분리논리 자동증명기 구현

2014년 7월 29일

제11회 소프트웨어무결점연구센터 여름 워크샵

박성우

소프트웨어 개발에서 중요한 질문

소프트웨어의

오류의 위치를 찾아주거나

오류가 없음을 확인해 주는

자동화된 도구가 있을까?

- 테스팅(testing)
- 정적 분석 (static analysis)
- 정형 검증 (formal verification)

정형 검증 (Formal Verification)

- 장단점
  - [+] 높은 신뢰도
    - 증명 (formal proof) 생성됨
  - [+] Functional correctness까지 증명할 수 있음
    - "The program correctly sorts a given array.“
  - [-] 사용하기 위해서 전문 지식이 필요함
- Hoare 논리 체계의 한계
- 분리 논리 (separation logic) 등장

분리논리 (Separation Logic)

- A Logic for Shared Mutable Data Structures [Reynolds 2002]
  - 힙(메모리 할당)을 다루는 프로그램 분석에 적합
  - supports local reasoning

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

where Program does not access variables in C

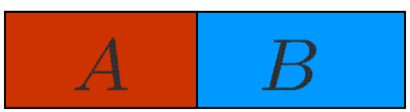
- 분리논리를 이용한 프로그램 검증도구
  - Smallfoot, Space Invader, THOR, SLAyer, HIP, VeriFast, jStar, Xisa, SeLoger, SLP, ...
  - Facebook의 Monoidics 합병
  - but focus only on separating conjunction $\star$.

분리논리 연산자

- <u>Separating conjunction</u>

$$A \star B$$

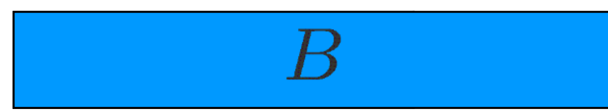
  - The current heap can be partitioned into two separate heaps;
  - A holds for one, and B holds for the other.



- <u>Separating implication</u>

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

  - If the current heap is extended with a separate heap for which A holds,
  - then B holds for the combined heap.



  - <u>**마법봉 (magic wand)**</u>

ㅁ 마력 —★

- thm

p $t)$

$\wedge\ noL$

$*\ (res$

$(x) \rightarrow$

- erat

c, xs

c, xs

- ring

$\mathcal{R}')$

	Name	Class	Level	Exp
1	Baron Noi	bar	86	1,146M
2	Baron Bugssy	bar	84	992,586K
3	Lord of-Souls	nec	84	966,645K
4	Baron Kortez	nec	83	925,825K
5	Baron WingsMage_XIX	nec	83	921,492K
6	Baron EdgE_SD	bar	83	903,047K
7	Baron Pheadrus	nec	82	844,657K
8	Baron Isenhart	pal	82	813,915K
9	Lord Xo_Martel	bar	81	771,194K
10	Lord Tribe_Necro	nec	81	753,054K
11	Baron Remesis	nec	81	752,302K
12	Baron Veran	nec	81	750,021K
13	Baron vthree	pal	81	747,571K
14	Baron Rialie-x	bar	80	738,888K
15	Lord Iustinias	bar	80	722,341K
16	Baron Nairabrab-x	bar	80	699,877K
17	Baron System_IV	bar	80	691,612K
18	Baron NecroSpirit	nec	80	687,848K
19	Baron Rolg	bar	80	685,070K
20	Baron Nagash_exe	bar	79	638,464K
21	Baron PikeMasta	bar	79	636,896K
22	Baron ChemicalMan	nec	79	634,468K
23	Baron Doomrazor	nec	78	600,729K
24	Baron Dahmy-BM	bar	78	595,520K
25	Baron jizzmilk	bar	78	582,118K
26	Baron Hadess	nec	78	580,416K
27	Baroness Mooshafu	ama	78	575,317K
28	Baron Nostradomus	pal	78	574,103K
29	Lord Russianlew	nec	78	572,485K
30	Lord Jirar	pal	77	560,089K
31	Baroness gladiusCold	sor	77	548,244K
32	Baron FallenNecro	nec	77	543,526K
33	Baron Cell-zf	bar	77	535,785K
34	Baron Robin-Hood	pal	77	525,501K
35	Baron Mosochist_Devil	nec	77	524,840K
36	Lord Slade-Rage	bar	76	512,969K
37	Baron Siphu	pal	76	511,027K
38	Baron Melkezadek	bar	76	504,379K
39	Baron holyDeAthWish	nec	76	500,757K
40	Lord Dragon-Hellfire	bar	76	497,782K
41	Baron Ziph	nec	76	491,697K
42	Baron Papa_Doc	nec	76	490,250K
43	Baron Blodia	pal	76	486,052K
44	Baron Kyle-Barb	bar	76	484,355K
45	Lord Plur	bar	76	481,164K
46	Lady Giulia	ama	75	474,213K
47	Baron Vildain	nec	75	468,353K
48	Baron Yvonkrever	bar	75	465,655K

(Screenshot taken on July 26, 2000)

왜 마법봉 연산자가 필수적인가?

- 완전한 프로그램 검증 시스템 구현에 필수
  - weakest precondition 이용한 역방향 추론에 필수

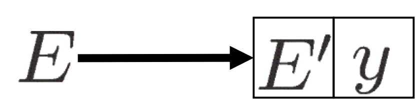
$$\texttt{WP:}\ \exists x, y.\ (E \mapsto x, y) \star ((E \mapsto E', y) \mathrel{-\!\!\star} C)$$

↑

$$E\texttt{.1 := }E'$$

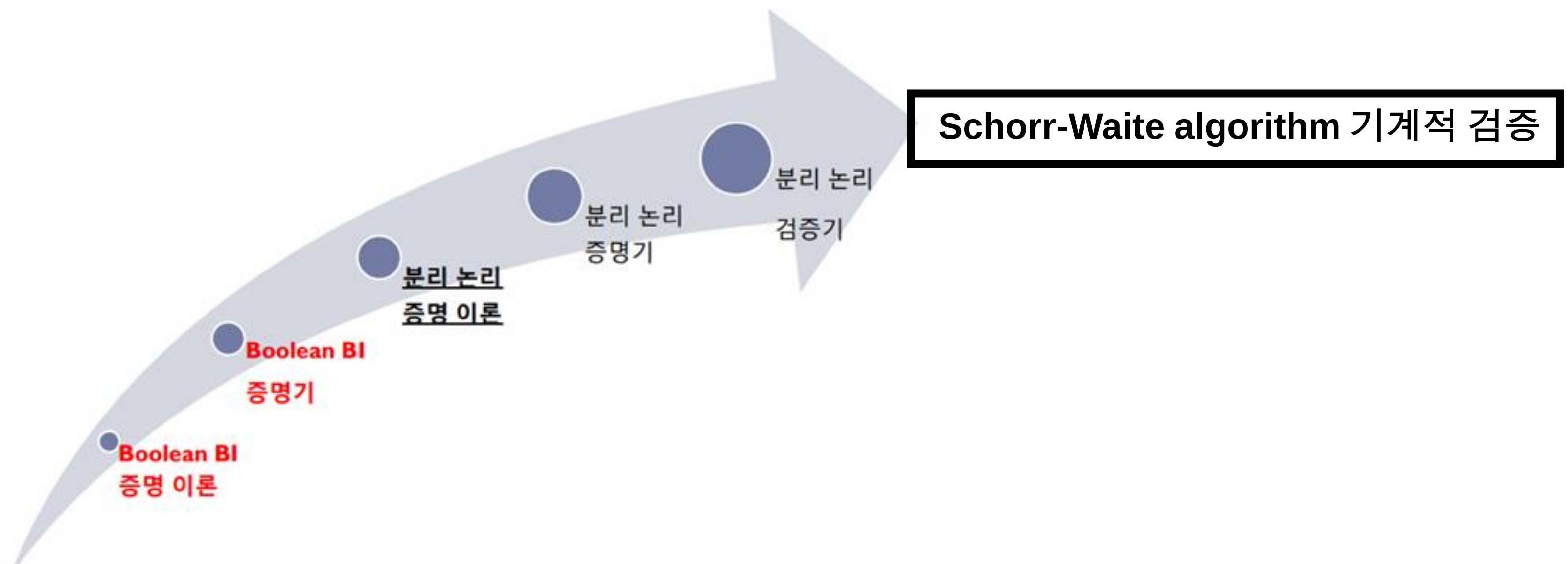
↑

$$\texttt{@post}\ \ C$$



3단계 연구 계획

- 1단계: Boolean BI 증명론 완성
  - Boolean BI 증명기 개발
- 2단계: 분리 논리 증명론 완성
- 3단계: 분리 논리 검증기 개발
  - 분리 논리 증명기 개발
- 목표 결과물: Schorr-Waite algorithm 기계적 검증



분리 논리 정의

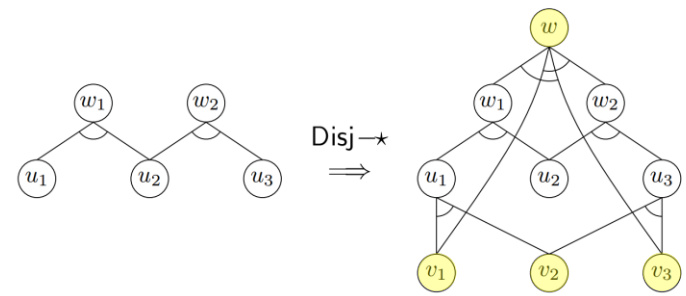
- Classical first-order logic + Intuitionistic linear logic

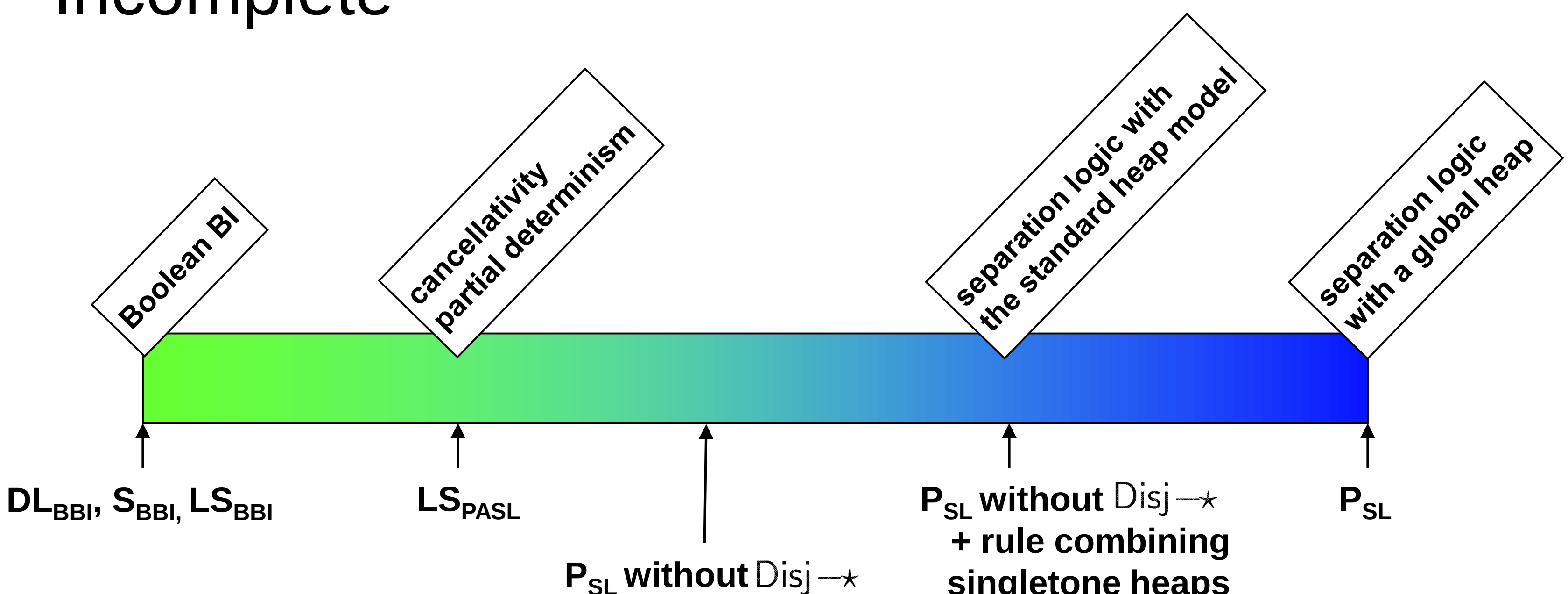
$$
\begin{array}{rlll}
\text{formula} & A & ::= & P \mid \bot \mid \neg A \mid A \vee A \mid \\
 & & & \mathsf{I} \mid A \star A \mid A \multimap\!\star\, A \mid \exists a.A \\
\text{primitive formula} & P & ::= & [l \mapsto E] \mid E = E \mid \cdots \\
\text{expression} & E & ::= & x \mid a \mid \mathsf{L} \mid \cdots \\
\text{location expression} & l & ::= & x \mid a \mid \mathsf{L} \\
\text{value} & V & ::= & \mathsf{L} \mid \cdots \\
\text{location} & \mathsf{L}_1, \mathsf{L}_2, \cdots & & \\
\text{stack variable} & x, y, z & & \\
\text{local variable} & a, b, c & &
\end{array}
$$

- $[l \mapsto E]$ describes a singleton heap at location l.
- Stack variables = program variables
- list, tree 등과 같은 inductive predicate은 우선 제외

분리 논리 시스템 P_{SL}

- P_{SL} 첫 버전은 분리 논리와 다른 semantics 이용
  - Unsound
- 새로 완성한 P_{SL}
  - Sound
  - Incomplete





1

Rules for points-to relations:

$$\frac{\Theta; \Sigma, w \doteq [l \mapsto E] \parallel \Pi, [\Gamma \Longrightarrow \Delta]^w}{\Theta; \Sigma \parallel \Pi, [\Gamma, [l \mapsto E] \Longrightarrow \Delta]^w} \mapsto\!\mathsf{L} \qquad \frac{\Theta; \Sigma, w \not\doteq [l \mapsto E] \parallel \Pi, [\Gamma \Longrightarrow \Delta]^w}{\Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, [l \mapsto E]]^w} \mapsto\!\mathsf{R}$$

Rules for propositional formulas:

$$\frac{}{\Theta; \Sigma \parallel \Pi, [\Gamma, \bot \Longrightarrow \Delta]^w} \bot\mathsf{L} \qquad \frac{\Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, A]^w}{\Theta; \Sigma \parallel \Pi, [\Gamma, \neg A \Longrightarrow \Delta]^w} \neg\mathsf{L} \qquad \frac{\Theta; \Sigma \parallel \Pi, [\Gamma, A \Longrightarrow \Delta]^w}{\Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, \neg A]^w} \neg\mathsf{R}$$

$$\frac{\Theta; \Sigma \parallel \Pi, [\Gamma, A \Longrightarrow \Delta]^w \quad \Theta; \Sigma \parallel \Pi, [\Gamma, B \Longrightarrow \Delta]^w}{\Theta; \Sigma \parallel \Pi, [\Gamma, A \vee B \Longrightarrow \Delta]^w} \vee\mathsf{L} \qquad \frac{\Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, A, B]^w}{\Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, A \vee B]^w} \vee\mathsf{R}$$

Rules for multiplicative formulas:

$$\frac{\Theta; \Sigma, w \doteq \epsilon \parallel \Pi, [\Gamma \Longrightarrow \Delta]^w}{\Theta; \Sigma \parallel \Pi, [\Gamma, \mathsf{I} \Longrightarrow \Delta]^w} \mathsf{IL} \qquad \frac{\Theta; \Sigma, w \not\doteq \epsilon \parallel \Pi, [\Gamma \Longrightarrow \Delta]^w}{\Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, \mathsf{I}]^w} \mathsf{IR}$$

$$\frac{\textit{fresh } w_1, w_2 \qquad \Theta; \Sigma, w \doteq w_1 \circ w_2 \parallel \Pi, [\Gamma \Longrightarrow \Delta]^w, [A \Longrightarrow \cdot]^{w_1}, [B \Longrightarrow \cdot]^{w_2}}{\Theta; \Sigma \parallel \Pi, [\Gamma, A \star B \Longrightarrow \Delta]^w} \star\mathsf{L}$$

$$\frac{w \doteq w_1 \circ w_2 \in \Sigma \qquad \begin{array}{l} \Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, A \star B]^w, [\Gamma_1 \Longrightarrow \Delta_1, A]^{w_1}, [\Gamma_2 \Longrightarrow \Delta_2]^{w_2} \\ \Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, A \star B]^w, [\Gamma_1 \Longrightarrow \Delta_1]^{w_1}, [\Gamma_2 \Longrightarrow \Delta_2, B]^{w_2} \end{array}}{\Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, A \star B]^w, [\Gamma_1 \Longrightarrow \Delta_1]^{w_1}, [\Gamma_2 \Longrightarrow \Delta_2]^{w_2}} \star\mathsf{R}$$

$$\frac{w_2 \doteq w \circ w_1 \in \Sigma \qquad \begin{array}{l} \Theta; \Sigma \parallel \Pi, [\Gamma, A \mathrel{-\!\!\star} B \Longrightarrow \Delta]^w, [\Gamma_1 \Longrightarrow \Delta_1, A]^{w_1}, [\Gamma_2 \Longrightarrow \Delta_2]^{w_2} \\ \Theta; \Sigma \parallel \Pi, [\Gamma, A \mathrel{-\!\!\star} B \Longrightarrow \Delta]^w, [\Gamma_1 \Longrightarrow \Delta_1]^{w_1}, [\Gamma_2, B \Longrightarrow \Delta_2]^{w_2} \end{array}}{\Theta; \Sigma \parallel \Pi, [\Gamma, A \mathrel{-\!\!\star} B \Longrightarrow \Delta]^w, [\Gamma_1 \Longrightarrow \Delta_1]^{w_1}, [\Gamma_2 \Longrightarrow \Delta_2]^{w_2}} \mathrel{-\!\!\star}\mathsf{L}$$

$$\frac{\textit{fresh } w_1, w_2 \qquad \Theta; \Sigma, w_2 \doteq w \circ w_1 \parallel \Pi, [\Gamma \Longrightarrow \Delta]^w, [A \Longrightarrow \cdot]^{w_1}, [\cdot \Longrightarrow B]^{w_2}}{\Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, A \mathrel{-\!\!\star} B]^w} \mathrel{-\!\!\star}\mathsf{R}$$

Rules for first-order formulas:

$$\frac{\textit{fresh } x \qquad \Theta; \Sigma \parallel \Pi, [\Gamma, [x/a]A \Longrightarrow \Delta]^w}{\Theta; \Sigma \parallel \Pi, [\Gamma, \exists a.A \Longrightarrow \Delta]^w} \exists\mathsf{L} \qquad \frac{\Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, [E/a]A, \exists a.A]^w}{\Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, \exists a.A]^w} \exists\mathsf{R}$$

Rules for primitive formulas for expressions:

$$\frac{\Theta, E = E'; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta]^w}{\Theta; \Sigma \parallel \Pi, [\Gamma, E = E' \Longrightarrow \Delta]^w} {=}\mathsf{L} \qquad \frac{\Theta, E \neq E'; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta]^w}{\Theta; \Sigma \parallel \Pi, [\Gamma \Longrightarrow \Delta, E = E']^w} {=}\mathsf{R} \qquad \frac{\Theta \vdash \bot}{\Theta; \Sigma \parallel \Pi} \mathsf{ExpCont}$$

2

Rule for disambiguating heap relations: { }

$$\frac{\{w \doteq u_1 \circ u_2, w \doteq v_1 \circ v_2\} \subset \Sigma \quad \textit{fresh}\ w_1, w_2, w_3, w_4 \quad \Theta; \Sigma, \begin{array}{l} u_1 \doteq w_1 \circ w_2, \\ u_2 \doteq w_3 \circ w_4, \\ v_1 \doteq w_1 \circ w_3, \\ v_2 \doteq w_2 \circ w_4 \end{array} \parallel \Pi, \begin{array}{l} [\cdot \implies \cdot]^{w_1}, \\ [\cdot \implies \cdot]^{w_2}, \\ [\cdot \implies \cdot]^{w_3}, \\ [\cdot \implies \cdot]^{w_4} \end{array}}{\Theta; \Sigma \parallel \Pi} \text{Disj}\star$$

Rules for applying associativity of the union of disjoint heaps {:}

$$\frac{\{w \doteq u \circ v, u \doteq u_1 \circ u_2\} \subset \Sigma \quad \textit{fresh}\ u' \quad \Theta; \Sigma, u' \doteq u_2 \circ v, w \doteq u_1 \circ u' \parallel \Pi, [\cdot \implies \cdot]^{u'}}{\Theta; \Sigma \parallel \Pi} \text{Assoc}$$

Rules for propagating atomic heap relations:

$$\frac{\{w \doteq \epsilon, w \doteq w_1 \circ w_2\} \subset \Sigma \quad \Theta; \Sigma, w_1 \doteq \epsilon, w_2 \doteq \epsilon \parallel \Pi}{\Theta; \Sigma \parallel \Pi} \text{Prop}\,\epsilon$$

$$\frac{\{w \doteq [l \mapsto E], w \doteq w_1 \circ w_2\} \subset \Sigma \quad \begin{array}{l} \Theta; \Sigma, w_1 \doteq [l \mapsto E], w_2 \doteq \epsilon \parallel \Pi \\ \Theta; \Sigma, w_1 \doteq \epsilon, w_2 \doteq [l \mapsto E] \parallel \Pi \end{array}}{\Theta; \Sigma \parallel \Pi} \text{Prop}\mapsto$$

Rules for normalizing heap relations:

$$\frac{\Theta; [w/w'](\Sigma, w \doteq u \circ v) \parallel [w/w']\Pi}{\Theta; \Sigma, w \doteq u \circ v, w' \doteq u \circ v \parallel \Pi} \text{NormEq}$$

$$\frac{\Theta; [w/u](\Sigma, v \doteq \epsilon) \parallel [w/u]\Pi}{\Theta; \Sigma, w \doteq u \circ v, v \doteq \epsilon \parallel \Pi} \text{NormPC} \qquad \frac{\Theta; [w/u](\Sigma, w \doteq \epsilon) \parallel [w/u]\Pi}{\Theta; \Sigma, w \doteq \epsilon, u \doteq \epsilon \parallel \Pi} \text{NormEmpty}$$

Rules for creating an empty heap and applying the monoid laws for empty heaps:

$$\frac{\textit{fresh}\ w_\epsilon \quad \Theta; \Sigma, w_\epsilon \doteq \epsilon \parallel \Pi, [\cdot \implies \cdot]^{w_\epsilon}}{\Theta; \Sigma \parallel \Pi} \text{ENew}$$

$$\frac{w_\epsilon \doteq \epsilon \in \Sigma \quad \Theta; \Sigma, w \doteq w \circ w_\epsilon \parallel \Pi}{\Theta; \Sigma \parallel \Pi} \text{EJoin} \qquad \frac{w \doteq w \circ u \in \Sigma \quad \Theta; \Sigma, u \doteq \epsilon \parallel \Pi}{\Theta; \Sigma \parallel \Pi} \text{ECancel}$$

3

$$\frac{}{\Theta; \Sigma, w \doteq \epsilon, w \doteq [l \mapsto E] \parallel \Pi}\ \mathsf{Cont}\,\epsilon{\mapsto} \qquad \frac{}{\Theta; \Sigma, w \doteq \epsilon, w \not\doteq \epsilon \parallel \Pi}\ \mathsf{Cont}\,\epsilon{\not\doteq}$$

$$\frac{\Theta, l = l', E = E'; \Sigma, w \doteq [l \mapsto E], w \doteq [l' \mapsto E'] \parallel \Pi}{\Theta; \Sigma, w \doteq [l \mapsto E], w \doteq [l' \mapsto E'] \parallel \Pi}\ \mathsf{Cont}{\mapsto}{\doteq} \qquad \frac{\begin{array}{c}\Theta, l \neq l'; \Sigma, w \doteq [l \mapsto E], w \not\doteq [l' \mapsto E'] \parallel \Pi \\ \Theta, E \neq E'; \Sigma, w \doteq [l \mapsto E], w \not\doteq [l' \mapsto E'] \parallel \Pi\end{array}}{\Theta; \Sigma, w \doteq [l \mapsto E], w \not\doteq [l' \mapsto E'] \parallel \Pi}\ \mathsf{Cont}{\mapsto}{\not\doteq}$$

$$\frac{\Theta, l_1 \neq l_2; \Sigma, w \doteq w_1 \circ w_2, w_1 \doteq [l_1 \mapsto E_1], w_2 \doteq [l_2 \mapsto E_2] \parallel \Pi}{\Theta; \Sigma, w \doteq w_1 \circ w_2, w_1 \doteq [l_1 \mapsto E_1], w_2 \doteq [l_2 \mapsto E_2] \parallel \Pi}\ \mathsf{Cont}\,{\circ}{\mapsto} \qquad \frac{}{\Theta; \Sigma, w \doteq u \circ u, u \doteq [l \mapsto E] \parallel \Pi}\ \mathsf{Cont}\,{\circ}{\mapsto}2$$

반례

- 분리 논리에서는 참이지만
- P_{SL}에서는 증명 불가능

$$\neg(\neg\mathsf{I} \mathrel{-\!\!\star} \mathsf{I}),$$

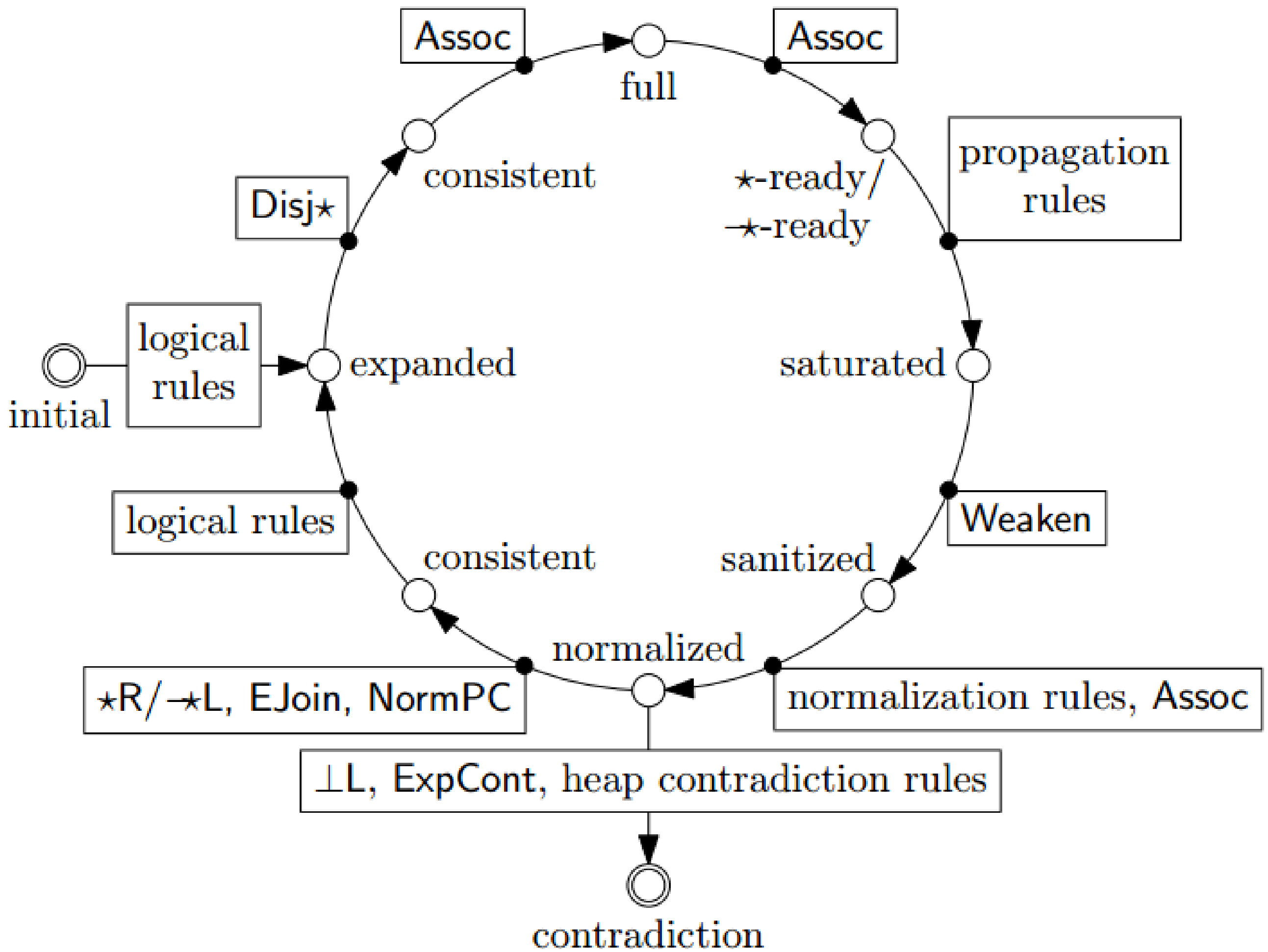
$$\mathsf{I} \supset \neg([l \mapsto E] \mathrel{-\!\!\star} \neg[l \mapsto E]).$$

$$\neg\mathsf{I} \supset \Big((\neg\mathsf{I} \wedge \neg(\neg\mathsf{I} \star \neg\mathsf{I})) \star \top\Big).$$

- Incompleteness는 실질적으로 크게 문제되지 않음

$$\begin{aligned} & preLoopInvR(Stack, P, T, STree, root) \\ \equiv\ & noDanglingR \wedge noDangling(T) \wedge noDangling(P) \\ & \wedge listMarkedNodesR(Stack, P) * (restoredListR(Stack, T) \mathrel{-\!\!*} spansR(STree, root)) \\ & \wedge markedR * \begin{pmatrix} unmarkedR \\ \wedge (\forall x.\ allocated(x) \to (reach(T, x) \vee reachRightChildInList(Stack, x))) \end{pmatrix} \end{aligned}$$

P_{SL}을 구현하자 – 증명 탐색 전략 SS



SS의 성질

- Sound
- 항상 종결
- P_{SL}에 대해서 incomplete
  - P_{SL}에서 증명 가능
  - SS로는 증명 불가능

$$(\neg \mathsf{I} \star \neg \mathsf{I}) \supset (A \star A), \text{ where } A = \neg \mathsf{I} \wedge \neg([l \mapsto E] \star [l' \mapsto E])$$

- 괘한타!

계획

- 분리 논리 증명기 개발
- Inductive predicate 추가
  – 실제 proof tree를 제시할 수 있다는 점이 강점
  – file:///Z:/psl14-leewy/cceye-leewy-140120/result-weaken/6.4-2.pdf
  – Cf. Dynamic frame technique
- SMT solver 연결 (Z3)

- 세계 최초 Schorr-Waite algorithm <u>기계적</u> 검증!

감사합니다

gla@postech.ac.kr