A Theory of Non-associative Classical BI

포항공과대학교 프로그래밍 언어 연구실 박종현

목표

- 미션
  - 포인터 연산 및 동적 메모리 할당 등을 사용한
  - 산업적으로 활용되는 C 프로그램을 대상으로
  - 배우기 쉽고 사용하기 편리한
  - 연역 검증 도구를 개발하자!

- 세부 목표
  - 분리 논리에 기초한 연역 검증 도구 개발

연역 검증?

- 연역 검증이란?
  - 프로그램이 특정 성질을 만족하는지 여부를 정리 증명 기법을 이용해서 엄밀하게 증명
- 왜?
  - 복잡한 성질에 대한 엄밀한 검증 요구
    - Ex) 임베디드 소프트웨어
  - 점차 그 중요성이 증대되고 있음
- 이론?
  - 분리 논리
  - ……

분리 논리?

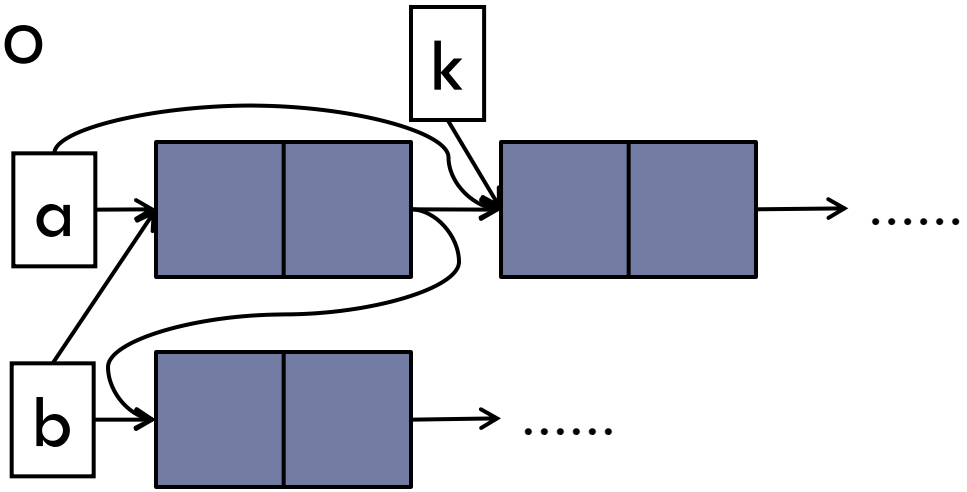
- John C. Reynolds가 제안한 확장된 Hoare 논리
  - 분리 연산자 제공

- 포인터 연산 및 동적 메모리 할당을 수행하는 프로그램에 대한 검증이 용이

- **지역 추론**이 가능
  - 사람의 직관적인 검증 방식과 유사
  - 지역 추론 결과로부터 전역 추론 결과를 도출
  - 증명의 복잡도 감소 ⇒ 복잡한 프로그램의 검증 가능

Hoare 논리 vs. 분리 논리

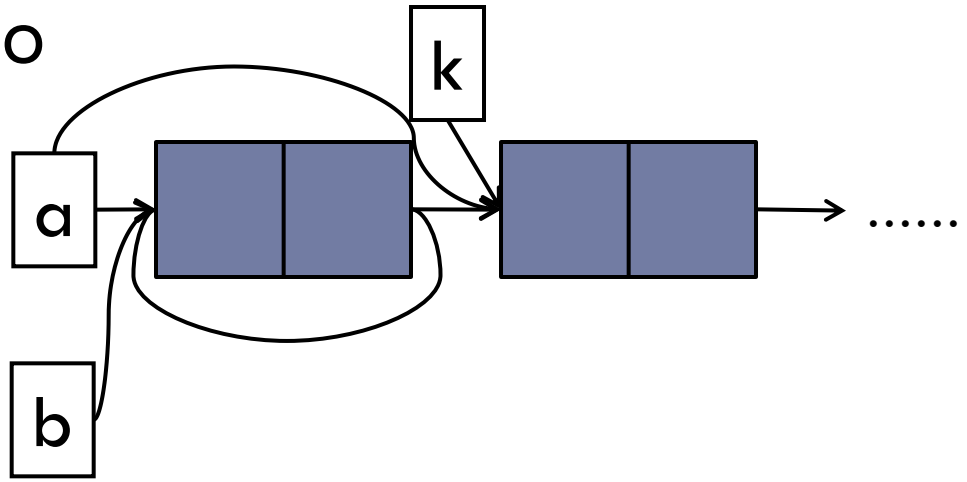
```
b := nil
while a != nil do
  k := [a + 1];
  [a + 1] := b;
  b := a;
  a := k;
end while
```



Hoare 논리 vs. 분리 논리

```
b :: α0^R = α^R · β
while a != nil do
  k := [a + 1];
  [a + 1] := b;
  b := a;
  a := k;
end while
```

$\alpha_0^R = \beta$



오류 발생???

실제로는 발생하지 않음

Hoare 논리 vs. 분리 논리

```
while a != nil do
  k := [a + 1];
  [a + 1] := b;
  b := a;
  a := k;
end while
```

Hoare 논리:

$(\exists\, \alpha, \beta.\ \mathrm{List}\ \alpha\ a \wedge \mathrm{List}\ \beta\ b \wedge {\alpha_0}^R = \alpha^R \cdot \beta) \wedge$
$(\forall\, k.\ \mathrm{Reach}(a, k) \wedge \mathrm{Reach}(b, k) \Rightarrow k = \mathrm{nil})$

분리 논리:

$(\exists\, \alpha, \beta.\ \mathrm{List}\ \alpha\ a * \mathrm{List}\ \beta\ b \wedge {\alpha_0}^R = \alpha^R \cdot \beta)$

Hoare 논리 vs. 분리 논리

```
while a != nil do
  k := [a + 1];
  [a + 1] := b;
  b := a;
  a := k;
end while
```

Hoare 논리:

$(\exists\, \alpha, \beta.\ \mathrm{List}\ \alpha\ a \wedge \mathrm{List}\ \beta\ b \wedge \alpha_0^R = \alpha^R \cdot \beta) \wedge \mathrm{List}\ \gamma\ x$

$(\forall\, k.\ \mathrm{Reach}(a, k) \wedge \mathrm{Reach}(b, k) \Rightarrow k = \mathrm{nil}) \wedge$

$(\forall\, k.\ \mathrm{Reach}(x, k) \wedge (\mathrm{Reach}(a, k) \vee \mathrm{Reach}(b, k)) \Rightarrow k = \mathrm{nil})$

분리 논리:

$(\exists\, \alpha, \beta.\ \mathrm{List}\ \alpha\ a * \mathrm{List}\ \beta\ b * \mathrm{List}\ \gamma\ x \wedge \alpha_0^R = \alpha^R \cdot \beta)$

⇧

$(\exists\, \alpha, \beta.\ \mathrm{List}\ \alpha\ a * \mathrm{List}\ \beta\ b \wedge \alpha_0^R = \alpha^R \cdot \beta)$

만약에 메모리에 다른 독립된 리스트가 있다면???

문제점?

- 분리 논리의 이론적 기초?
  - 부울 BI 논리
    - 고전 논리 + 직관 선형 논리

- 부울 BI 논리를 위한 컷 제거 귀추 계산법 X
  - 효율적인 자동 정리 증명 기법 설계가 어려움

부울 BI를 위한 컷 제거 귀추 계산법 개발

진행 상황 (~2010. 02)

직관 BI 논리를 위한 컷 제거 귀추 계산법 연구

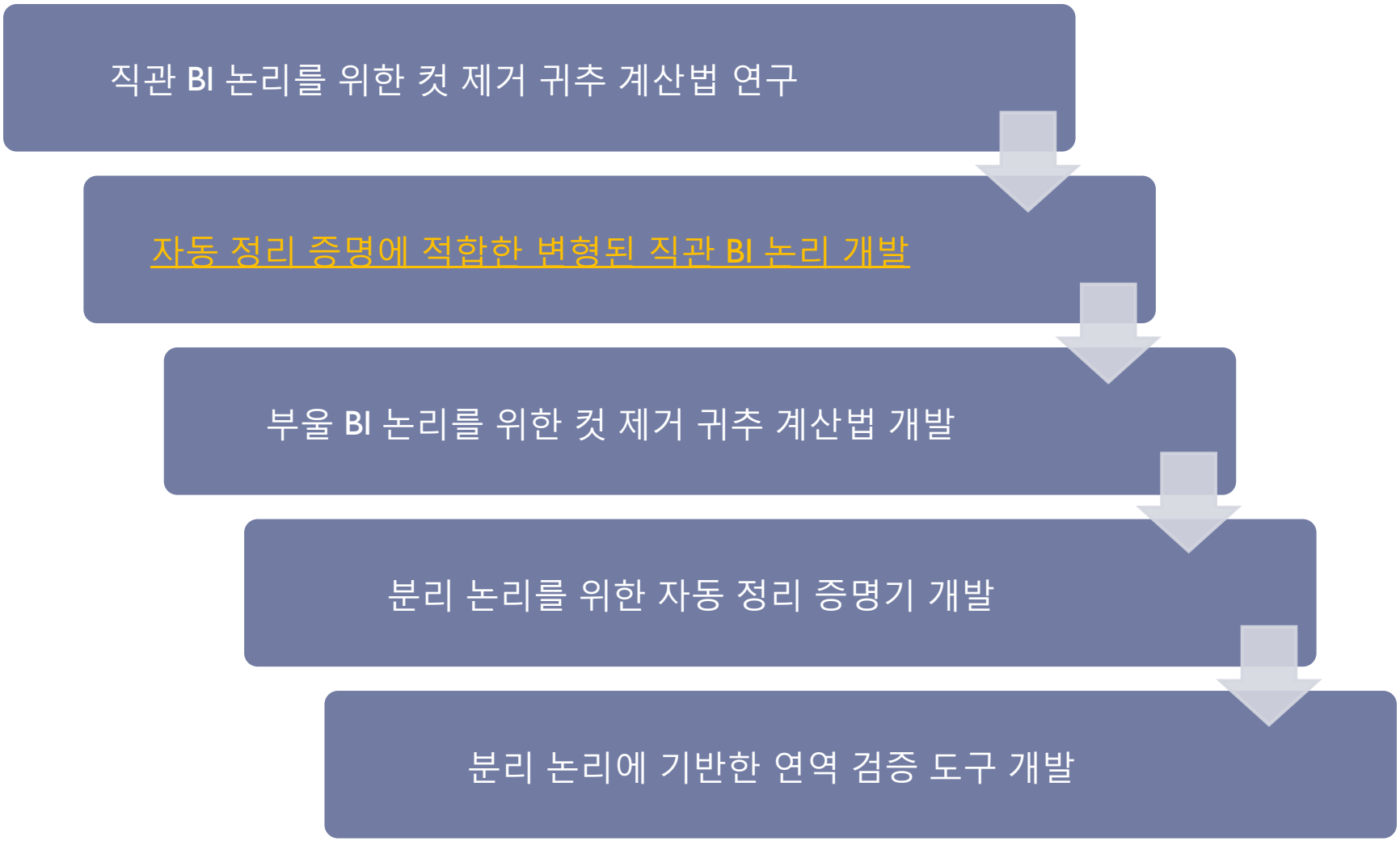
자동 정리 증명에 적합한 변형된 직관 BI 논리 개발

부울 BI 논리를 위한 컷 제거 귀추 계산법 개발

분리 논리를 위한 자동 정리 증명기 개발

분리 논리에 기반한 연역 검증 도구 개발

진행 상황 (~2010. 02)


직관 BI 논리를 위한 컷 제거 귀추 계산법 연구
자동 정리 증명에 적합한 변형된 직관 BI 논리 개발
부울 BI 논리를 위한 컷 제거 귀추 계산법 개발
분리 논리를 위한 자동 정리 증명기 개발
분리 논리에 기반한 연역 검증 도구 개발

$$\frac{A \text{ atomic}}{A \Longrightarrow A}\ Init \qquad \frac{A \text{ atomic}}{\Delta; A \Longrightarrow A}\ Init'$$

$$\frac{\delta(\Delta) \Longrightarrow C}{\delta(\Delta, \mathsf{I}) \Longrightarrow C}\ \mathsf{W}' \qquad \frac{\delta(\Delta) \Longrightarrow C}{\delta(\Delta, (\mathsf{I}; \Sigma)) \Longrightarrow C}\ \mathsf{W}''$$

$$\frac{}{\Delta \Longrightarrow \top}\ \top R \qquad \frac{}{\delta(\bot) \Longrightarrow C}\ \bot L$$

$$\frac{\Delta; A \supset B \Longrightarrow A \quad \delta(\Delta; A \supset B; B) \Longrightarrow C}{\delta(\Delta; A \supset B) \Longrightarrow C}\ \supset L \qquad \frac{\Delta; A \Longrightarrow B}{\Delta \Longrightarrow A \supset B}\ \supset R$$

$$\frac{\delta(A; B; A \wedge B) \Longrightarrow C}{\delta(A \wedge B) \Longrightarrow C}\ \wedge L \qquad \frac{\Delta \Longrightarrow A \quad \Delta \Longrightarrow B}{\Delta \Longrightarrow A \wedge B}\ \wedge R$$

$$\frac{\delta(A; A \vee B) \Longrightarrow C \quad \delta(B; A \vee B) \Longrightarrow C}{\delta(A \vee B) \Longrightarrow C}\ \vee L \qquad \frac{\Delta \Longrightarrow A}{\Delta \Longrightarrow A \vee B}\ \vee R_L \qquad \frac{\Delta \Longrightarrow B}{\Delta \Longrightarrow A \vee B}\ \vee R_R$$

$$\frac{\Delta \Longrightarrow A \quad \delta(\Delta', B) \Longrightarrow C}{\delta(\mathsf{WC}[\Delta, \Delta', A \mathrel{-\!\!\star} B]) \Longrightarrow C}\ \mathrel{-\!\!\star} L \qquad \frac{\mathsf{I} \Longrightarrow A \quad \delta(\Delta, B) \Longrightarrow C}{\delta(\mathsf{WC}[\Delta, A \mathrel{-\!\!\star} B]) \Longrightarrow C}\ \mathrel{-\!\!\star} L' \qquad \frac{\Delta \Longrightarrow A \quad \delta(B) \Longrightarrow C}{\delta(\mathsf{WC}[\Delta, A \mathrel{-\!\!\star} B]) \Longrightarrow C}\ \mathrel{-\!\!\star} L''$$

$$\frac{\Delta, A \Longrightarrow B}{\Delta \Longrightarrow A \mathrel{-\!\!\star} B}\ \mathrel{-\!\!\star} R \qquad \frac{\delta(A, B) \Longrightarrow C}{\delta(A \star B) \Longrightarrow C}\ \star L$$

$$\frac{\Delta \Longrightarrow A \quad \Delta' \Longrightarrow B}{\mathsf{WC}[\Delta, \Delta'] \Longrightarrow A \star B}\ \star R \qquad \frac{\mathsf{I} \Longrightarrow A \quad \Delta \Longrightarrow B}{\Delta \Longrightarrow A \star B}\ \star R' \qquad \frac{\Delta \Longrightarrow A \quad \mathsf{I} \Longrightarrow B}{\Delta \Longrightarrow A \star B}\ \star R''$$

진행 상황 (~2010. 02)

직관 BI 논리를 위한 컷 제거 귀추 계산법 연구

자동 정리 증명에 적합한 변형된 직관 BI 논리 개발

부울 BI 논리를 위한 컷 제거 귀추 계산법 개발

분리 논리를 위한 자동 정리 증명기 개발

분리 논리에 기반한 연역 검증 도구 개발

$$\frac{A \text{ atomic}}{\omega[A \longrightarrow_{\mathsf{B}} A]}\ \mathit{Init}$$

$$\frac{\omega[\Delta \longrightarrow_{\mathsf{B}} \Psi]}{\omega[\Delta; \Delta' \longrightarrow_{\mathsf{B}} \Psi]}\ \mathsf{W} \qquad \frac{\omega[\Delta \longrightarrow_{\mathsf{B}} \Psi]}{\omega[\Delta \longrightarrow_{\mathsf{B}} \Psi; A]}\ \mathsf{W}' \qquad \frac{\omega[\Delta; \Delta'; \Delta' \longrightarrow_{\mathsf{B}} \Psi]}{\omega[\Delta; \Delta' \longrightarrow_{\mathsf{B}} \Psi]}\ \mathsf{C} \qquad \frac{\omega[\Delta \longrightarrow_{\mathsf{B}} \Psi; A; A]}{\omega[\Delta \longrightarrow_{\mathsf{B}} \Psi; A]}\ \mathsf{C}'$$

$$\frac{}{\omega[\bot \longrightarrow_{\mathsf{B}} \cdot]}\ \bot L \qquad \frac{\omega[\Delta \longrightarrow_{\mathsf{B}} \Psi]}{\omega[\Delta \longrightarrow_{\mathsf{B}} \Psi; \bot]}\ \bot R \qquad \frac{\omega[\Delta \longrightarrow_{\mathsf{B}} A; \Psi]}{\omega[\Delta; \neg A \longrightarrow_{\mathsf{B}} \Psi]}\ \neg L \qquad \frac{\omega[\Delta; A \longrightarrow_{\mathsf{B}} \Psi]}{\omega[\Delta \longrightarrow_{\mathsf{B}} \neg A; \Psi]}\ \neg R$$

$$\frac{\omega[\Delta; A; B \longrightarrow_{\mathsf{B}} \Psi]}{\omega[\Delta; A \wedge B \longrightarrow_{\mathsf{B}} \Psi]}\ \wedge L \qquad \frac{\omega[\Delta \longrightarrow_{\mathsf{B}} A; \Psi] \quad \omega[\Delta \longrightarrow_{\mathsf{B}} B; \Psi']}{\omega[\Delta \longrightarrow_{\mathsf{B}} A \wedge B; \Psi; \Psi']}\ \wedge R$$

$$\frac{\omega[\Delta; \emptyset_{\mathsf{m}} \longrightarrow_{\mathsf{B}} \Psi]}{\omega[\Delta; \mathsf{I} \longrightarrow_{\mathsf{B}} \Psi]}\ \mathsf{I}L \qquad \frac{}{\omega[\emptyset_{\mathsf{m}} \longrightarrow_{\mathsf{B}} \mathsf{I}]}\ \mathsf{I}R$$

$$\frac{\omega[(\Delta' \longrightarrow_{\mathsf{B}} \Psi'; A), (\Delta \longrightarrow_{\mathsf{B}} \Psi); \Delta'' \longrightarrow_{\mathsf{B}} \Psi''] \quad \omega[B; \Delta'' \longrightarrow_{\mathsf{B}} \Psi'']}{\omega[(\Delta' \longrightarrow_{\mathsf{B}} \Psi'), (\Delta; A \mathrel{-\!\!\star} B \longrightarrow_{\mathsf{B}} \Psi); \Delta'' \longrightarrow_{\mathsf{B}} \Psi'']}\ \mathrel{-\!\!\star} L$$

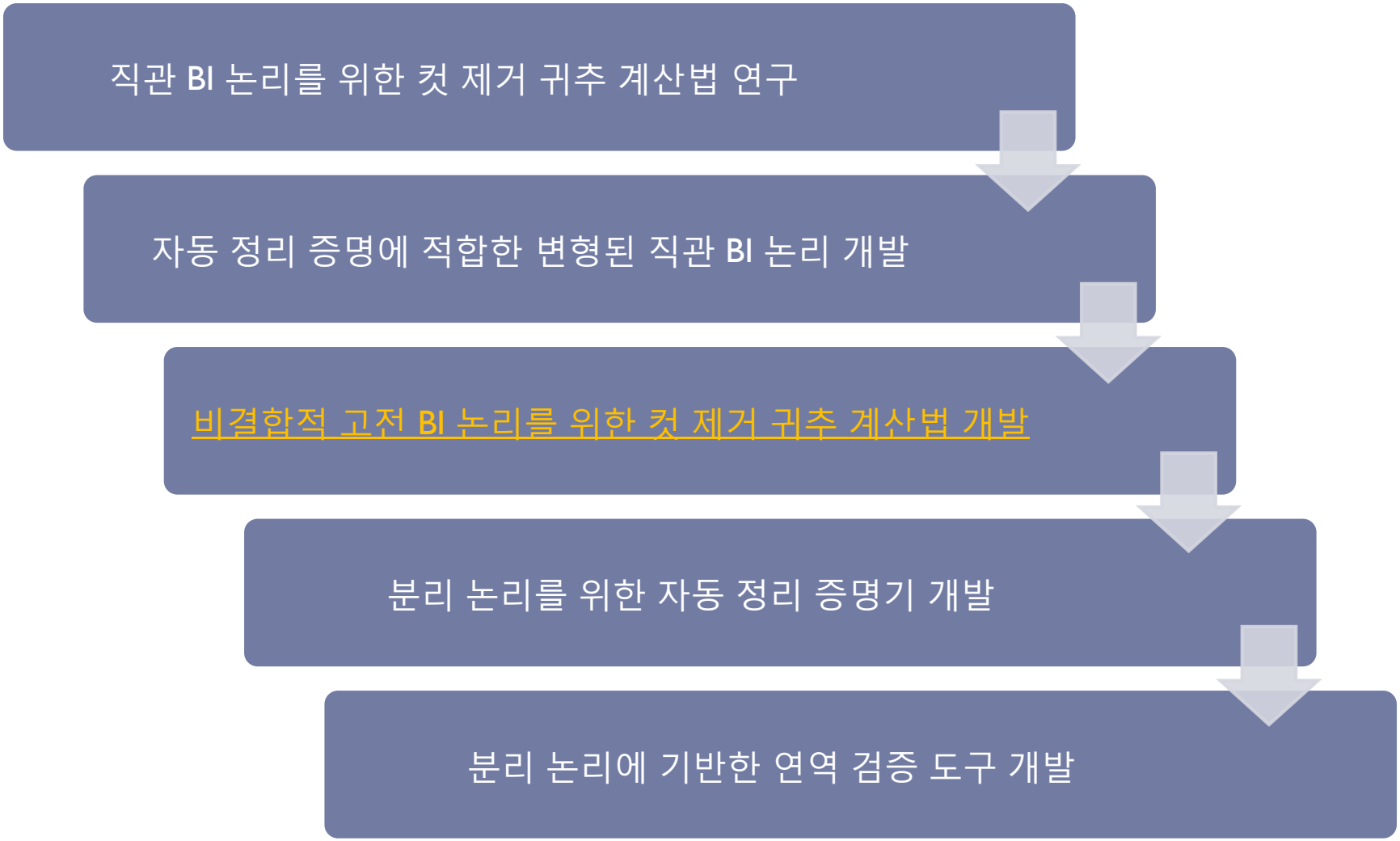
$$\frac{(\Delta \longrightarrow_{\mathsf{B}} \Psi), (A \longrightarrow_{\mathsf{B}} \cdot) \longrightarrow_{\mathsf{B}} B}{\omega[(\Delta \longrightarrow_{\mathsf{B}} A \mathrel{-\!\!\star} B; \Psi), (\Delta' \longrightarrow_{\mathsf{B}} \Psi'); \Delta'' \longrightarrow_{\mathsf{B}} \Psi'']}\ \mathrel{-\!\!\star} R$$

$$\frac{\omega[\Delta; (A \longrightarrow_{\mathsf{B}} \cdot), (B \longrightarrow_{\mathsf{B}} \cdot) \longrightarrow_{\mathsf{B}} \Psi]}{\omega[\Delta; A \star B \longrightarrow_{\mathsf{B}} \Psi]}\ \star L$$

$$\frac{\omega[\Delta''; (\Delta \longrightarrow_{\mathsf{B}} \Psi; A), (\Delta' \longrightarrow_{\mathsf{B}} \Psi') \longrightarrow_{\mathsf{B}} \Psi''] \quad \omega[\Delta''; (\Delta \longrightarrow_{\mathsf{B}} \Psi), (\Delta' \longrightarrow_{\mathsf{B}} \Psi'; B) \longrightarrow_{\mathsf{B}} \Psi'']}{\omega[\Delta''; (\Delta \longrightarrow_{\mathsf{B}} \Psi), (\Delta' \longrightarrow_{\mathsf{B}} \Psi') \longrightarrow_{\mathsf{B}} A \star B; \Psi'']}\ \star R$$

Theorem 4.1 (cut elimination). *If* $\omega[\Delta \longrightarrow_{\mathsf{B}} \Psi; C]$ *and* $\omega[\Delta; C \longrightarrow_{\mathsf{B}} \Psi]$, *then* $\omega[\Delta \longrightarrow_{\mathsf{B}} \Psi]$.

진행 상황 (~2010. 08)


직관 BI 논리를 위한 컷 제거 귀추 계산법 연구
자동 정리 증명에 적합한 변형된 직관 BI 논리 개발
비결합적 고전 BI 논리를 위한 컷 제거 귀추 계산법 개발
분리 논리를 위한 자동 정리 증명기 개발
분리 논리에 기반한 연역 검증 도구 개발

비결합적 고전 BI 논리?

- 부울 BI 논리와의 공통점
  - 부울 BI 논리와 같은 연산자 사용
  - 고전 논리 + 선형 논리
    - 고전 논리? $\neg\neg A \supset A$ 성립
    - 선형 논리? 분리 연산자 지원

- 부울 BI 논리와의 차이점
  - 비결합적 분리 연산자
    $A \star (B \star C) \supset (A \star B) \star C$
    $(A \star B) \star C \supset A \star (B \star C)$
    $A \star \mathsf{I} \supset A$
    $A \supset A \star \mathsf{I}$
  - 분리 연산자의 고전적 해석

연구 결과

- 비결합적 고전 BI 논리에 대한 귀추 계산법 완성
  - 새로운 형태의 귀추 계산법 제안
- 컷 제거 성질 증명

Lemma 4.2. *If* $l_{\mathcal{D}} \sim l_{\mathcal{E}}$ *and* $l_{\mathcal{D}}[\Delta \longrightarrow_{\mathsf{B}} \Psi; C]$ *and* $l_{\mathcal{E}}[\Delta'; C \longrightarrow_{\mathsf{B}} \Psi']$, *then* $l_{\mathcal{D}} \cdot l_{\mathcal{E}}[\Delta; \Delta' \longrightarrow_{\mathsf{B}} \Psi; \Psi']$.

  - 비결합적 성질을 이용
  - 부울 BI 논리를 위한 컷 제거 귀추 계산법이 없음을 암시
- PSPL 2010 논문 발표
  - 저널 논문 준비 중

향후 계획

- 직관 BI 논리를 위한 컷 제거 귀추 계산법 연구
- 자동 정리 증명에 적합한 변형된 직관 BI 논리 개발
- 비결합적 고전 BI 논리를 위한 컷 제거 귀추 계산법 개발
- 비결합적 분리 논리를 위한 자동 정리 증명기 개발
- 비결합적 분리 논리에 기반한 연역 검증 도구 개발

향후 계획

- 비결합적 분리 논리?
  - 비결합적 고전 BI 논리에 기초한 프로그램 검증 논리
  - 컷 제거 귀추 계산법 존재 ⇒
    효율적인 자동 정리 증명 기법 설계 가능

- 프로그램 검증에 응용
  - 비결합적 성질이 존재하는 프로그램 성질의 검증에 응용

질의응답