Refinement in a Separation Context

Ivana Mijajlović
Queen Mary,
University of London

Noah Torp-Smith
IT University of Copenhagen

SPACE 2004

Introduction

- Hoare did data refinement for imperative programs

Introduction

- Hoare did data refinement for imperative programs

- Pointers + Data abstraction = Trouble

Introduction

- Hoare did data refinement for imperative programs
- Pointers + Data abstraction = Trouble
- As usual dangling pointers are the problem

Introduction

- Hoare did data refinement for imperative programs
- Pointers + Data abstraction = Trouble
- As usual dangling pointers are the problem
- Linguistic approaches haven't worked

Modeling Clients and Modules

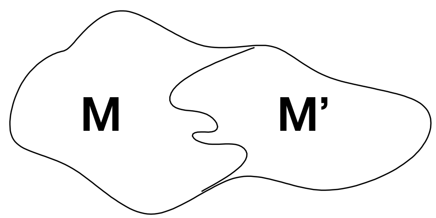
A relation $M \subseteq S \times H$ is **precise** if for any state s, h there is at most one subheap $h_0 \sqsubseteq h$, such that $(s, h_0) \in M$.

Modeling Clients and Modules

A relation $M \subseteq S \times H$ is **precise** if for any state s, h there is at most one subheap $h_0 \sqsubseteq h$, such that $(s, h_0) \in M$.

The **separating conjuction of unary relations** $M, M' \subseteq S \times H$

$$M{*}M' = \{(s,h) \mid \exists h_0, h_1.\ h_0 \# h_1 \wedge h = h_0{*}h_1 \wedge (s,h_0) \in M \wedge (s,h_1) \in M'\}.$$

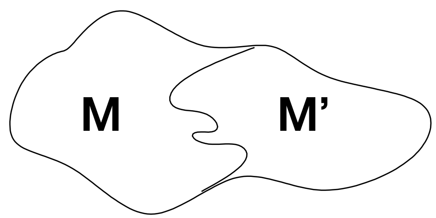


Modeling Clients and Modules

A relation $M \subseteq S \times H$ is **precise** if for any state s, h there is at most one subheap $h_0 \sqsubseteq h$, such that $(s, h_0) \in M$.

The **separating conjuction of unary relations** $M, M' \subseteq S \times H$

$M{*}M' = \{(s,h) \mid \exists h_0, h_1.\ h_0 \# h_1 \wedge h = h_0 {*} h_1 \wedge (s, h_0) \in M \wedge (s, h_1) \in M'\}.$



Let $t \subseteq (S \times H) \times (S \times H) \uplus \{wrong\}$. The relation $M \subseteq S \times H$ is **preserved** by relation t if for all $(s,h),(s',h')$, $(s,h) \in M$ and $(s,h)[t](s',h')$, imply $(s',h') \in M$.

Separation Context

$$c_{user} ::= \ \mathbf{oper}_i,\ i \in I \mid \mathbf{skip} \mid x := e \mid x := [e] \mid [e] := e \mid c_1; c_2$$
$$\mid \mathbf{if}\ e\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2 \mid \mathbf{while}\ e\ \mathbf{do}\ c$$

Separation Context

$$c_{user} ::= \ \mathbf{oper}_i, \ i \in I \mid \mathbf{skip} \mid x := e \mid x := [e] \mid [e] := e \mid c_1; c_2$$
$$\mid \mathbf{if} \ e \ \mathbf{then} \ c_1 \ \mathbf{else} \ c_2 \mid \mathbf{while} \ e \ \mathbf{do} \ c$$

Let $M \subseteq S \times H$ be a precise unary relation, and for $i \in I$ let $oper_i$ preserve relation $M * \mathsf{T}$. A program c is a **unary separation context** for M and $(oper_i)_{i \in I}$ if for all executions and all $(s, h) \in M * \mathsf{T}$ $c, s, h \not\rightsquigarrow av$ and $c, s, h \not\rightsquigarrow wrong$.

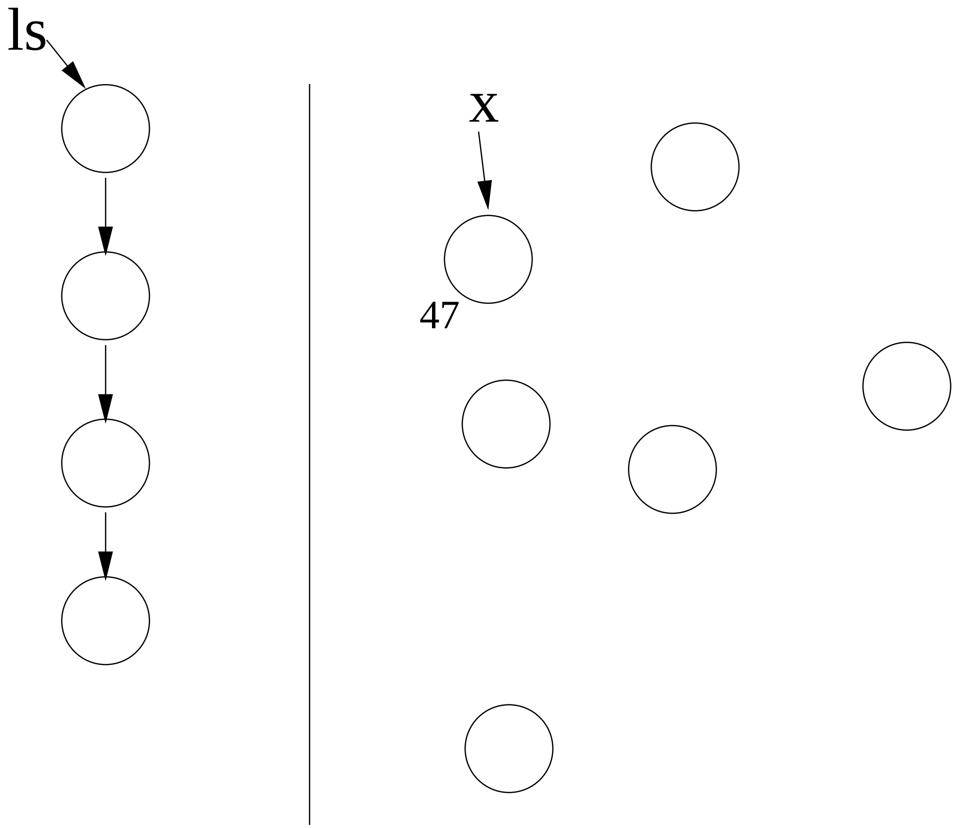
Separation Context

$$c_{user} ::= \mathbf{oper}_i,\ i \in I \mid \mathbf{skip} \mid x := e \mid x := [e] \mid [e] := e \mid c_1; c_2$$
$$\mid \mathbf{if}\ e\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2 \mid \mathbf{while}\ e\ \mathbf{do}\ c$$

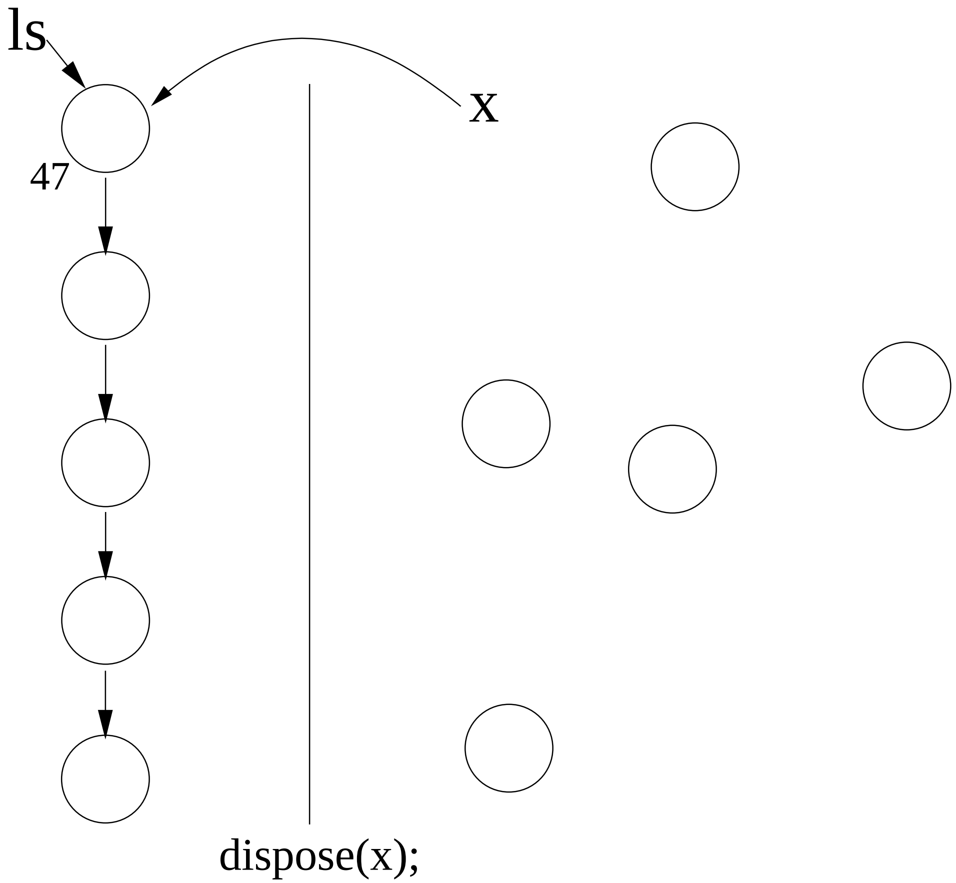
Let $M \subseteq S \times H$ be a precise unary relation, and for $i \in I$ let $oper_i$ preserve relation $M * \mathsf{T}$. A program c is a **unary separation context** for M and $(oper_i)_{i \in I}$ if for all executions and all $(s, h) \in M * \mathsf{T}$ $c, s, h \not\rightsquigarrow av$ and $c, s, h \not\rightsquigarrow wrong$.

Let $M \subseteq S \times H$ be a precise relation, and for $(i \in I)$ let $oper_i$ preserve $M * \mathsf{T}$, and let c be a separation context for M and $(oper_i)_{i \in I}$. If $(s, h) \in M * \mathsf{T}$, and $c, s, h \rightsquigarrow s', h'$, then $(s', h') \in M * \mathsf{T}$.

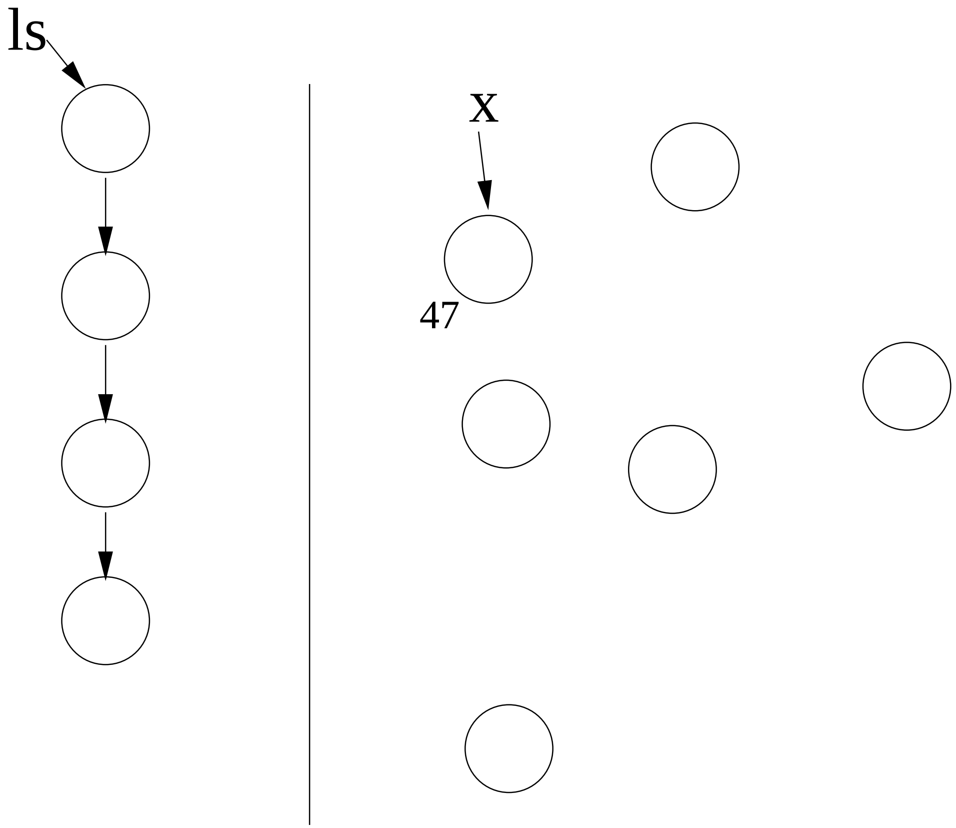
Separation context



Separation context

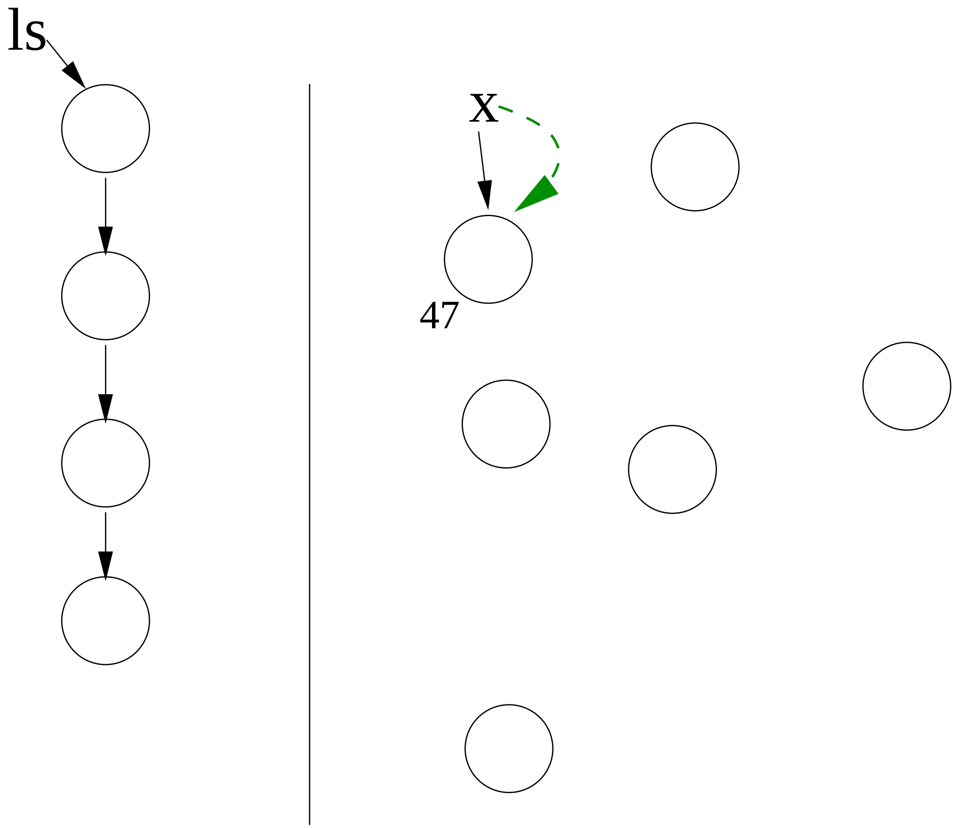


Separation context



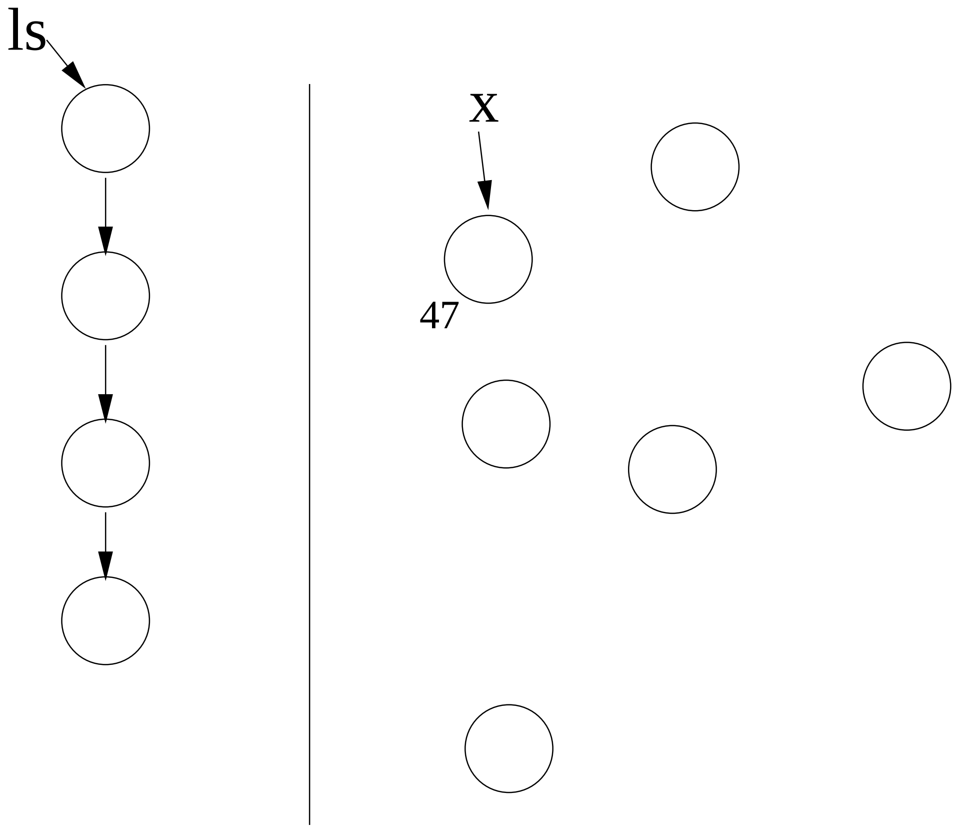
dispose(x); x=new();

Separation context

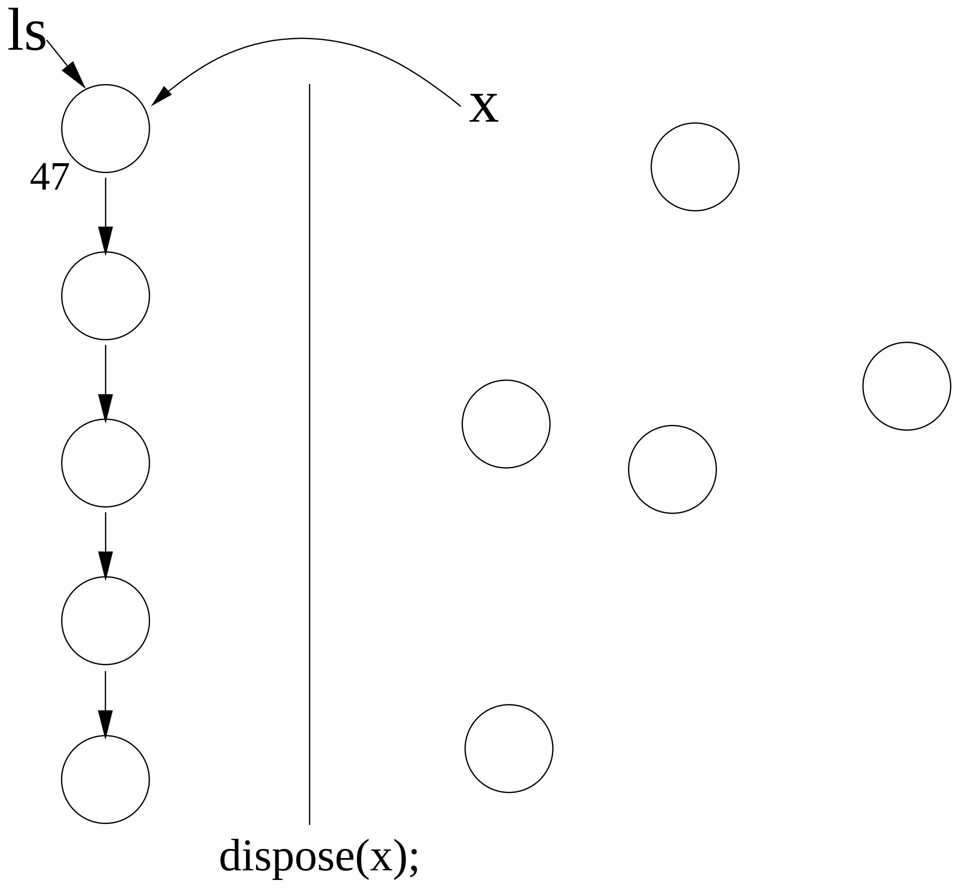


dispose(x); x=new(); y=[x]

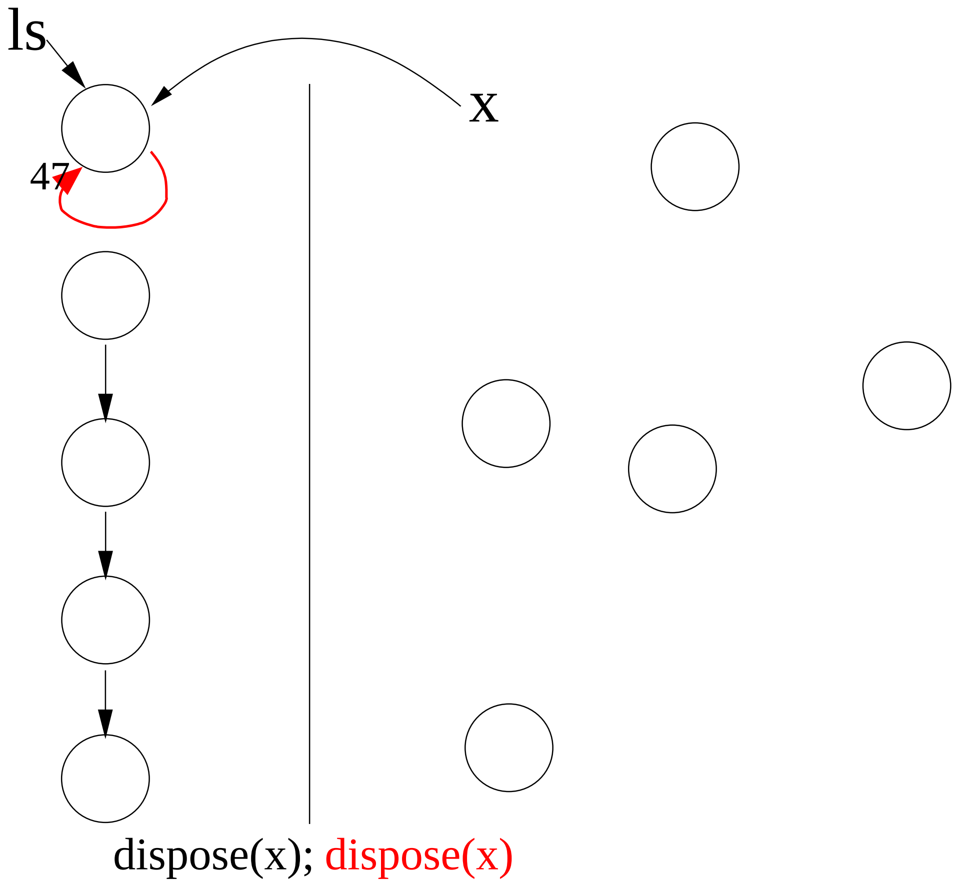
Non-separation context



Non-separation context



Non-separation context



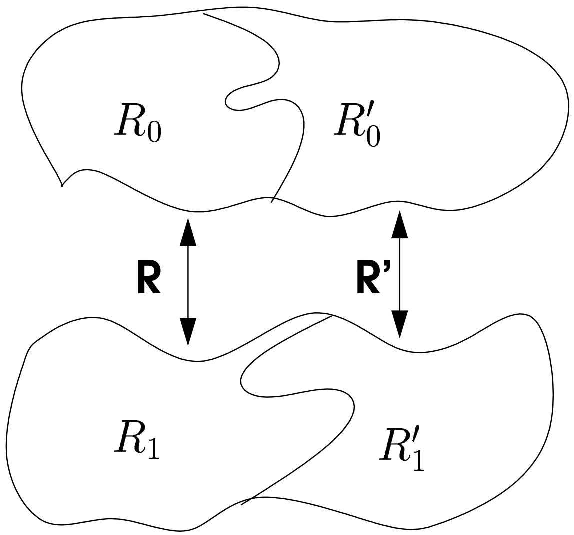
Binary Relations for Refinement

We say that binary relation R is **precise**, if each of its two projections on the set of states is precise.

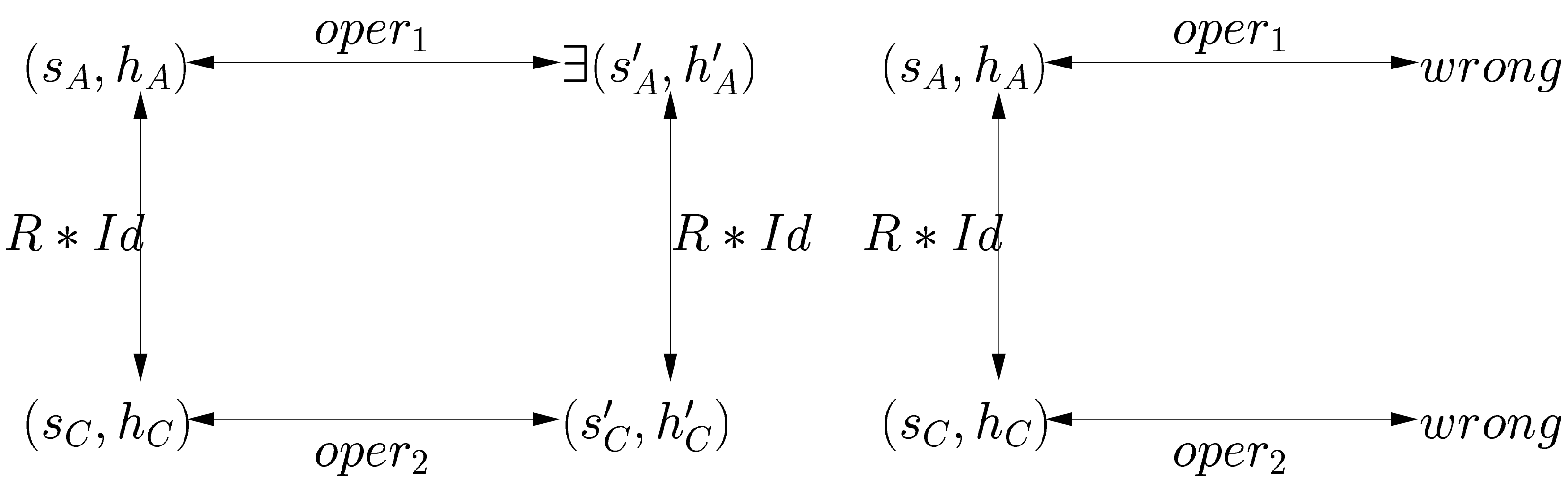
Binary Relations for Refinement

We say that binary relation R is **precise**, if each of its two projections on the set of states is precise.

Separating conjunction of binary relations



Refinement



$oper_1$
(s_A, h_A)
$\exists(s'_A, h'_A)$
$R * Id$
$R * Id$
(s_C, h_C)
(s'_C, h'_C)
$oper_2$
$oper_1$
(s_A, h_A)
$wrong$
$R * Id$
(s_C, h_C)
$wrong$
$oper_2$

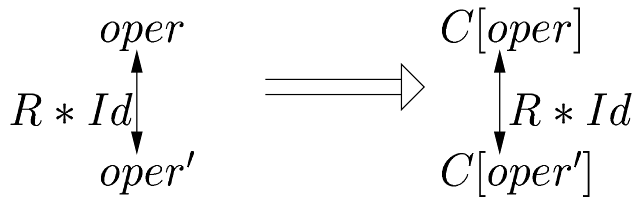
The Result

- A separation context for the abstract data type is a separation context for all its refinements

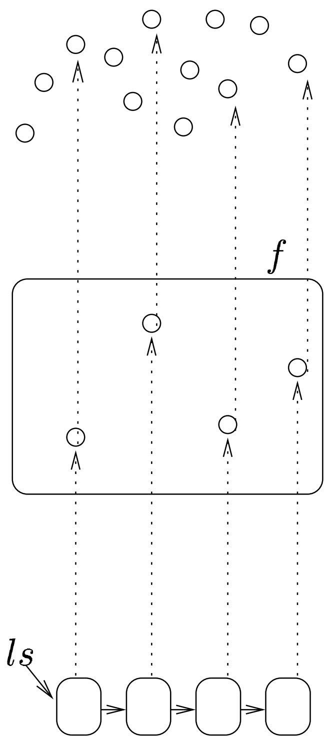
The Result

- A separation context for the abstract data type is a separation context for all its refinements

- Sepration contexts preserve $R * \mathsf{Id}$



Example - $new()$ and $dispose()$



Abstract – Magic

RI: emp

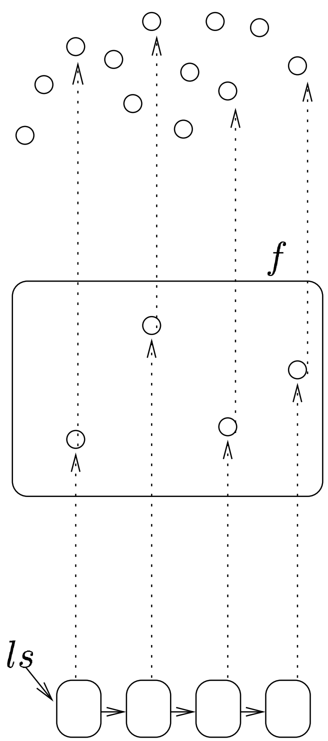
Intermediate – Set

RI: $\forall_* p \in f. p \mapsto \_, \_$

Concrete – List

RI: $list(\alpha, ls.nil)$

Example - $new()$ and $dispose()$



Abstract – Magic

RI: emp

Intermediate – Set

RI: $\forall_* p \in f.p \mapsto \_, \_$

Concrete – List

RI: $list(\alpha, ls.nil)$

$$R_1 = \{((s_A, h_A), (s_C, h_C)) \mid s_A, h_A \Vdash emp \wedge (s_C, h_C \Vdash \forall_* p \in f.\ p \mapsto \_, \_)\}$$

Future Work

- This is only a model

Future Work

- This is only a model
- We would like to have a logic