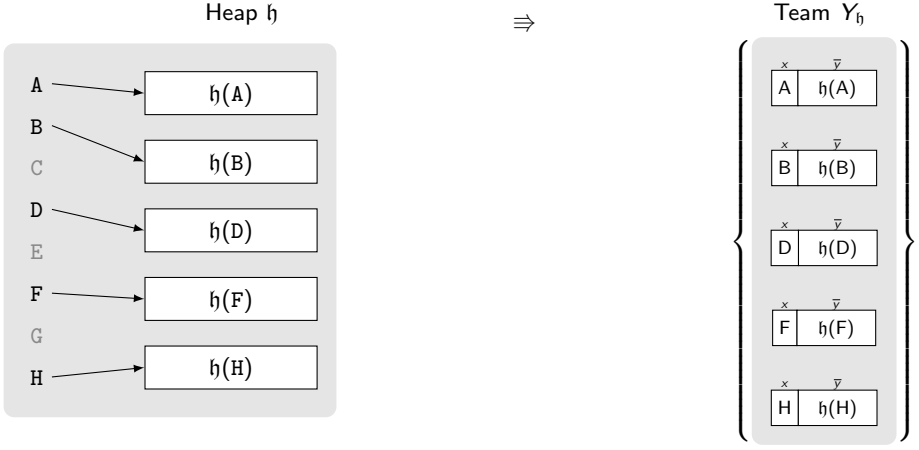


Connections and Differences between SL and TS



- Heap represented by team fulfills dependence atom $\mathrm{dep}(x; \overline{y})$

Connections and Differences between SL and TS



- Heap represented by team fulfills dependence atom $\mathrm{dep}(x; \overline{y})$
- Separating Conjunction ($*$) is similar to disjunction ($\vee$) of TS.

SL to TS – Goal

Goal: Compositional Translation of Separation Logic into a logic with Team Semantics

Find a translation of $\varphi \in \mathsf{SL}$ to a formula $\varphi^* \in \mathsf{TL}f\mathsf{SL}$ s.t. whenever $(\mathfrak{B}, X)$ represents $(\mathfrak{A}, \mathfrak{h}, s)$:

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \iff \mathfrak{B} \models_X \varphi^*$$

SL to TS – Goal

Goal: Compositional Translation of Separation Logic into a logic with Team Semantics

Find a translation of $\varphi \in \mathsf{SL}$ to a formula $\varphi^* \in \mathsf{TLfSL}$ s.t. whenever $(\mathfrak{B}, X)$ represents $(\mathfrak{A}, \mathfrak{h}, s)$:

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \iff \mathfrak{B} \models_X \varphi^*$$

(1) *Standard* TS-logics are not closed under negation.
$\Rightarrow$ Consider NNF formulas of Separation Logic.

SL to TS – Goal

Goal: Compositional Translation of Separation Logic into a logic with Team Semantics

Find a translation of $\varphi \in \mathsf{SL}$ to a formula $\varphi^* \in \mathsf{TL}f\mathsf{SL}$ s.t. whenever $(\mathfrak{B}, X)$ represents $(\mathfrak{A}, \mathfrak{h}, s)$:

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \iff \mathfrak{B} \models_X \varphi^*$$

(1) *Standard* TS-logics are not closed under negation.
$\Rightarrow$ Consider NNF formulas of Separation Logic.

(2) Also need to represent the assignment s of SL-interpretation in the team.

SL to TS – Goal

Goal: Compositional Translation of Separation Logic into a logic with Team Semantics

Find a translation of $\varphi \in \mathrm{SL}$ to a formula $\varphi^* \in \mathrm{TL}f\mathrm{SL}$ s.t. whenever $(\mathfrak{B}, X)$ represents $(\mathfrak{A}, \mathfrak{h}, s)$:

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \iff \mathfrak{B} \models_X \varphi^*$$

(1) *Standard* TS-logics are not closed under negation.
⇒ Consider NNF formulas of Separation Logic.

(2) Also need to represent the assignment s of SL-interpretation in the team.

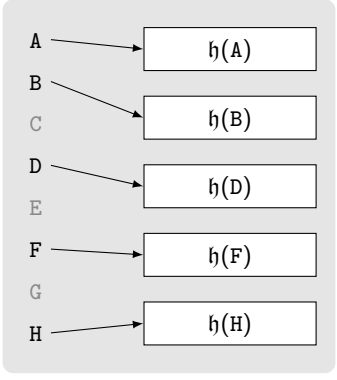
(3) Empty-team property of *standard* TS-logics: $\mathfrak{A} \models_\emptyset \varphi$ for all suitable φ.
⇒ Intuitive representation of empty heap by empty team not appropriate.

Section 3

Translation from Separation Logic into Team Semantics

Translation of Interpretations – Dummy Assignment

Heap $\mathfrak{h}$



$\Rrightarrow$

Team $Y_{\mathfrak{h}}$

x	$\overline{y}$
A	$\mathfrak{h}(A)$

x	$\overline{y}$
F	$\mathfrak{h}(F)$

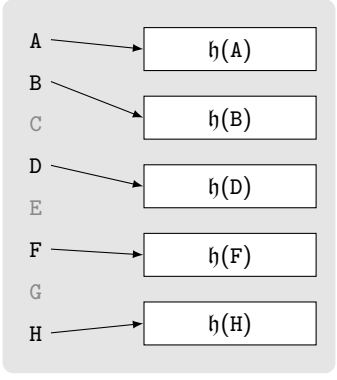
x	$\overline{y}$
B	$\mathfrak{h}(B)$

x	$\overline{y}$
H	$\mathfrak{h}(H)$

x	$\overline{y}$
D	$\mathfrak{h}(D)$

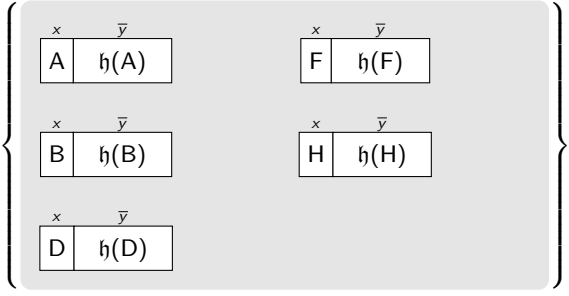
Translation of Interpretations – Dummy Assignment

Heap $\mathfrak{h}$ + Assignment s



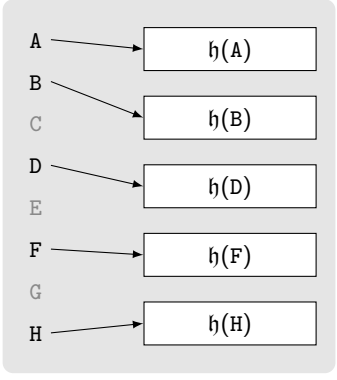
$\Rrightarrow$

Team $Y_{\mathfrak{h}}$



Translation of Interpretations – Dummy Assignment

Heap $\mathfrak{h}$ + Assignment s



$\Rrightarrow$

Team $Y_{\mathfrak{h},s}$

x	$\overline{y}$	$\overline{\mathrm{dom}(s)}$
A	$\mathfrak{h}(A)$	s

x	$\overline{y}$	$\overline{\mathrm{dom}(s)}$
F	$\mathfrak{h}(F)$	s

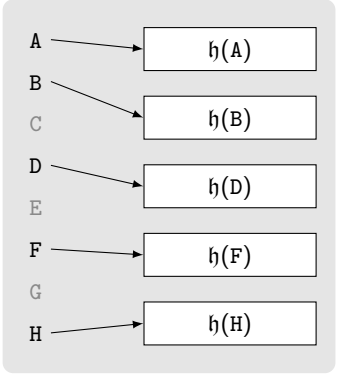
x	$\overline{y}$	$\overline{\mathrm{dom}(s)}$
B	$\mathfrak{h}(B)$	s

x	$\overline{y}$	$\overline{\mathrm{dom}(s)}$
H	$\mathfrak{h}(H)$	s

x	$\overline{y}$	$\overline{\mathrm{dom}(s)}$
D	$\mathfrak{h}(D)$	s

Translation of Interpretations – Dummy Assignment

Heap $\mathfrak{h}$ + Assignment s



$\Rrightarrow$

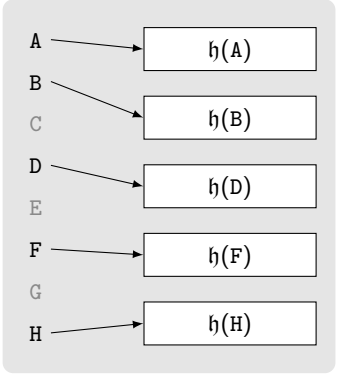
Team $Y_{\mathfrak{h},s}$

x	$\overline{y}$	$\overline{\mathrm{dom}(s)}$
A	$\mathfrak{h}(A)$	s
B	$\mathfrak{h}(B)$	s
D	$\mathfrak{h}(D)$	s
F	$\mathfrak{h}(F)$	s
H	$\mathfrak{h}(H)$	s

Add new dummy/nil value δ to structure $\mathfrak{A}$ to obtain structure $\mathfrak{A}^{\delta}$.

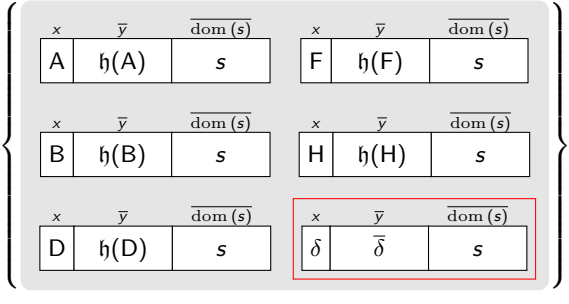
Translation of Interpretations – Dummy Assignment

Heap $\mathfrak{h}$ + Assignment s



$\Rrightarrow$

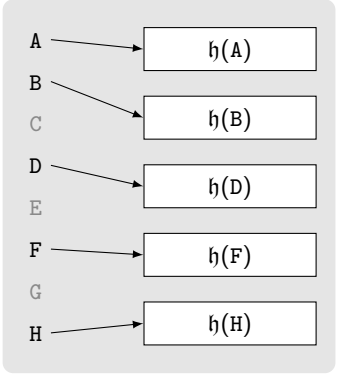
Team $X_{\mathfrak{h},s}$



Add new dummy/nil value δ to structure $\mathfrak{A}$ to obtain structure $\mathfrak{A}^{\delta}$.

Translation of Interpretations – Dummy Assignment

Heap $\mathfrak{h}$ + Assignment s



$\Rrightarrow$

Team $X_{\mathfrak{h},s}$

x	$\overline{y}$	$\overline{\mathrm{dom}(s)}$
A	$\mathfrak{h}(A)$	s
B	$\mathfrak{h}(B)$	s
D	$\mathfrak{h}(D)$	s

x	$\overline{y}$	$\overline{\mathrm{dom}(s)}$
F	$\mathfrak{h}(F)$	s
H	$\mathfrak{h}(H)$	s
δ	$\overline{\delta}$	s

Add new dummy/nil value δ to structure $\mathfrak{A}$ to obtain structure $\mathfrak{A}^{\delta}$.

Find a translation of $\varphi \in \mathrm{SL}$ (in negation normal form) to a formula $\varphi^* \in \mathrm{TL}f\mathrm{SL}$ s.t.

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \iff \mathfrak{A}^{\delta} \models_{X_{\mathfrak{h},s}} \varphi^*$$

TLfSL – Atoms and Connectives

What to include in TLfSL?

Dependence: $\mathfrak{A} \models_X \mathrm{dep}(\overline{x}; \overline{y}) :\Longleftrightarrow$ for all $s, s' \in X$ with $s(\overline{x}) = s'(\overline{x})$ also $s(\overline{y}) = s'(\overline{y})$.

Finiteness: $\mathfrak{A} \models_X \mathtt{Fin}(\overline{x}) :\Longleftrightarrow X(\overline{x})$ is finite.

Nonemptiness: $\mathfrak{A} \models_X \mathtt{NE} :\Longleftrightarrow X \neq \emptyset$.

Classical disjunction: $\mathfrak{A} \models_X \varphi \sqcup \psi :\Longleftrightarrow \mathfrak{A} \models_X \varphi$ or $\mathfrak{A} \models_X \psi$

Uniform quantification: $\mathfrak{A} \models_X \exists^1 u \psi :\Longleftrightarrow$ there exists $a \in \mathfrak{A}$, such that $\mathfrak{A} \models_{X[u \mapsto a]} \psi$,

$\mathfrak{A} \models_X \forall^1 u \psi :\Longleftrightarrow$ for all $a \in \mathfrak{A}$: $\mathfrak{A} \models_{X[u \mapsto a]} \psi$.

Dependent disjunction: $\mathfrak{A} \models_X \varphi \curlyvee_{\overline{x}} \psi :\Longleftrightarrow$ there is a disjoint decomposition $X = X_1 \dot{\cup} X_2$ satisfying $\mathfrak{A} \models_{X_1} \varphi, \mathfrak{A} \models_{X_2} \psi$ such that for all $s, s' \in X$, if $s \in X_i$ and $s(\overline{x}) = s'(\overline{x})$ then $s' \in X_i$.

Subsection 2

Separating Conjunction

Separating Conjunction

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi * \psi :\Longleftrightarrow \text{there is a split } (\mathfrak{h}_1, \mathfrak{h}_2) \text{ of } \mathfrak{h} \text{ with } \mathfrak{A}, \mathfrak{h}_1 \models_s \varphi \text{ and } \mathfrak{A}, \mathfrak{h}_2 \models_s \psi.$$

Separating Conjunction

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi * \psi :\Longleftrightarrow \text{there is a split } (\mathfrak{h}_1, \mathfrak{h}_2) \text{ of } \mathfrak{h} \text{ with } \mathfrak{A}, \mathfrak{h}_1 \models_s \varphi \text{ and } \mathfrak{A}, \mathfrak{h}_2 \models_s \psi.$$

- Similar to disjunction of Team Semantics, but always disjoint.

Separating Conjunction

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi * \psi :\Longleftrightarrow \text{there is a split } (\mathfrak{h}_1, \mathfrak{h}_2) \text{ of } \mathfrak{h} \text{ with } \mathfrak{A}, \mathfrak{h}_1 \models_s \varphi \text{ and } \mathfrak{A}, \mathfrak{h}_2 \models_s \psi.$$

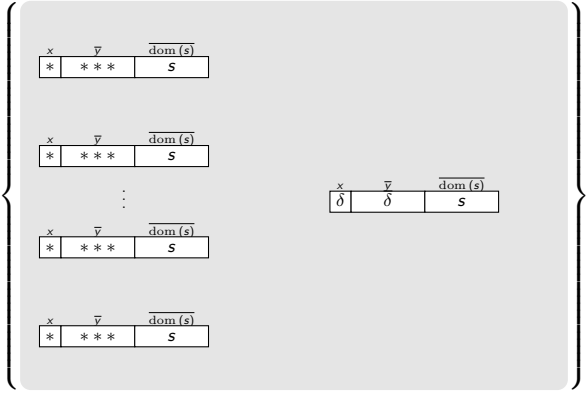
- Similar to disjunction of Team Semantics, but always disjoint.
- Heap is split depending on its domain ($\hat{=}$ the team's values of x).

Separating Conjunction

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi * \psi :\Longleftrightarrow \text{there is a split } (\mathfrak{h}_1, \mathfrak{h}_2) \text{ of } \mathfrak{h} \text{ with } \mathfrak{A}, \mathfrak{h}_1 \models_s \varphi \text{ and } \mathfrak{A}, \mathfrak{h}_2 \models_s \psi.$$

- Similar to disjunction of Team Semantics, but always disjoint.
- Heap is split depending on its domain ($\hat{=}$ the team's values of x).

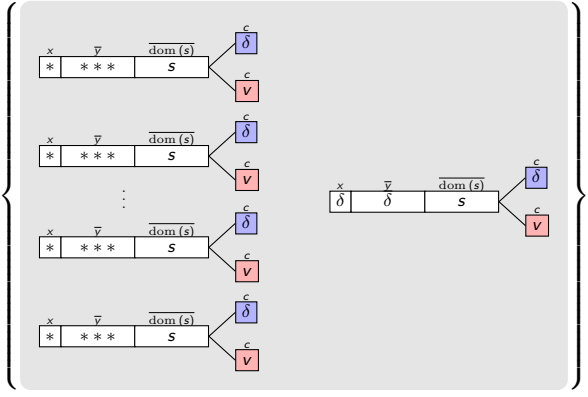
Separating Conjunction – 2



$\mathsf{split}(x, c) :=$

$(\varphi * \psi)^* := \exists c[\mathsf{split}(x, c)$]

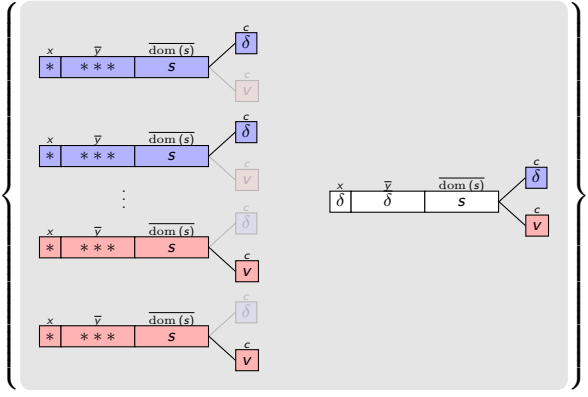
Separating Conjunction – 2



$\mathsf{split}(x, c) \coloneqq [(c = \delta \wedge \mathtt{NE}) \vee (c \neq \delta \wedge \mathsf{dep}(c) \wedge \mathtt{NE})]$

$(\varphi * \psi)^* \coloneqq \exists c[\mathsf{split}(x, c)$]

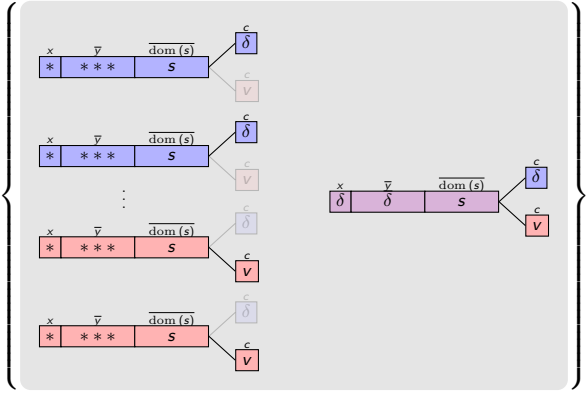
Separating Conjunction – 2



$\mathsf{split}(x, c) \coloneqq [(c = \delta \wedge \mathtt{NE}) \vee (c \neq \delta \wedge \mathsf{dep}(c) \wedge \mathtt{NE})] \wedge [(x \neq \delta \wedge \mathsf{dep}(x; c)) \vee$]

$(\varphi * \psi)^* \coloneqq \exists c[\mathsf{split}(x, c)$]

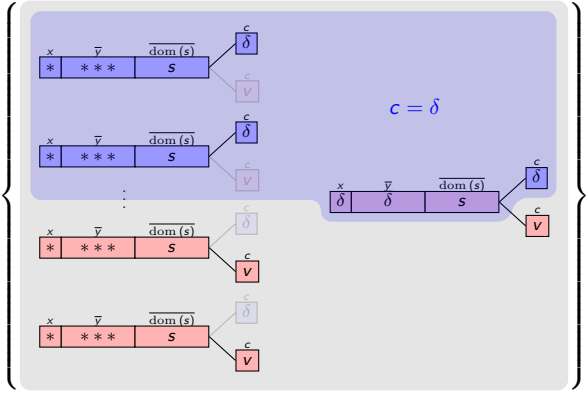
Separating Conjunction – 2



$$\mathsf{split}(x, c) \coloneqq [(c = \delta \wedge \mathtt{NE}) \vee (c \neq \delta \wedge \mathsf{dep}(c) \wedge \mathtt{NE})] \wedge [(x \neq \delta \wedge \mathsf{dep}(x; c)) \vee (x = \delta \wedge (\mathtt{NE}\ \Upsilon_c\ \mathtt{NE}))]$$

$$(\varphi * \psi)^* \coloneqq \exists c[\mathsf{split}(x, c) \qquad]$$

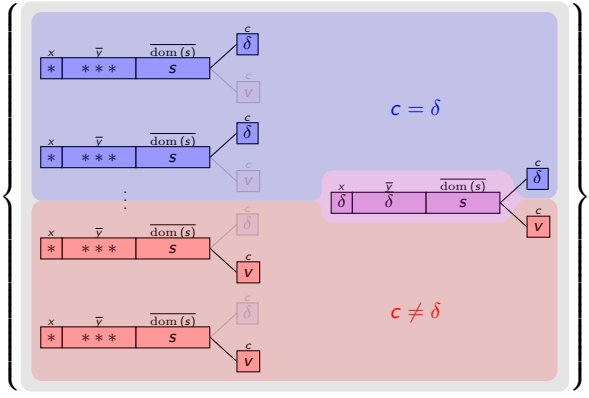
Separating Conjunction – 2



$$\mathsf{split}(x, c) \coloneqq [(c = \delta \wedge \mathtt{NE}) \vee (c \neq \delta \wedge \mathsf{dep}(c) \wedge \mathtt{NE})] \wedge [(x \neq \delta \wedge \mathsf{dep}(x; c)) \vee (x = \delta \wedge (\mathtt{NE} \curlyvee_c \mathtt{NE}))]$$

$$(\varphi * \psi)^* \coloneqq \exists c[\mathsf{split}(x, c) \wedge ((c = \delta \wedge \varphi^*) \vee (c \neq \delta \wedge \psi^*))]$$

Separating Conjunction – 2



$$\mathsf{split}(x, c) \coloneqq [(c = \delta \wedge \mathtt{NE}) \vee (c \neq \delta \wedge \mathsf{dep}(c) \wedge \mathtt{NE})] \wedge [(x \neq \delta \wedge \mathsf{dep}(x; c)) \vee (x = \delta \wedge (\mathtt{NE}\ \Upsilon_c\ \mathtt{NE}))]$$

$$(\varphi * \psi)^* \coloneqq \exists c[\mathsf{split}(x, c) \wedge ((c = \delta \wedge \varphi^*) \vee (c \neq \delta \wedge \psi^*))]$$

Subsection 3

Septraction

Septraction

$\mathfrak{A}, \mathfrak{h} \models_s \varphi \multimap \psi :\Longleftrightarrow$ there is a disjoint heap $\mathfrak{h}_\epsilon$ with $\mathfrak{A}, \mathfrak{h}_\epsilon \models_s \varphi$ and $\mathfrak{A}, (\mathfrak{h} \dot\cup \mathfrak{h}_\epsilon) \models_s \psi$.

Septraction

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \mathrel{-\!\!\circ} \psi :\Longleftrightarrow \text{there is a disjoint heap } \mathfrak{h}_\epsilon \text{ with } \mathfrak{A}, \mathfrak{h}_\epsilon \models_s \varphi \text{ and } \mathfrak{A}, (\mathfrak{h} \dot\cup \mathfrak{h}_\epsilon) \models_s \psi.$$

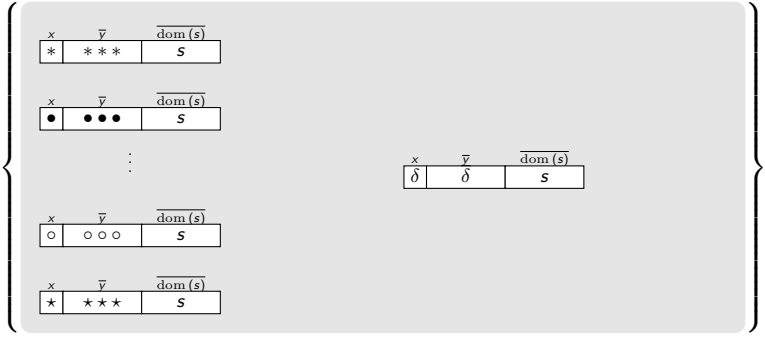
- Need to represent a disjoint extension of the current heap.

Septraction

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \mathrel{-\!\!\circ} \psi \coloneqq\!\iff \text{there is a disjoint heap } \mathfrak{h}_\epsilon \text{ with } \mathfrak{A}, \mathfrak{h}_\epsilon \models_s \varphi$$
$$\text{and } \mathfrak{A}, (\mathfrak{h} \dot{\cup} \mathfrak{h}_\epsilon) \models_s \psi.$$

- Need to represent a disjoint extension of the current heap.
- Use new variables $u, \overline{v}$ to represent the extended heap.

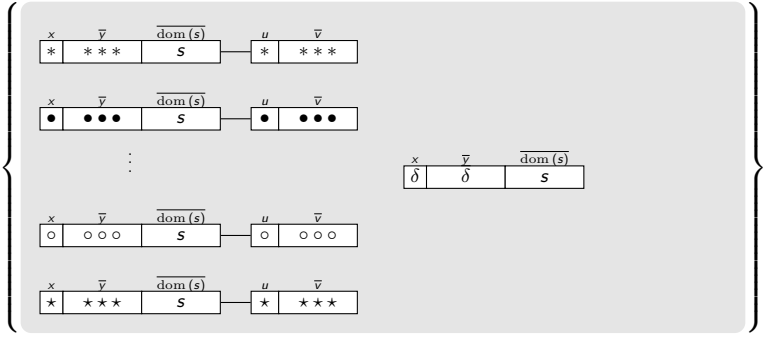
Septraction



$$\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \coloneqq [\qquad\qquad \vee \qquad\qquad]$$

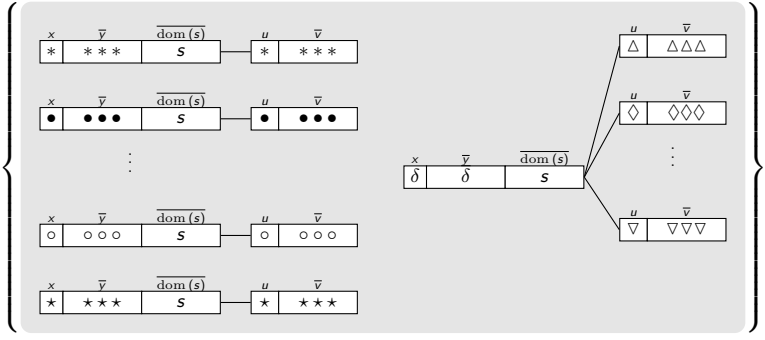
$$(\varphi \multimap \psi)^* \coloneqq \exists u \exists \overline{v}(\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \qquad\qquad)$$

Septraction



$$\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \coloneqq [(x \neq \delta \wedge u = x \wedge \overline{v} = \overline{y}) \vee \qquad]$$
$$(\varphi \multimap \psi)^* \coloneqq \exists u \exists \overline{v} (\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \qquad)$$

Septraction

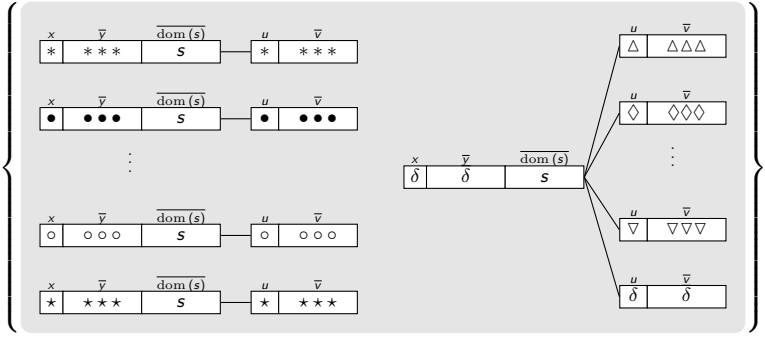


$\mathsf{heap}(u, \overline{v}) :=$

$\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) := [(x \neq \delta \wedge u = x \wedge \overline{v} = \overline{y}) \vee (x = \delta \wedge \mathsf{heap}(u, \overline{v})\quad)]$

$(\varphi \multimap \psi)^* := \exists u \exists \overline{v}(\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v})\quad)$

Septraction

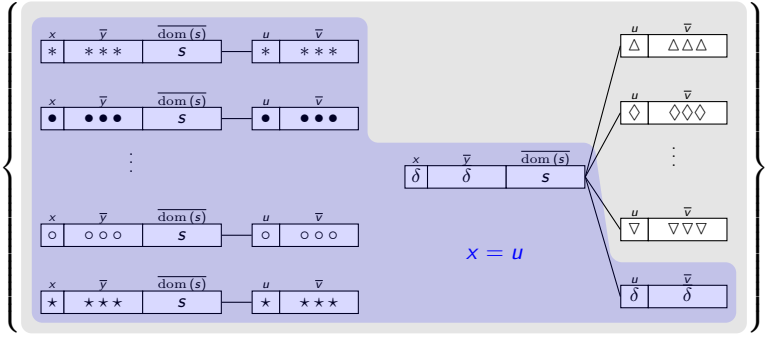


$$\mathsf{heap}(u, \overline{v}) \coloneqq \mathtt{dep}(u; \overline{v}) \wedge \mathtt{Fin}(u) \wedge [(u = \delta \wedge \overline{v} = \overline{\delta} \wedge \mathtt{NE}) \vee (u \neq \delta \wedge \bigwedge_{i=1}^{k} v_i \neq \delta)]$$

$$\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \coloneqq [(x \neq \delta \wedge u = x \wedge \overline{v} = \overline{y}) \vee (x = \delta \wedge \mathsf{heap}(u, \overline{v}) \quad)]$$

$$(\varphi \multimap \psi)^* \coloneqq \exists u \exists \overline{v} (\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \quad)$$

Septraction

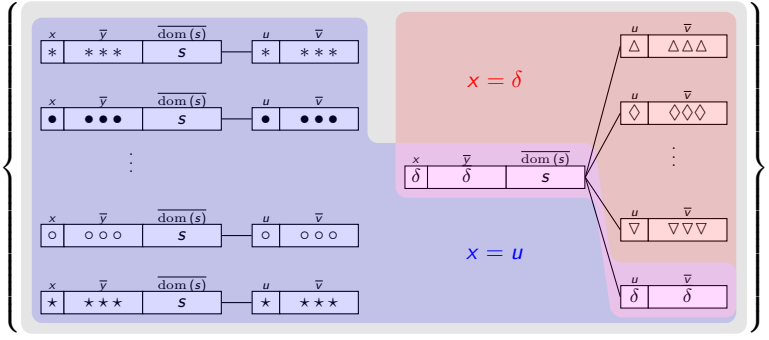


$$\mathsf{heap}(u, \overline{v}) \coloneqq \mathsf{dep}(u; \overline{v}) \wedge \mathtt{Fin}(u) \wedge [(u = \delta \wedge \overline{v} = \overline{\delta} \wedge \mathtt{NE}) \vee (u \neq \delta \wedge \bigwedge_{i=1}^{k} v_i \neq \delta)]$$

$$\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \coloneqq [(x \neq \delta \wedge u = x \wedge \overline{v} = \overline{y}) \vee (x = \delta \wedge \mathsf{heap}(u, \overline{v}) \quad)]$$

$$(\varphi \multimap \psi)^* \coloneqq \exists u \exists \overline{v} (\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \quad)$$

Septraction

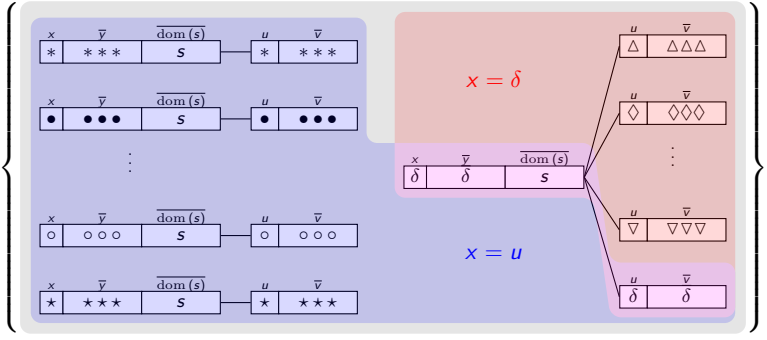


$$\mathsf{heap}(u, \overline{v}) := \mathsf{dep}(u; \overline{v}) \wedge \mathtt{Fin}(u) \wedge [(u = \delta \wedge \overline{v} = \overline{\delta} \wedge \mathtt{NE}) \vee (u \neq \delta \wedge \bigwedge_{i=1}^{k} v_i \neq \delta)]$$

$$\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) := [(x \neq \delta \wedge u = x \wedge \overline{v} = \overline{y}) \vee (x = \delta \wedge \mathsf{heap}(u, \overline{v}) \quad)]$$

$$(\varphi \multimap \psi)^* := \exists u \exists \overline{v} (\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \quad)$$

Septraction

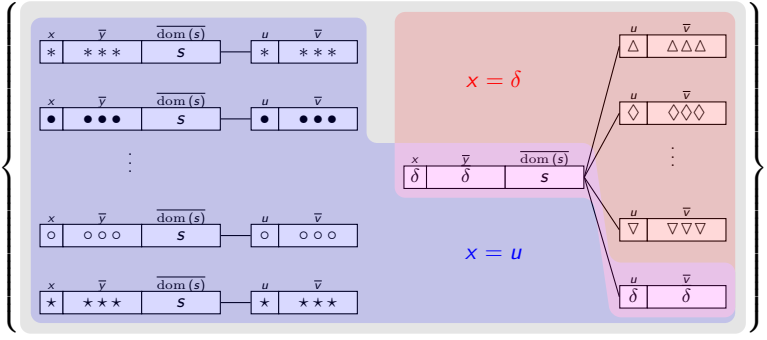


$$\mathsf{heap}(u, \overline{v}) := \mathsf{dep}(u; \overline{v}) \wedge \mathtt{Fin}(u) \wedge [(u = \delta \wedge \overline{v} = \overline{\delta} \wedge \mathtt{NE}) \vee (u \neq \delta \wedge \bigwedge_{i=1}^{k} v_i \neq \delta)]$$

$$\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) := [(x \neq \delta \wedge u = x \wedge \overline{v} = \overline{y}) \vee (x = \delta \wedge \mathsf{heap}(u, \overline{v}) \quad)] \wedge \mathsf{dep}(u; x)$$

$$(\varphi \multimap \psi)^* := \exists u \exists \overline{v} (\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \quad)$$

Septraction



$$\mathsf{heap}(u, \overline{v}) := \mathsf{dep}(u; \overline{v}) \wedge \mathtt{Fin}(u) \wedge [(u = \delta \wedge \overline{v} = \overline{\delta} \wedge \mathtt{NE}) \vee (u \neq \delta \wedge \bigwedge_{i=1}^{k} v_i \neq \delta)]$$

$$\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) := [(x \neq \delta \wedge u = x \wedge \overline{v} = \overline{y}) \vee (x = \delta \wedge \mathsf{heap}(u, \overline{v}) \wedge \varphi^*[u, \overline{v}])] \wedge \mathsf{dep}(u; x)$$

$$(\varphi \multimap \psi)^* := \exists u \exists \overline{v} (\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \wedge \psi^*[u, \overline{v}])$$

Section 4

Separating Implication and Separating Disjunction

Separating Implication and Separating Disjunction

Remaining, e.g. Magic Wand

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \mathrel{-\!\!*} \psi :\Longleftrightarrow \text{for every disjoint heap } \mathfrak{h}_\epsilon \text{ with } \mathfrak{A}, \mathfrak{h}_\epsilon \models_s \varphi, \text{ also } \mathfrak{A}, (\mathfrak{h} \dot\cup \mathfrak{h}_\epsilon) \models_s \psi.$$

Separating Implication and Separating Disjunction

Remaining, e.g. Magic Wand

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \mathrel{-\!\!*} \psi :\Longleftrightarrow \text{for every disjoint heap } \mathfrak{h}_\epsilon \text{ with } \mathfrak{A}, \mathfrak{h}_\epsilon \models_s \varphi, \text{ also } \mathfrak{A}, (\mathfrak{h} \dot\cup \mathfrak{h}_\epsilon) \models_s \psi.$$

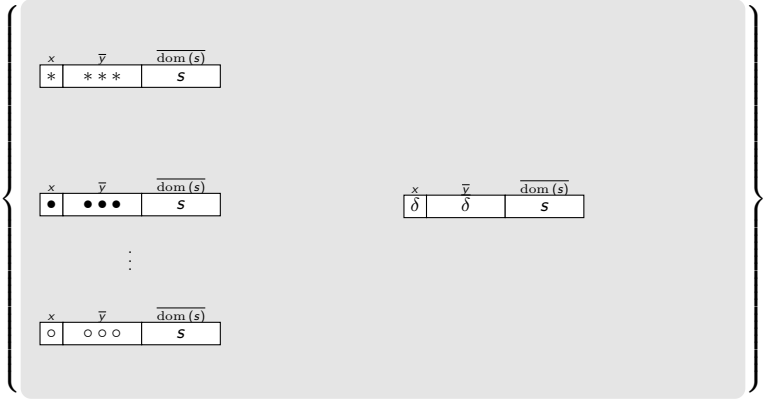
Approach:

Intuitionistic implication: $\mathfrak{A} \models_X \varphi \to \psi :\Longleftrightarrow$ for all $Y \subseteq X$ with $\mathfrak{A} \models_Y \varphi$ also $\mathfrak{A} \models_Y \psi$.

Use in combination with universal quantifiers to represent $\varphi \mathrel{-\!\!*} \psi$ and $\varphi \circ \psi$ based on the translation of the dual connectives:

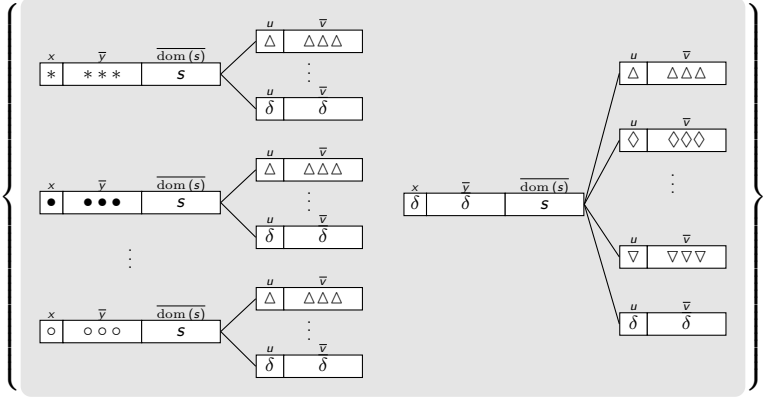
$$\varphi \mathrel{-\!\!\circ} \psi^* := \exists u \exists \overline{v} (\varphi_{\text{ext}}(x, \overline{y}, u, \overline{v}) \wedge \psi^*[u, \overline{v}])$$
$$\varphi \mathrel{-\!\!*} \psi^* \rightsquigarrow \forall u \forall \overline{v} (\varphi_{\text{ext}}(x, \overline{y}, u, \overline{v}) \to \psi^*[u, \overline{v}])$$

Magic Wand – First Try



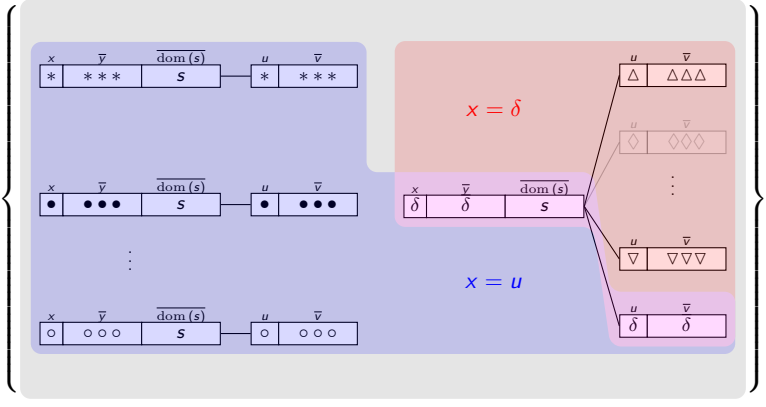
$$\forall u \forall \overline{v}(\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \to \psi^*[u, \overline{v}])$$

Magic Wand – First Try



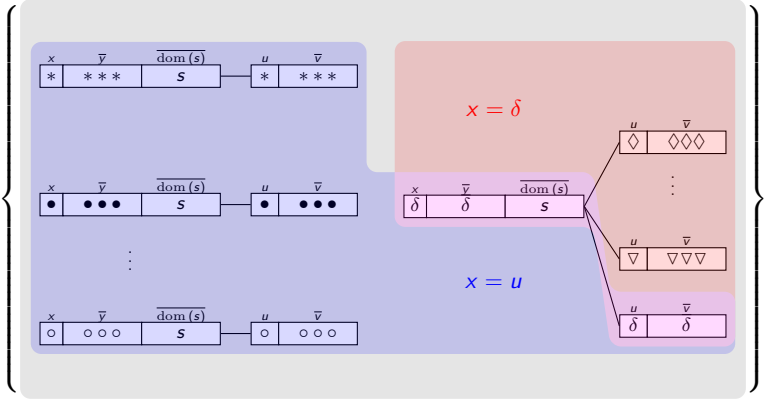
$$\forall u \forall \overline{v} (\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \to \psi^*[u, \overline{v}])$$

Magic Wand – First Try



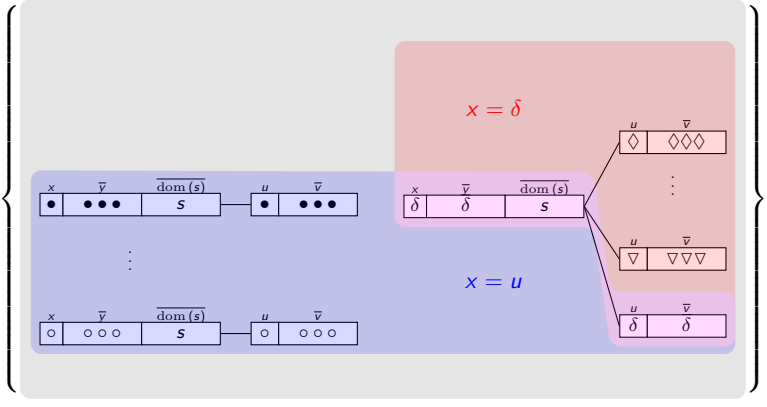
$$\forall u \forall \overline{v} (\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \to \psi^*[u, \overline{v}])$$

Magic Wand – First Try



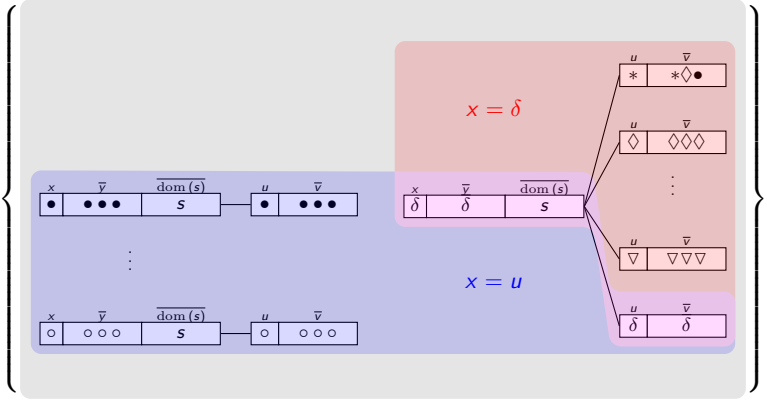
$$\forall u \forall \overline{v}(\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \rightarrow \psi^*[u, \overline{v}])$$

Magic Wand – First Try



$$\forall u \forall \overline{v}(\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \to \psi^*[u, \overline{v}])$$

Magic Wand – First Try



$$\forall u \forall \overline{v} (\varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v}) \to \psi^*[u, \overline{v}])$$

Preventing Loss of Heap Content

Problem of considering teams which do not represent extensions of the original heap $\mathfrak{h}$.

Preventing Loss of Heap Content

Problem of considering teams which do not represent extensions of the original heap $\mathfrak{h}$.
$\Rightarrow$ Force all assignments representing content of $\mathfrak{h}$ to always be present.

Preventing Loss of Heap Content

Problem of considering teams which do not represent extensions of the original heap $\mathfrak{h}$.
$\Rightarrow$ Force all assignments representing content of $\mathfrak{h}$ to always be present.

- Connect assignments s.t. one can force the presence of all assignments in a subteam.

Preventing Loss of Heap Content

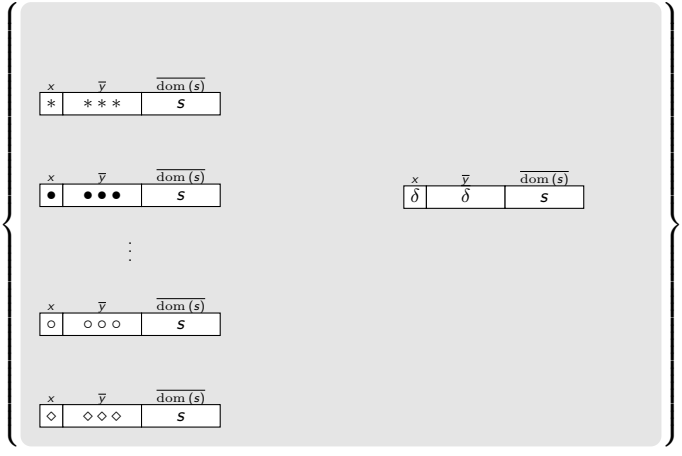
Problem of considering teams which do not represent extensions of the original heap $\mathfrak{h}$.
$\Rightarrow$ Force all assignments representing content of $\mathfrak{h}$ to always be present.

- Connect assignments s.t. one can force the presence of all assignments in a subteam.

$\rightsquigarrow$ Cyclic permutation of heap domain.

Equiextension: $\mathfrak{A} \models_X \overline{x} \bowtie \overline{y} :\Longleftrightarrow X(\overline{x}) = X(\overline{y})$

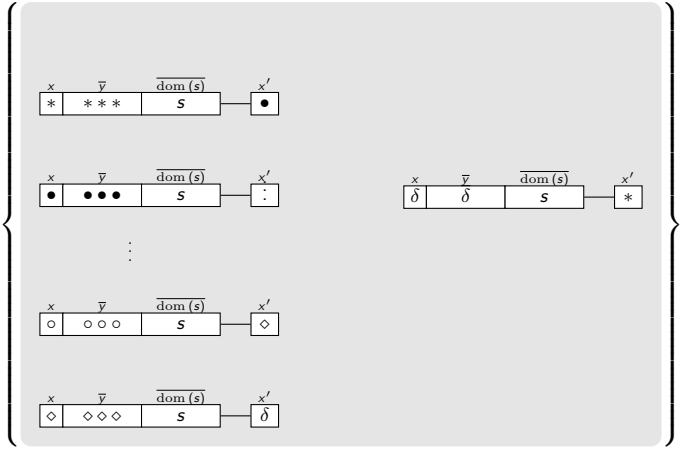
Connecting Assignments in a Cycle



$$\mathsf{cycle}(x, x') := x \bowtie x' \wedge \mathsf{dep}(x; x') \wedge \mathsf{dep}(x'; x) \wedge [(x \bowtie x' \wedge \mathtt{NE}) \to ((x = \delta \wedge \mathtt{NE}) \vee x \neq \delta)]$$

Subteams $Y \subseteq X$ with $\mathfrak{A} \models_Y \mathtt{NE} \wedge x \bowtie x'$

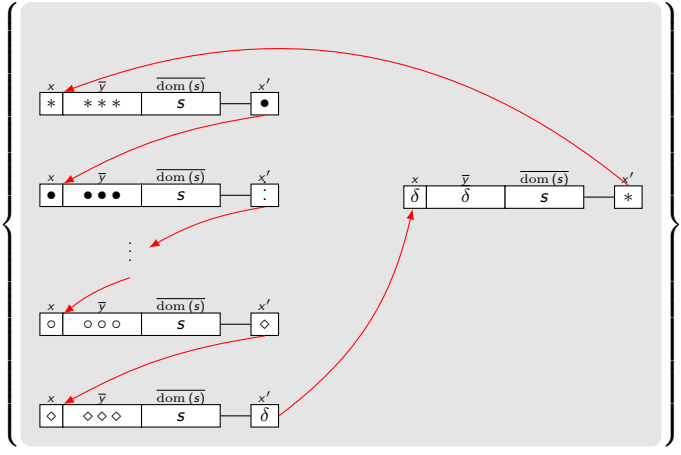
Connecting Assignments in a Cycle



$$\mathsf{cycle}(x, x') := x \bowtie x' \wedge \mathsf{dep}(x; x') \wedge \mathsf{dep}(x'; x) \wedge [(x \bowtie x' \wedge \mathtt{NE}) \to ((x = \delta \wedge \mathtt{NE}) \vee x \neq \delta)]$$

Subteams $Y \subseteq X$ with $\mathfrak{A} \models_Y \mathtt{NE} \wedge x \bowtie x'$

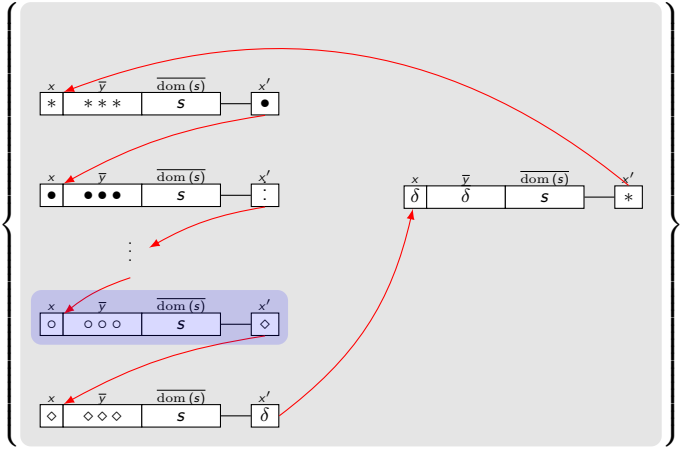
Connecting Assignments in a Cycle



$$\mathsf{cycle}(x, x') := x \bowtie x' \wedge \mathsf{dep}(x; x') \wedge \mathsf{dep}(x'; x) \wedge [(x \bowtie x' \wedge \mathtt{NE}) \to ((x = \delta \wedge \mathtt{NE}) \vee x \neq \delta)]$$

Subteams $Y \subseteq X$ with $\mathfrak{A} \models_Y \mathtt{NE} \wedge x \bowtie x'$

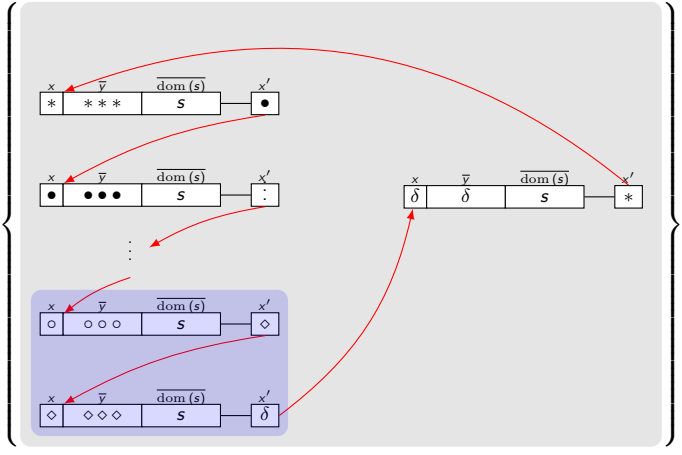
Connecting Assignments in a Cycle



$$\mathsf{cycle}(x, x') := x \bowtie x' \wedge \mathsf{dep}(x; x') \wedge \mathsf{dep}(x'; x) \wedge [(x \bowtie x' \wedge \mathtt{NE}) \rightarrow ((x = \delta \wedge \mathtt{NE}) \vee x \neq \delta)]$$

Subteams $Y \subseteq X$ with $\mathfrak{A} \models_Y \mathtt{NE} \wedge x \bowtie x'$

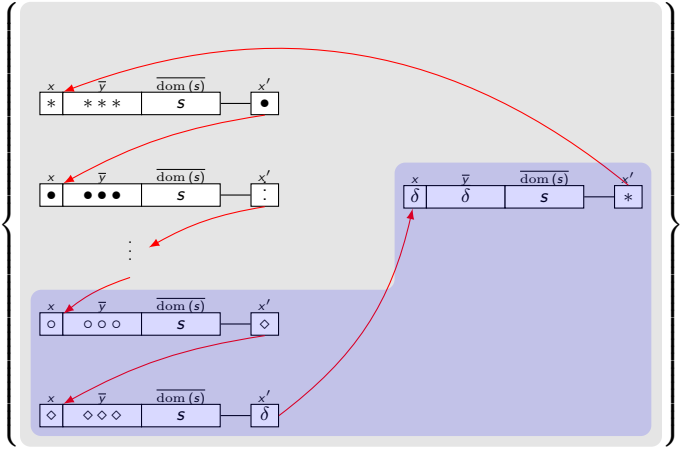
Connecting Assignments in a Cycle



$$\mathsf{cycle}(x, x') := x \bowtie x' \wedge \mathsf{dep}(x; x') \wedge \mathsf{dep}(x'; x) \wedge [(x \bowtie x' \wedge \mathtt{NE}) \to ((x = \delta \wedge \mathtt{NE}) \vee x \neq \delta)]$$

Subteams $Y \subseteq X$ with $\mathfrak{A} \models_Y \mathtt{NE} \wedge x \bowtie x'$

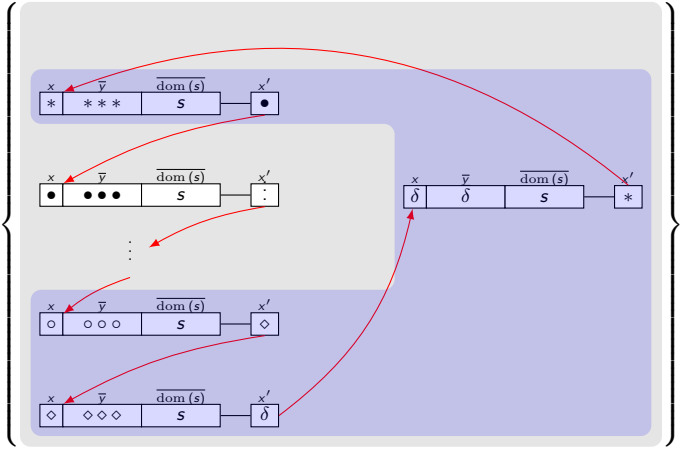
Connecting Assignments in a Cycle



$$\mathsf{cycle}(x, x') := x \bowtie x' \wedge \mathsf{dep}(x; x') \wedge \mathsf{dep}(x'; x) \wedge [(x \bowtie x' \wedge \mathtt{NE}) \to ((x = \delta \wedge \mathtt{NE}) \vee x \neq \delta)]$$

Subteams $Y \subseteq X$ with $\mathfrak{A} \models_Y \mathtt{NE} \wedge x \bowtie x'$

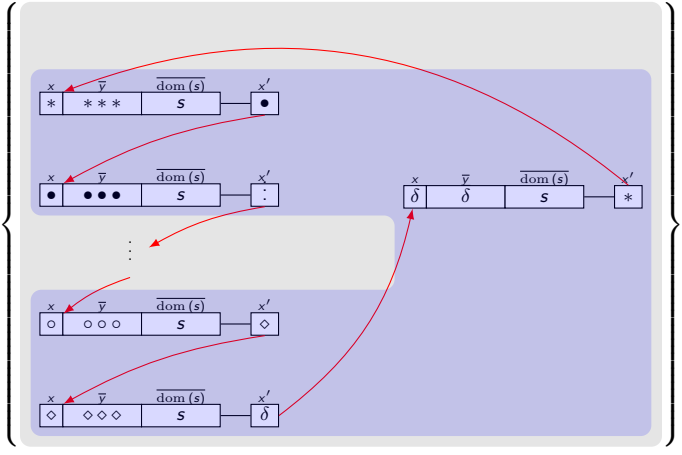
Connecting Assignments in a Cycle



$$\mathsf{cycle}(x, x') := x \bowtie x' \wedge \mathsf{dep}(x; x') \wedge \mathsf{dep}(x'; x) \wedge [(x \bowtie x' \wedge \mathtt{NE}) \to ((x = \delta \wedge \mathtt{NE}) \vee x \neq \delta)]$$

Subteams $Y \subseteq X$ with $\mathfrak{A} \models_Y \mathtt{NE} \wedge x \bowtie x'$

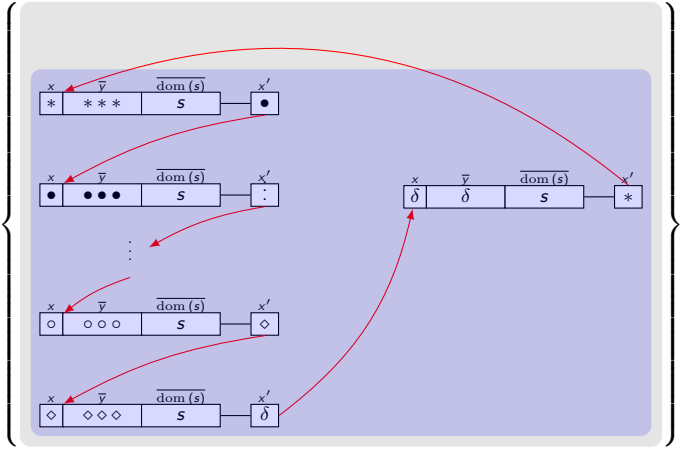
Connecting Assignments in a Cycle



$$\mathsf{cycle}(x, x') \coloneqq x \bowtie x' \wedge \mathsf{dep}(x; x') \wedge \mathsf{dep}(x'; x) \wedge [(x \bowtie x' \wedge \mathtt{NE}) \to ((x = \delta \wedge \mathtt{NE}) \vee x \neq \delta)]$$

Subteams $Y \subseteq X$ with $\mathfrak{A} \models_Y \mathtt{NE} \wedge x \bowtie x'$

Connecting Assignments in a Cycle



$$\mathsf{cycle}(x, x') := x \bowtie x' \wedge \mathsf{dep}(x; x') \wedge \mathsf{dep}(x'; x) \wedge [(x \bowtie x' \wedge \mathtt{NE}) \to ((x = \delta \wedge \mathtt{NE}) \vee x \neq \delta)]$$

Subteams $Y \subseteq X$ with $\mathfrak{A} \models_Y \mathtt{NE} \wedge x \bowtie x'$

Magic Wand and Separating Disjunction

Magic Wand:

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \mathrel{-\!\!*} \psi :\Longleftrightarrow \text{for every disjoint heap } \mathfrak{h}_\epsilon \text{ with } \mathfrak{A}, \mathfrak{h}_\epsilon \models_s \varphi, \text{ also } \mathfrak{A}, (\mathfrak{h} \dot\cup \mathfrak{h}_\epsilon) \models_s \psi.$$

$$(\varphi \mathrel{-\!\!*} \psi)^* := \exists x' [\mathsf{cycle}(x, x') \wedge \forall u \forall \overline{v} ((x \bowtie x' \wedge \mathtt{NE} \wedge \varphi_{\mathsf{ext}}(x, \overline{y}, u, \overline{v})) \rightarrow \psi^*[u, \overline{v}])]$$

Separating Disjunction:

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \circ \psi :\Longleftrightarrow \text{for all splits } (\mathfrak{h}_1, \mathfrak{h}_2) \text{ of } \mathfrak{h} \colon \mathfrak{A}, \mathfrak{h}_1 \models_s \varphi \text{ or } \mathfrak{A}, \mathfrak{h}_2 \models_s \psi.$$

$$\begin{aligned}(\varphi \circ \psi)^* := \exists x' [\mathsf{cycle}(x, x') \wedge \forall c ((x \bowtie x' \wedge \mathtt{NE} \wedge \mathsf{split}(x, c)) \rightarrow \\ (((c = \delta \wedge \varphi^*) \vee c \neq \delta) \sqcup ((c \neq \delta \wedge \psi^*) \vee c = \delta)))]\end{aligned}$$

Section 5

Conclusion

Conclusion

- Similarities of concepts in Separation Logic and Team Semantics' operations prompt search for compositional translation.
- Translation uses $\mathrm{dep}(\cdot;\cdot), \mathtt{Fin}(\cdot), \mathtt{NE}, \sqcup, \exists^1, \forall^1, \curlyvee, \bowtie, \to$ to represent Separation Logic operations.
- Intuitionistic implication $\to$ and equiextension $\bowtie$ only needed to represent magic wand and separating disjunction.

$\Rightarrow$ TLfSL as the extension of Dependence Logic by $\mathtt{NE}, \sqcup, \forall^1, \to$.

Given $\varphi \in$ SL (in negation normal form) there is a compositional translation to a formula $\varphi^* \in$ TLfSL s.t.

$$\mathfrak{A}, \mathfrak{h} \models_s \varphi \iff \mathfrak{A}^\delta \models_{X_{\mathfrak{h},s}} \varphi^*$$