분리논리 자동정리증명기 개발
(A Theorem Prover for Separation Logic)

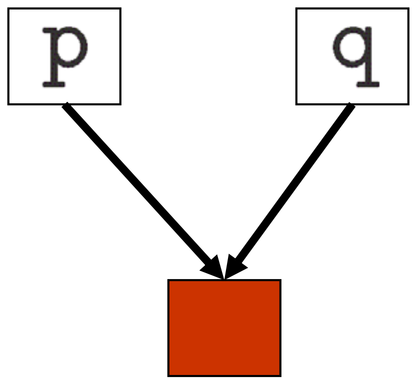
박성우
POSTECH

제9회 소프트웨어무결점연구센터 워크샵
2013년 1월 31일

Shared Mutable Data Structures

- An updatable field can be referenced from more than one point.

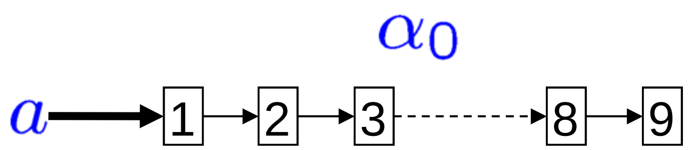
```
*p := 0;
*q := 1;
if (*p = 0)
    return true;
else
    return false;
```

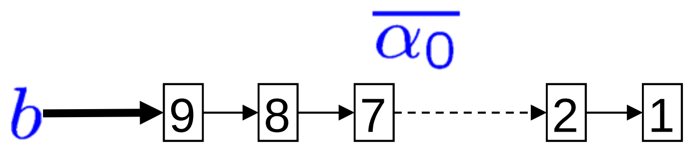


- Hoare logic fails to scale to programs of even moderate size.

Specification of In-place List Reversal

```
@pre:    list α0 a
b := nil;
while (a != nil) do
   k := [a + 1];
   [a + 1] := b;
   b := a;
   a := k;
end while;
@post:  list ᾱ0 b
```





Loop Invariant in Hoare Logic

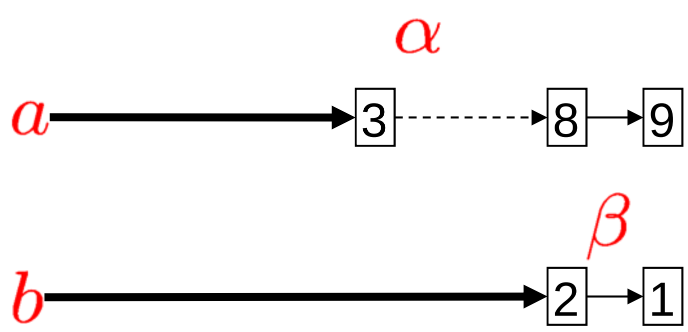
```
@pre    list α0 a
b := nil;
@L: ∃α, β. list α a ∧ list β b ∧ rev(α0) = rev(α) · β
while (a != nil) do
   k := [a + 1];
   [a + 1] := b;
   b := a;
   a := k;
end while;
@post  list rev(α0) b
```

$@pre \quad \mathsf{list}\ \alpha_0\ a$

$b := \mathtt{nil};$

$@L{:}\ \exists \alpha, \beta.\ \mathsf{list}\ \alpha\ a \ \wedge\ \mathsf{list}\ \beta\ b \ \wedge\ \overline{\alpha_0} = \overline{\alpha} \cdot \beta$

$@post \quad \mathsf{list}\ \overline{\alpha_0}\ b$



- However, lists a and b must be disjoint.

Complex Loop Invariant in Hoare Logic

```
@pre    list α₀ a
b := nil;
@L: ∃α, β. list α a ∧ list β b ∧ ‾α₀ = ‾R · β
    ∧ (∀k. reachable(a, k) ∧ reachable(b, k) ⊃ k = nil)
while (a != nil) do
  k := [a + 1];
  [a + 1] := b;
  b := a;
  a := k;
end while;
@post  list ‾α₀ b
```

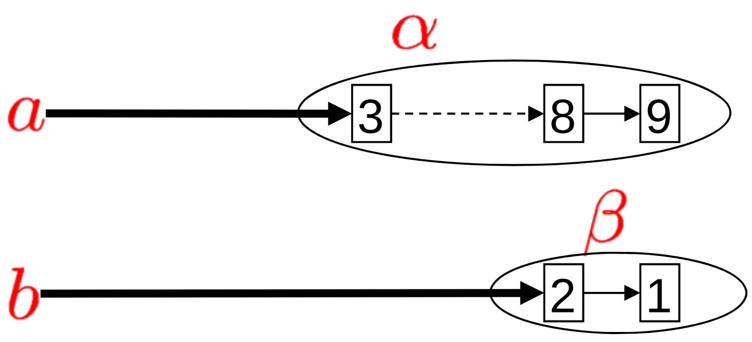
$$\texttt{@L: } \exists \alpha, \beta.\ \mathsf{list}\ \alpha\ a\ \wedge\ \mathsf{list}\ \beta\ b\ \wedge\ \overline{\alpha_0} = \overline{R} \cdot \beta \ \wedge\ (\forall k.\ \mathsf{reachable}(a, k) \wedge \mathsf{reachable}(b, k) \supset k = \mathsf{nil})$$

$$\texttt{@post}\ \ \mathsf{list}\ \overline{\alpha_0}\ b$$

Lists a and b are disjoint.

Loop Invariant in Separation Logic

```
@pre    list α₀ a
b := nil;
@L: ∃α, β. list α a ⋆ list β b ∧ ᾱ₀ = ᾱ · β
while (a != nil) do
  k := [a + 1];
  [a + 1] := b;
  b := a;
  a := k;
end while;
@post  list ᾱ₀ b
```



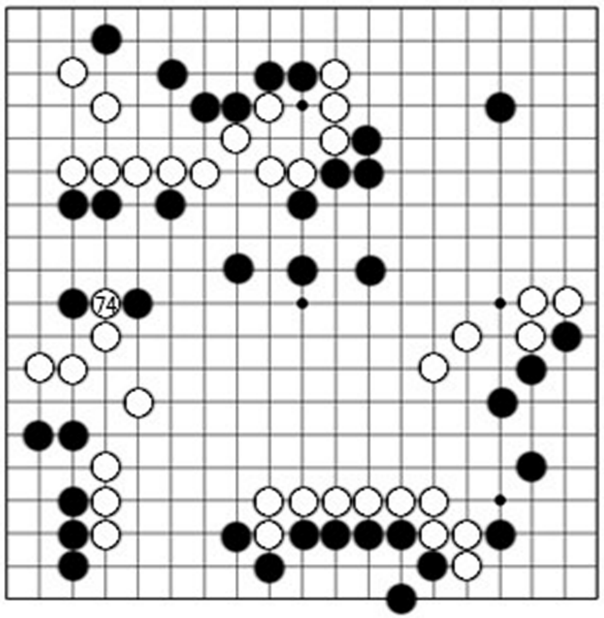
- $\star$ = separating conjunction
  - – describes two disjoint memory portions

Frame Rule in Separation Logic

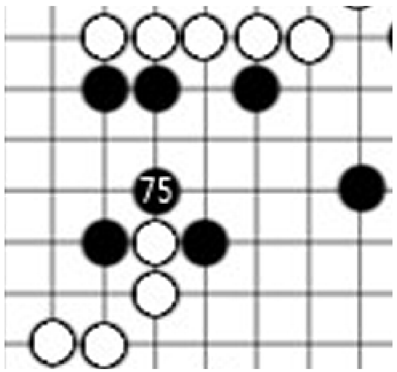
- Supports local reasoning

$$\frac{\{A\}\ \text{Program}\ \{B\}}{\{A \star C\}\ \text{Program}\ \{B \star C\}}$$

where Program does not access variables in C



제1회 응씨배 결승5번기 제5국
백 九단 녜 웨이핑(중국) / 흑 九단 조 훈 현(한국)



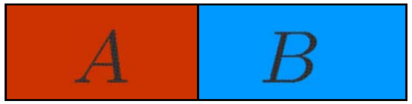
Logical Connectives in Separation Logic

- Separating conjunction

  $$A \star B$$

  – The current heap can be partitioned into two separate heaps;

  – A holds for one, and B holds for the other.



- Separating implication

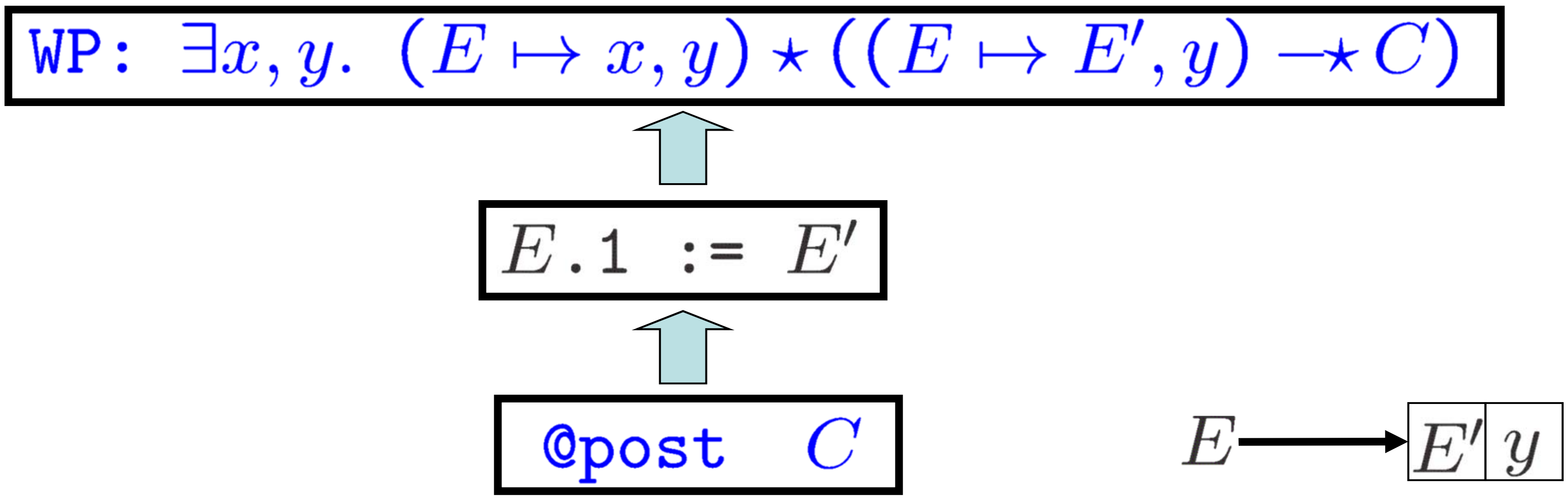
  $$A \mathbin{-\!\!\star} B$$

  – If the current heap is extended with a separate heap for which A holds,

  – then B holds for the combined heap.

Separating Implication $A \mathrel{-\!\!\star} B$

- Essential to building a complete verification system
  - with backward reasoning by weakest precondition generation



- No existing verification tools fully support $A \mathrel{-\!\!\star} B$.
  - Smallfoot, Space Invader, THOR, SLAyer, HIP, VeriFast, jStar, Xisa, ...

Goal

- Build a theorem prover for full separation logic
  - with separating conjunction $\star$
  - **also with separating implication** $-\!\!\star$

> This incompleteness could be dealt with if we instead used the backwards-running weakest preconditions of Separation Logic [4]. Unfortunately, there is no existing automatic theorem prover which can deal with the form of these assertions (which use quantification and the separating implication $-\!\!*$). If there were such a prover, we would be eager consumers of it.

Symbolic Execution with Separation Logic.
Josh Berdine and Cristiano Calcagno and Peter O'Hearn. APLAS'05.

- Schorr-Waite Algorithm의 기계적 검증

Contents

- Introduction *V*
- **Theorem prover for Boolean BI**
- Theorem prover for separation logic

Building a Theorem Prover for Boolean BI

- Boolean BI (Bunched Implications)
  - underlying theory of separation logic
  - classical logic extended with $\star$ and $-\!\!\star$

$$A \;::=\; P \mid \top \mid \neg A \mid A \wedge B \mid \mathrm{I} \mid A \star B \mid A -\!\!\star B$$

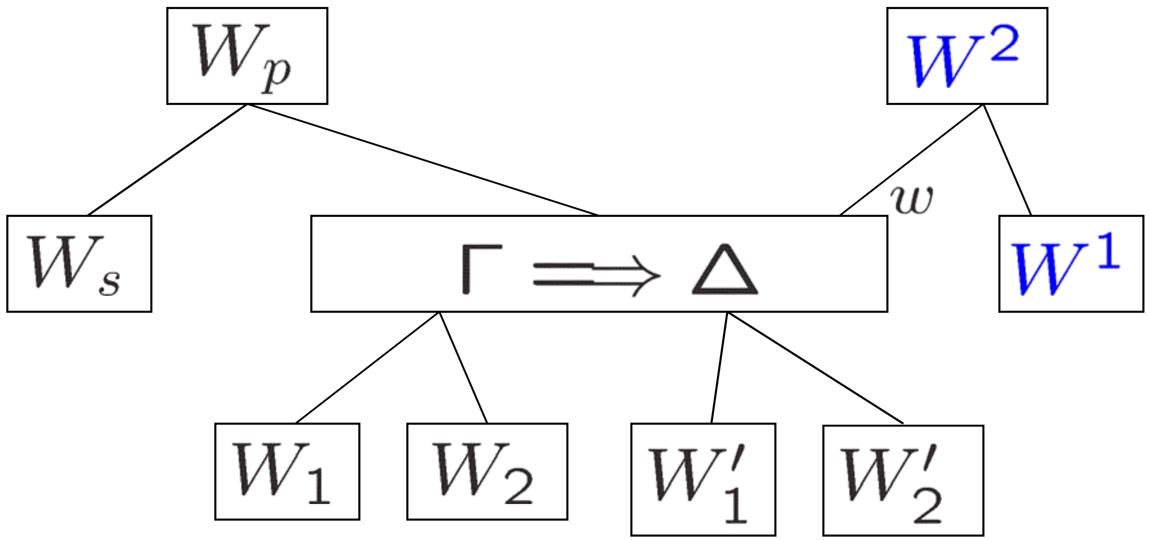
- S_{BBI}
  - nested sequent calculus for Boolean BI

Nested Sequents with **Graph** Structures

- Classical logic + $A \star B$ + $A \mathrel{-\!\!\star} B$

formula	A	$::=$	$P \mid \bot \mid \neg A \mid A \vee B \mid A \star B$
truth ctx.	Γ	$::=$	$\cdot \mid \Gamma; S$
false. ctx.	Δ	$::=$	$\cdot \mid \Delta; A$
node state	S	$::=$	$A \mid \emptyset_{\mathrm{m}} \mid W_1, W_2 \mid W^1\langle\langle W^2 \rangle\rangle$
sequent	W	$=$	$\Gamma \Longrightarrow \Delta$

$W^1\langle\langle W^2 \rangle\rangle$: a sibling sequent W^1 and a common parent sequent W^2



$\mathsf{S}_{\mathsf{BBI}}$: Nested Sequent Calculus for Boolean BI

Structural rules:

$$\frac{\Gamma \Longrightarrow \Delta}{\Gamma; S \Longrightarrow \Delta}\, WL_{\mathcal{S}} \quad \frac{\Gamma \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta; A}\, WR_{\mathcal{S}} \quad \frac{\Gamma; S; S \Longrightarrow \Delta}{\Gamma; S \Longrightarrow \Delta}\, CL_{\mathcal{S}} \quad \frac{\Gamma \Longrightarrow \Delta; A; A}{\Gamma \Longrightarrow \Delta; A}\, CR_{\mathcal{S}} \quad \frac{\Gamma; W', W \Longrightarrow \Delta}{\Gamma; W, W' \Longrightarrow \Delta}\, EC_{\mathcal{S}}$$

$$\frac{\Gamma; W_1, (W_2, W_3 \Longrightarrow \cdot) \Longrightarrow \Delta}{\Gamma; (W_1, W_2 \Longrightarrow \cdot), W_3 \Longrightarrow \Delta}\, EA_{\mathcal{S}} \quad \frac{\Gamma_1; (\Gamma_2 \Longrightarrow \Delta_2), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \Delta_1}{\Gamma_1; \Gamma_2 \Longrightarrow \Delta_1; \Delta_2}\, \emptyset_{\mathsf{m}} U_{\mathcal{S}} \quad \frac{\Gamma_1; \Gamma_2 \Longrightarrow \Delta_1; \Delta_2}{\Gamma_1; (\Gamma_2 \Longrightarrow \Delta_2), (\emptyset_{\mathsf{m}} \Longrightarrow \cdot) \Longrightarrow \Delta_1}\, \emptyset_{\mathsf{m}} D_{\mathcal{S}}$$

Traverse rules:

$$\frac{\Gamma_{\mathsf{c1}}; (\Gamma_{\mathsf{c2}} \Longrightarrow \Delta_{\mathsf{c2}})\langle \Gamma \Longrightarrow \Delta \rangle \Longrightarrow \Delta_{\mathsf{c1}}}{\Gamma; (\Gamma_{\mathsf{c1}} \Longrightarrow \Delta_{\mathsf{c1}}), (\Gamma_{\mathsf{c2}} \Longrightarrow \Delta_{\mathsf{c2}}) \Longrightarrow \Delta}\, TC_{\mathcal{S}} \quad \frac{\Gamma_{\mathsf{p}}; (\Gamma \Longrightarrow \Delta), (\Gamma_{\mathsf{s}} \Longrightarrow \Delta_{\mathsf{s}}) \Longrightarrow \Delta_{\mathsf{p}}}{\Gamma; (\Gamma_{\mathsf{s}} \Longrightarrow \Delta_{\mathsf{s}})\langle \Gamma_{\mathsf{p}} \Longrightarrow \Delta_{\mathsf{p}} \rangle \Longrightarrow \Delta}\, TP_{\mathcal{S}}$$

Logical rules:

$$\frac{}{A \Longrightarrow A}\, Init_{\mathcal{S}} \quad \frac{}{\bot \Longrightarrow \cdot}\, \bot L_{\mathcal{S}} \quad \frac{\Gamma \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta; \bot}\, \bot R_{\mathcal{S}} \quad \frac{\Gamma \Longrightarrow \Delta; A}{\Gamma; \neg A \Longrightarrow \Delta}\, \neg L_{\mathcal{S}} \quad \frac{\Gamma; A \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta; \neg A}\, \neg R_{\mathcal{S}} \quad \frac{\Gamma_1; A \Longrightarrow \Delta_1 \quad \Gamma_2; B \Longrightarrow \Delta_2}{\Gamma_1; \Gamma_2; A \vee B \Longrightarrow \Delta_1; \Delta_2}\, \vee L_{\mathcal{S}}$$

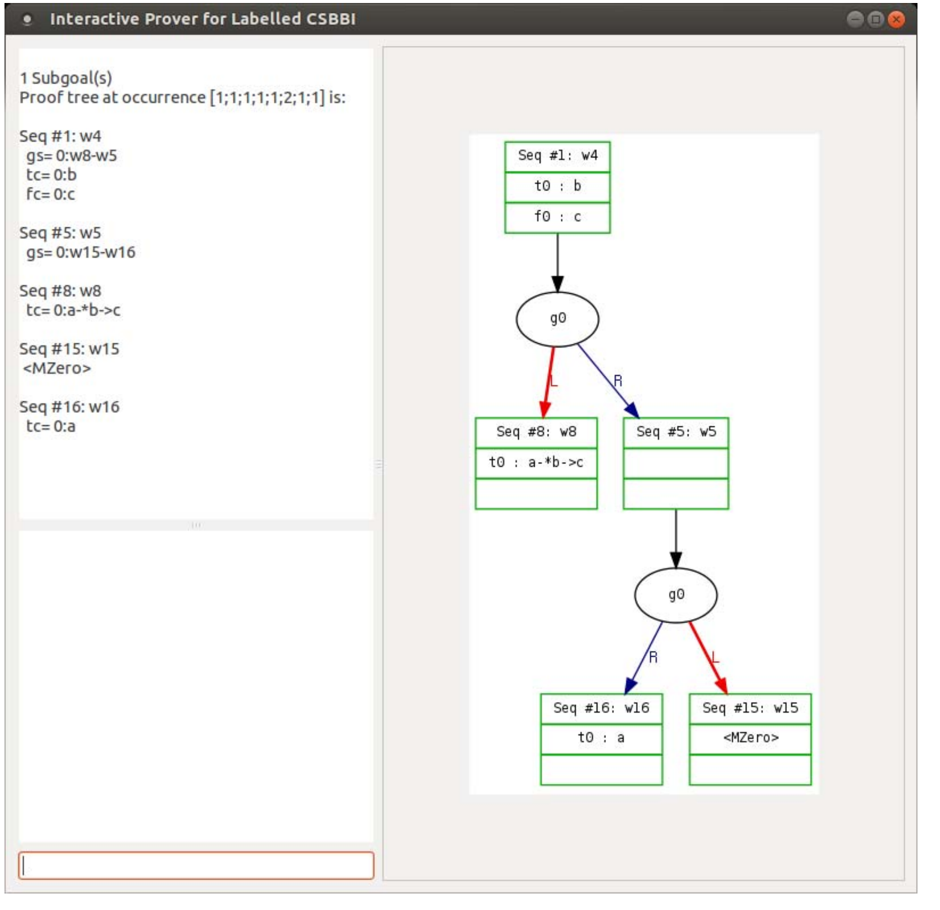
$$\frac{\Gamma \Longrightarrow \Delta; A; B}{\Gamma \Longrightarrow \Delta; A \vee B}\, \vee R_{\mathcal{S}} \quad \frac{\Gamma; \emptyset_{\mathsf{m}} \Longrightarrow \Delta}{\Gamma; \mathsf{I} \Longrightarrow \Delta}\, \mathsf{I} L_{\mathcal{S}} \quad \frac{}{\emptyset_{\mathsf{m}} \Longrightarrow \mathsf{I}}\, \mathsf{I} R_{\mathcal{S}} \quad \frac{\Gamma; (A \Longrightarrow \cdot), (B \Longrightarrow \cdot) \Longrightarrow \Delta}{\Gamma; A \star B \Longrightarrow \Delta}\, \star L_{\mathcal{S}}$$

$$\frac{\Gamma_1 \Longrightarrow \Delta_1; A \quad \Gamma_2 \Longrightarrow \Delta_2; B}{(\Gamma_1 \Longrightarrow \Delta_1), (\Gamma_2 \Longrightarrow \Delta_2) \Longrightarrow A \star B}\, \star R_{\mathcal{S}} \quad \frac{\Gamma_1 \Longrightarrow \Delta_1; A \quad \Gamma_2; B \Longrightarrow \Delta_2}{(\Gamma_1 \Longrightarrow \Delta_1)\langle \Gamma_2 \Longrightarrow \Delta_2 \rangle; A \mathrel{-\!\!\star} B \Longrightarrow \cdot}\, {-\!\!\star} L_{\mathcal{S}} \quad \frac{\Gamma; (A \Longrightarrow \cdot)\langle \cdot \Longrightarrow B \rangle \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta; A \mathrel{-\!\!\star} B}\, {-\!\!\star} R_{\mathcal{S}}$$

Theorem (Cut elimination):
If $\Gamma \Longrightarrow \Delta; C$ and $\Gamma'; C \Longrightarrow \Delta'$, then $\Gamma; \Gamma' \Longrightarrow \Delta; \Delta'$.

BBeye: A Theorem Prover for Boolean BI

- Interactive
- Supports both CUI and GUI
- Written in OCaml
- Online demo at
  http://pl.postech.ac.kr/BBI/



- Jonghyun Park, Jeongbong Seo, Sungwoo Park. *A Theorem Prover for Boolean BI.* POPL 2013.

- **<u>Now we know how to deal with $-\!\!\ast$.</u>**

Contents

- Introduction *V*
- Theorem prover for Boolean BI *V*
- **Theorem prover for separation logic**

Separation Logic

- 정의

location	$\mathsf{L}_1, \mathsf{L}_2, \mathsf{L}_3, \cdots$		
expression	E	$::=$	$\mathsf{L}_i \mid x \mid a \mid \cdots$
location expression	l	$::=$	$\mathsf{L}_i \mid x \mid a$
primitive formula	P	$::=$	$[l \mapsto E] \mid E = E \mid \cdots$
formula	A, B, C	$::=$	$P \mid \bot \mid \neg A \mid A \vee A \mid$
			$\mathsf{I} \mid A \star A \mid A \mathrel{-\!\star} A \mid \exists a.A$

- Judgment

$$(S, H) \models A$$

- 문제:

  "주어진 formula A가 모든 stack S와 모든 heap H에 대해서 참인지 판별하라"

  – quantifier 9가 없으면: decidable

  – quantifier 9가 있으면: undecidable

Proof System for Separation Logic

- 첫번째 key idea (from BBeye):
  - use a graph structure of sequents
  - label each sequent with a heap variable.
- 두번째 key idea
  - 전체 system의 completeness = primitive formula를 다루는 system의 completeness

- 현재 soundness와 completeness 증명 중
  - soundness: proof system은 옳다
  - completeness: 놓치는 경우가 없다

CCeye: A Theorem Prover for Separation Logic

- 현재 설계 중

expression relation	θ	$::=$	$E = E' \mid E \neq E'$
expression relations	Θ	$=$	$\theta_1, \cdots, \theta_n$
heap variable	w, u, v		
heap relation	σ	$::=$	$w \doteq \epsilon \mid w \dot{\neq} \epsilon \mid w \doteq [l \mapsto E] \mid$ $w \dot{\neq} [l \mapsto E] \mid w \doteq w' \circ w'' \mid w \doteq w'$
heap relations	Σ	$=$	$\sigma_1, \cdots, \sigma_n$
truth context	Γ	$::=$	$\cdot \mid \Gamma, A$
falsehood context	Δ	$::=$	$\cdot \mid \Delta, A$
heap sequent	$[\Gamma \Longrightarrow \Delta]^w$		
heap sequents	Π	$=$	$[\Gamma \Longrightarrow \Delta]^w, \cdots, [\Gamma' \Longrightarrow \Delta']^{w'}$
world description	$\Theta; \Sigma \parallel \Pi$		

- Challenge: Complexity 문제 처리

謝謝

감사합니다

gla@postech.ac.kr