

대형 C프로그램을 통째로 고속 분석하는 방법

오학주, 허기홍, 이원찬, 이우석, 이광근

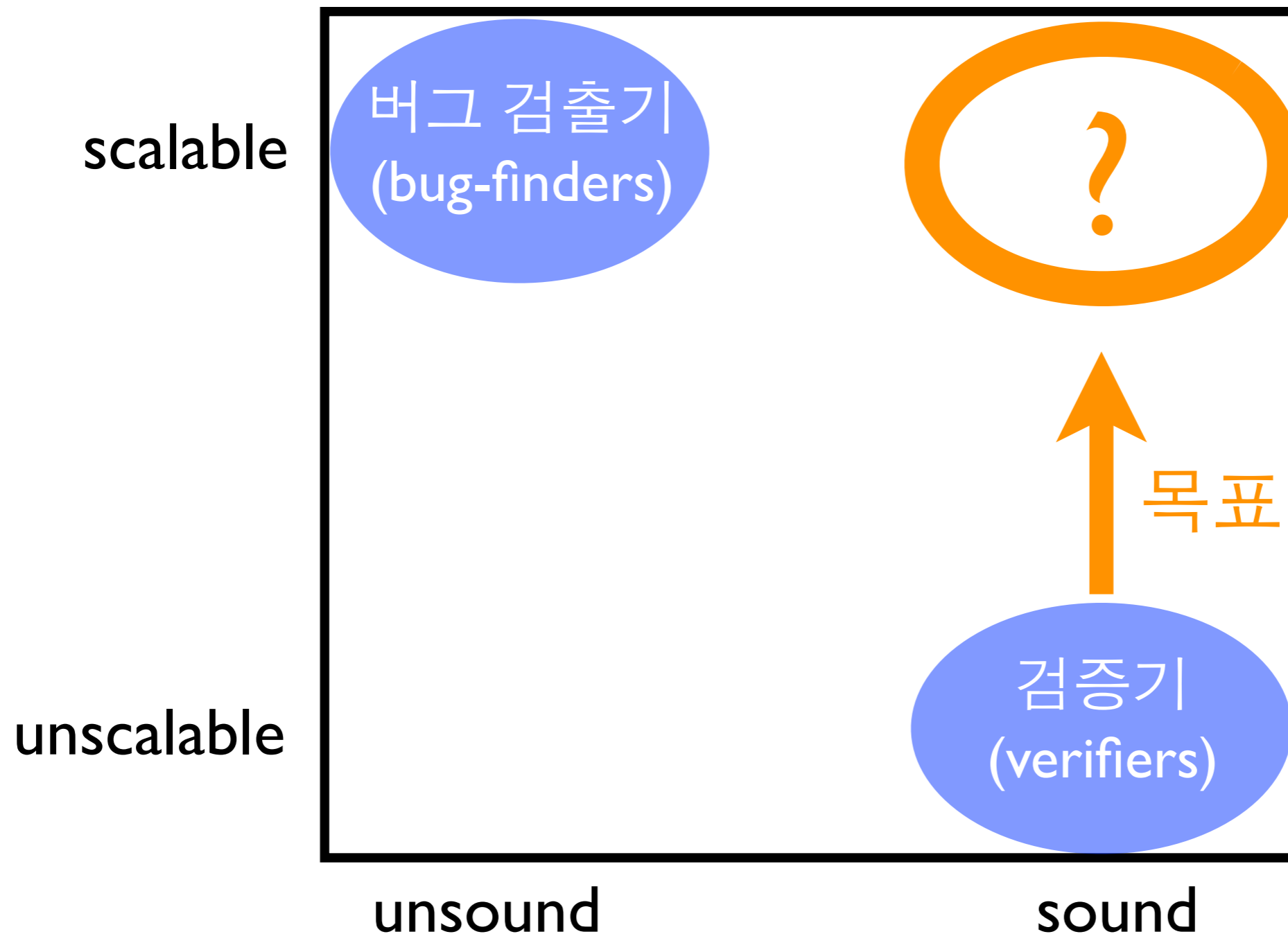
프로그래밍 연구실
서울대학교

2012.07.26 @ ROSAEC 워크샵


정적 분석의 난제

실제 실행을 모두 포섭하면서 (**sound**)
정확하게 (**precise**)
큰 프로그램을 (**scalable**) 분석하기

현실: “검출기” vs. “검증기”



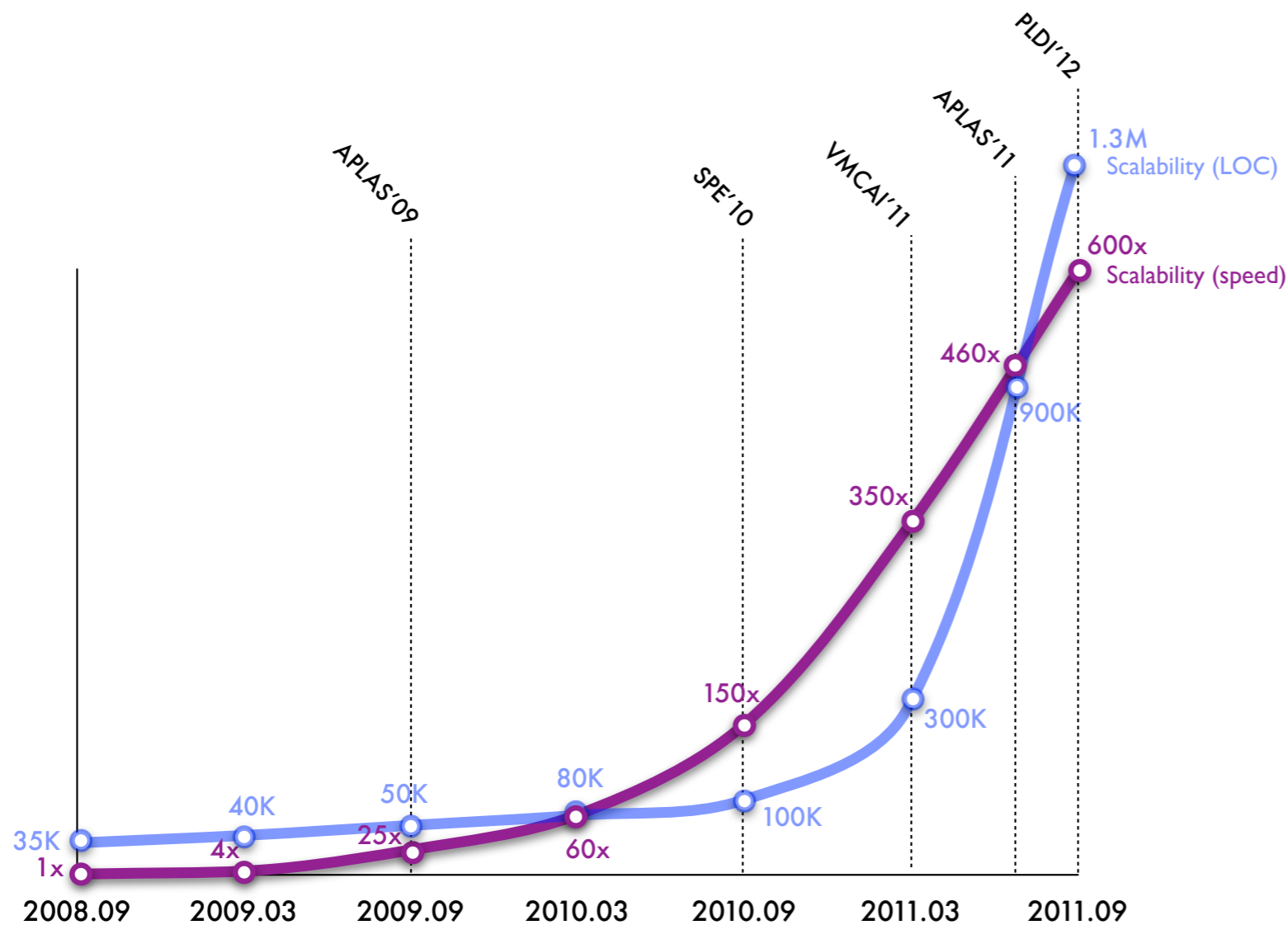
동기

- (2007) 정적 분석기 상용화  Sparrow
The Early Bird
 - C 프로그램 메모리 오류 검출기
 - 요약해석(abstract interpretation) 기반, 디자인은 안전
 - 현실은 큰 프로그램 분석 위해 안전성 포기
- 실제적인 분석기 실험 환경 갖춰짐
 - “안전한 버전의 성능을 높여보자”

속도/성능 향상 정도



sound & global analysis version



- **< 1.4M in 10hrs**
with intervals
- **< 0.14M in 20hrs**
with octagons

스파스 분석 디자인 이론 (Sparse Analysis Framework)

요약 해석

스파스 버전

$$\hat{F} : \hat{D} \rightarrow \hat{D} \quad \xrightarrow{\text{sparsify}} \quad \hat{F}_s : \hat{D} \rightarrow \hat{D}$$

$$\text{fix } \hat{F} \quad \stackrel{\text{still}}{=} \quad \text{fix } \hat{F}_s$$

*“An important strength is that the **theoretical result is very general** ... The result should be **highly influential on future work in sparse analysis.**” (from PLDI reviews)*

Sparse Analysis Framework

프로그램

$\langle \mathbb{C}, \hookrightarrow \rangle$

- \mathbb{C} : 프로그램 지점(program point)들의 집합
- $\hookrightarrow \subseteq \mathbb{C} \times \mathbb{C}$: 실행흐름관계 (control flow graph)

$c' \hookrightarrow c$ (c is the next program point to c')

cf) 요약 해석 (Abstract Interpretation)

- 올바른 정적분석 디자인을 위한 강력한 이론

실제 실행 $\llbracket P \rrbracket = \text{fix } F \in D$

요약 해석 $\llbracket \hat{P} \rrbracket = \text{fix } \hat{F} \in \hat{D}$ s.t. $D \begin{matrix} \xleftarrow{\gamma} \\ \xrightarrow{\alpha} \end{matrix} \hat{D}$
 $\alpha \circ F \sqsubseteq \hat{F} \circ \alpha$

결과 $\llbracket P \rrbracket \sqsubseteq \llbracket \hat{P} \rrbracket$

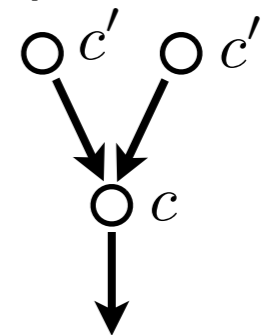
베이스라인 분석

- 요약의미공간(abstract domain): 각 프로그램 지점마다 도달가능한 상태들을 모으고 요약

$$\begin{aligned}
 [\hat{P}] \in \mathbb{C} \rightarrow \hat{\mathbb{S}} &= \text{fix } \hat{F} \\
 \hat{\mathbb{S}} &= \hat{\mathbb{L}} \rightarrow \hat{\mathbb{V}}
 \end{aligned}$$

- 요약 실행 함수 (abstract semantic function)

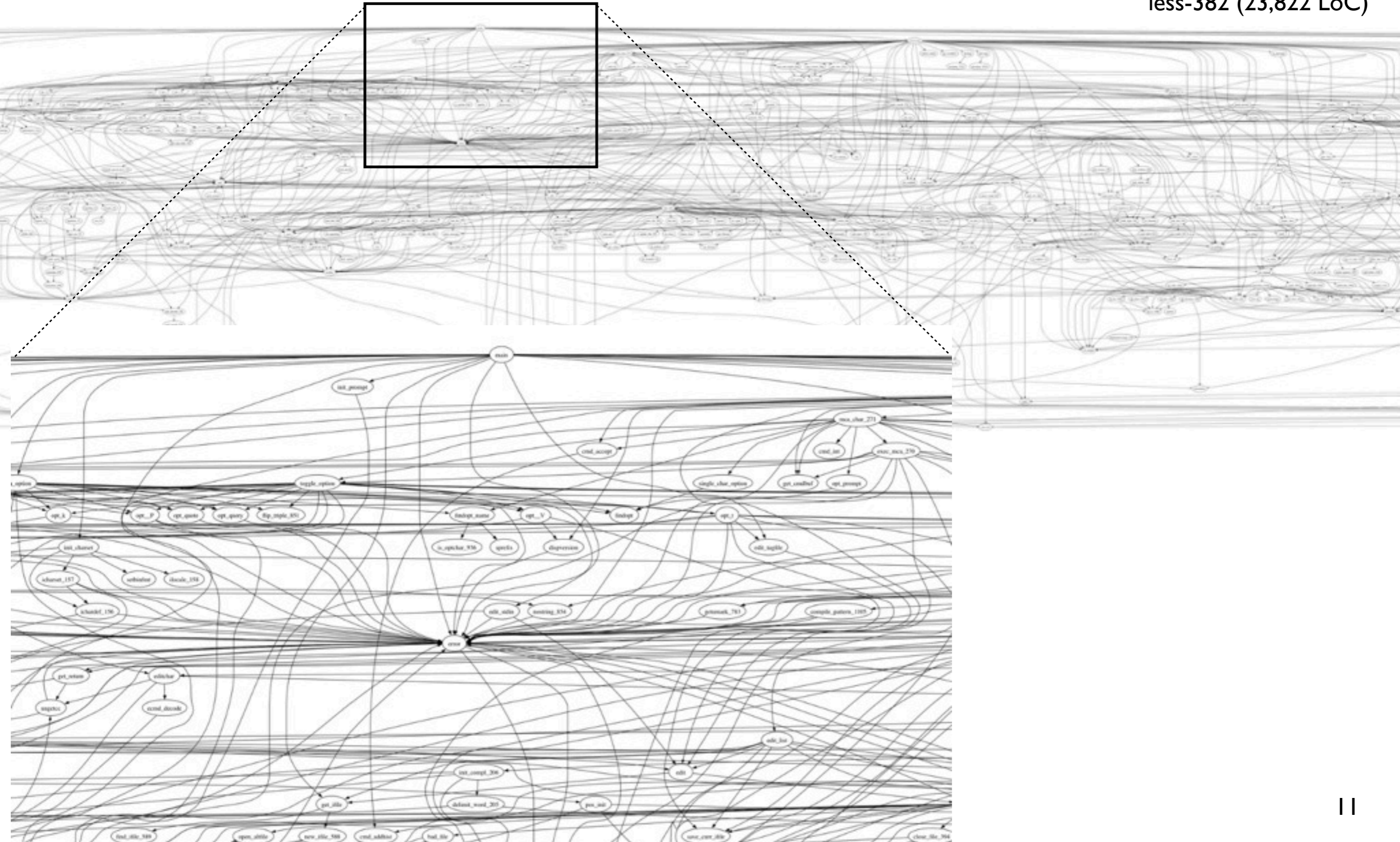
$$\begin{aligned}
 \hat{F} &\in (\mathbb{C} \rightarrow \hat{\mathbb{S}}) \rightarrow (\mathbb{C} \rightarrow \hat{\mathbb{S}}) \\
 \hat{F}(\hat{X}) &= \lambda c \in \mathbb{C}. \hat{f}_c \left(\bigsqcup_{c' \hookrightarrow c} \hat{X}(c') \right)
 \end{aligned}$$



$$\hat{f}_c \in \hat{\mathbb{S}} \rightarrow \hat{\mathbb{S}} : \text{abstract semantics at point } c$$

큰 프로그램 분석에 역부족

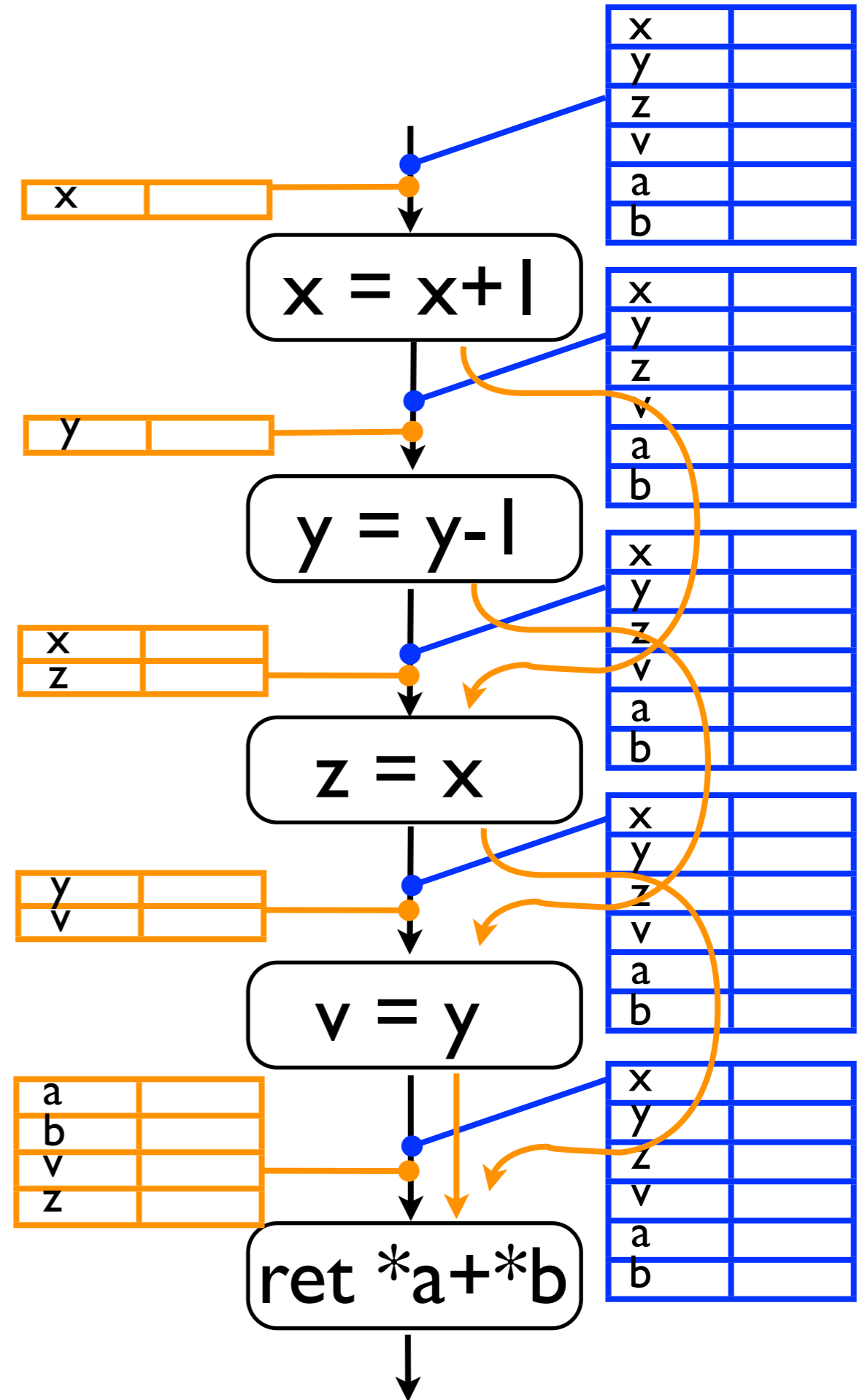
less-382 (23,822 LoC)



스파스 분석: 핵심 아이디어

$$\hat{F}(\hat{X}) = \lambda_{c \in \mathbb{C}} \cdot \hat{f}_c \left(\bigsqcup_{c' \hookrightarrow c} \hat{X}(c') \right).$$

replace syntactic dependency
by semantic dependency
(data dependency)



스파스 분석 유도

분석기는 의미함수 고정점을 계산 $\hat{F} \in (\mathbb{C} \rightarrow \hat{\mathcal{S}}) \rightarrow (\mathbb{C} \rightarrow \hat{\mathcal{S}})$

- 베이스라인 분석

$$\hat{F}(\hat{X}) = \lambda c \in \mathbb{C} \cdot \hat{f}_c \left(\bigsqcup_{c' \hookrightarrow c} \hat{X}(c') \right).$$

- 스파스 분석의 수학적 정의 (실현 불가능)

$$\hat{F}_s(\hat{X}) = \lambda c \in \mathbb{C} \cdot \hat{f}_c \left(\bigsqcup_{c' \overset{l}{\rightsquigarrow} c} \hat{X}(c') \mid l \right).$$

- 실현가능한 스파스 분석

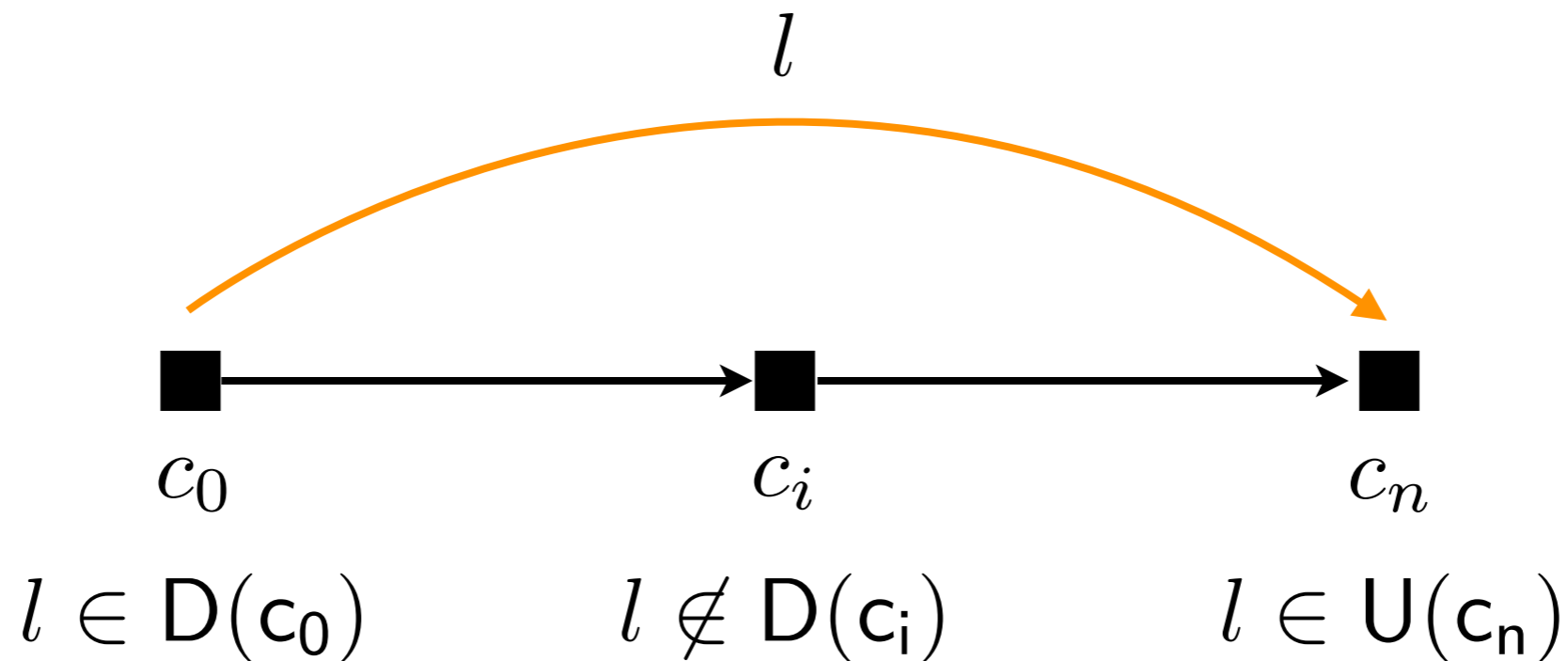
$$\hat{F}_a(\hat{X}) = \lambda c \in \mathbb{C} \cdot \hat{f}_c \left(\bigsqcup_{c' \overset{l}{\rightsquigarrow}_a c} \hat{X}(c') \mid l \right).$$

스파스 분석 (실현 불가능)

$$\hat{F}_s(\hat{X}) = \lambda c \in \mathbb{C}. \hat{f}_c \left(\bigsqcup_{c' \rightsquigarrow c} \hat{X}(c') | l \right).$$

Data Dependency

$$c_0 \rightsquigarrow^l c_n \triangleq \exists c_0 \dots c_n \in \text{Paths}, l \in \hat{\mathbb{L}}. \\ l \in D(c_0) \cap U(c_n) \wedge \forall i \in (0, n). l \notin D(c_i)$$



스파스 분석 (실현 불가능)

$$\hat{F}_s(\hat{X}) = \lambda c \in \mathbb{C}. \hat{f}_c(\bigsqcup_{c' \rightsquigarrow c} \hat{X}(c')|_l).$$

Data Dependency

$$c_0 \rightsquigarrow^l c_n \triangleq \exists c_0 \dots c_n \in \text{Paths}, l \in \hat{\mathbb{L}}. \\ l \in D(c_0) \cap U(c_n) \wedge \forall i \in (0, n). l \notin D(c_i)$$

Def-Use Sets

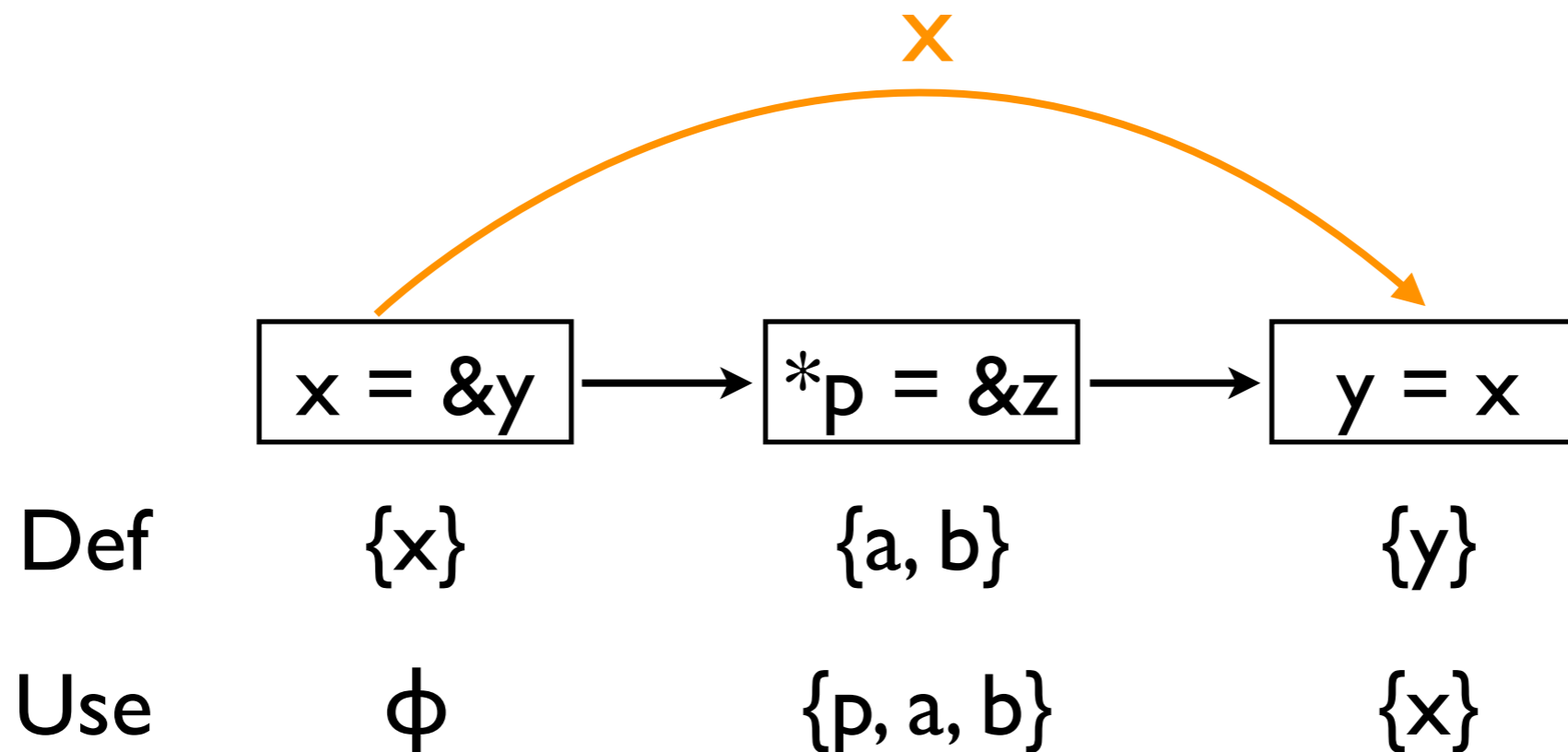
$$D(c) \triangleq \{l \in \hat{\mathbb{L}} \mid \exists \hat{s} \sqsubseteq \bigsqcup_{c' \hookrightarrow c} (\text{fix } \hat{F})(c'). \hat{f}_c(\hat{s})(l) \neq \hat{s}(l)\}.$$

$$U(c) \triangleq \{l \in \hat{\mathbb{L}} \mid \exists \hat{s} \sqsubseteq \bigsqcup_{c' \hookrightarrow c} (\text{fix } \hat{F})(c'). \hat{f}_c(\hat{s})|_{D(c)} \neq \hat{f}_c(\hat{s} \setminus l)|_{D(c)}\}.$$

Preserving

$$\text{fix } \hat{F} = \text{fix } \hat{F}_s \quad \text{modulo } D$$

Data Dependency 예제



실현가능한 스파스 분석

$$\hat{F}_a(\hat{X}) = \lambda c \in \mathbb{C} \cdot \hat{f}_c \left(\bigsqcup_{c' \overset{l}{\rightsquigarrow}_a c} \hat{X}(c') | l \right).$$

Realizable Data Dependency

$$c_0 \overset{l}{\rightsquigarrow}_a c_n \triangleq \exists c_0 \dots c_n \in \text{Paths}, l \in \hat{\mathbb{L}}. \\ l \in \hat{D}(c_0) \cap \hat{U}(c_n) \wedge \forall i \in (0, n). l \notin \hat{D}(c_i)$$

Preserving

$$\text{fix } \hat{F} \overset{\text{still}}{=} \text{fix } \hat{F}_a \quad \text{modulo } \hat{D}$$

If the following two conditions hold

\hat{D} & \hat{U} 이 만족해야 하는 조건

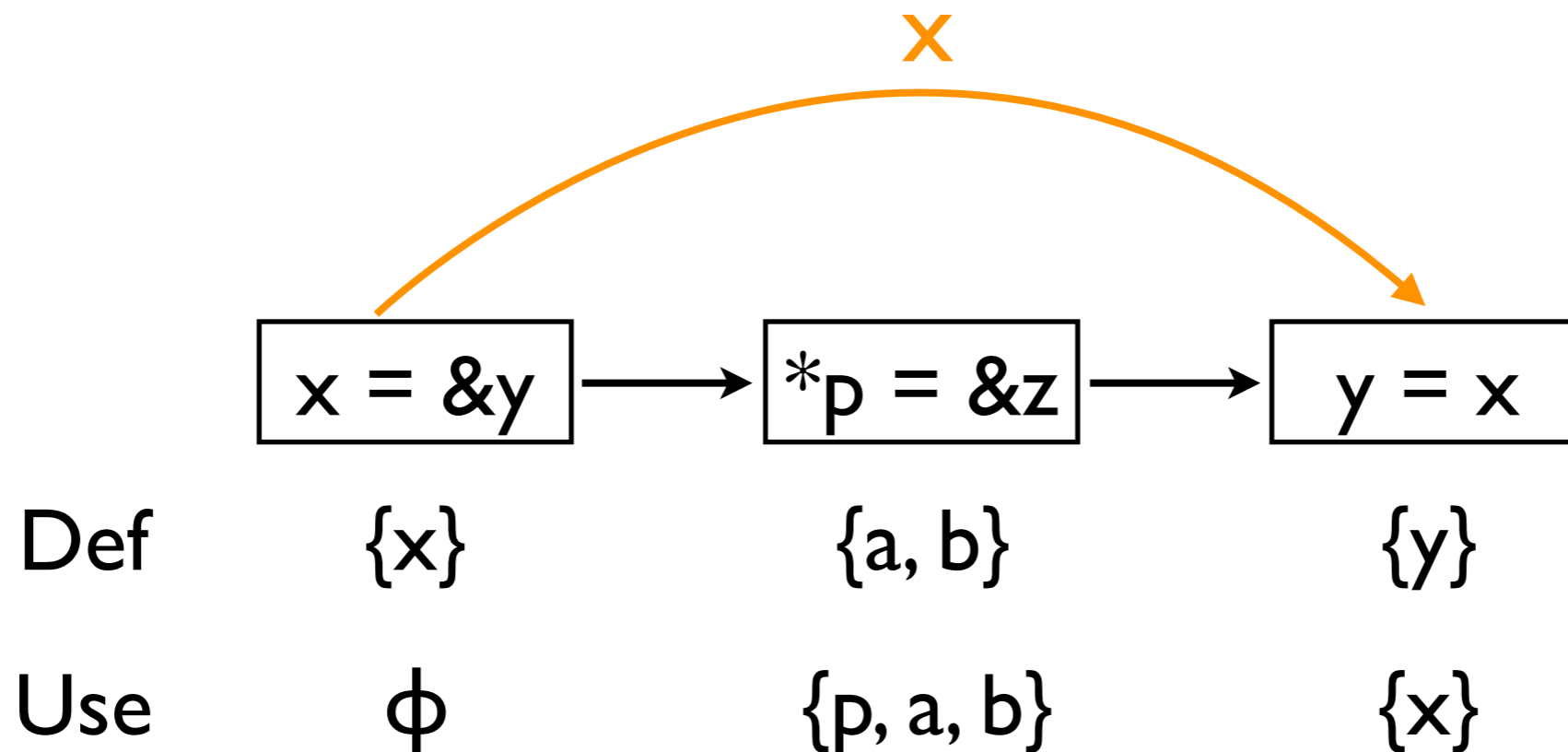
- 실제를 모두 포섭

$$\hat{D}(c) \supseteq D(c) \wedge \hat{U}(c) \supseteq U(c)$$

- 어림잡은 가짜 definition들을 특별히 처리

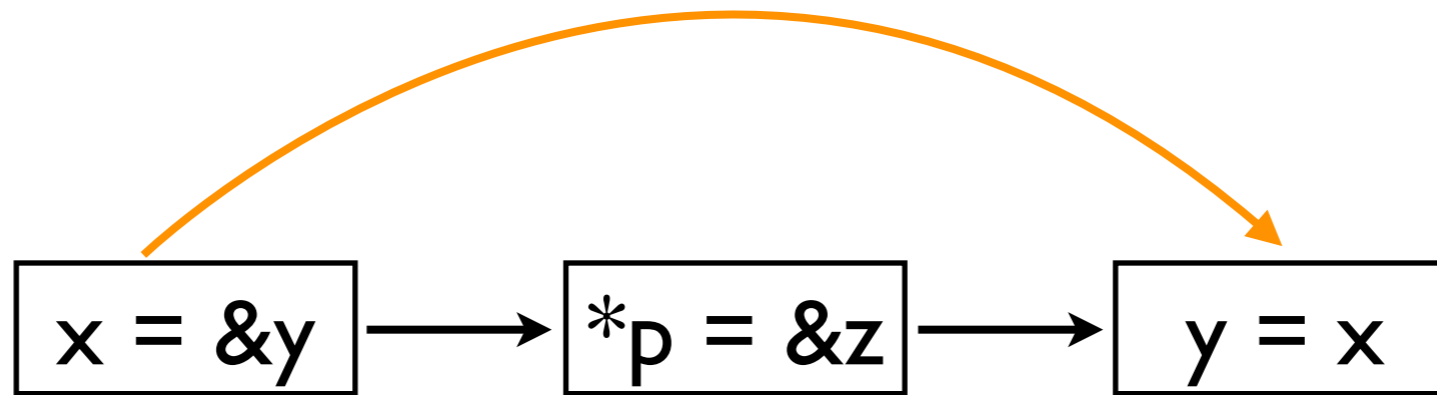
$$\hat{D}(c) - D(c) \subseteq \hat{U}(c)$$

두번째 조건이 필요한 이유



두번째 조건이 필요한 이유

x



Approx. Def

{x}

{a, b, x}

{y}

Approx. Use

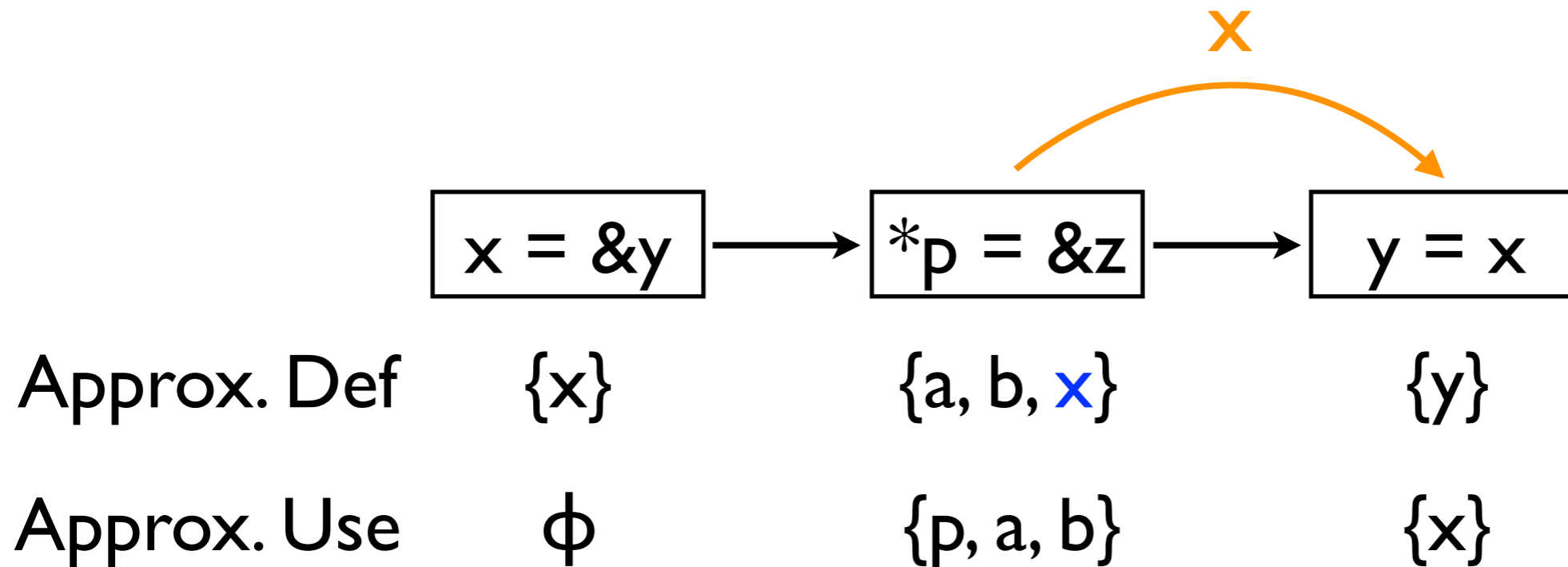
ϕ

{p, a, b}

{x}

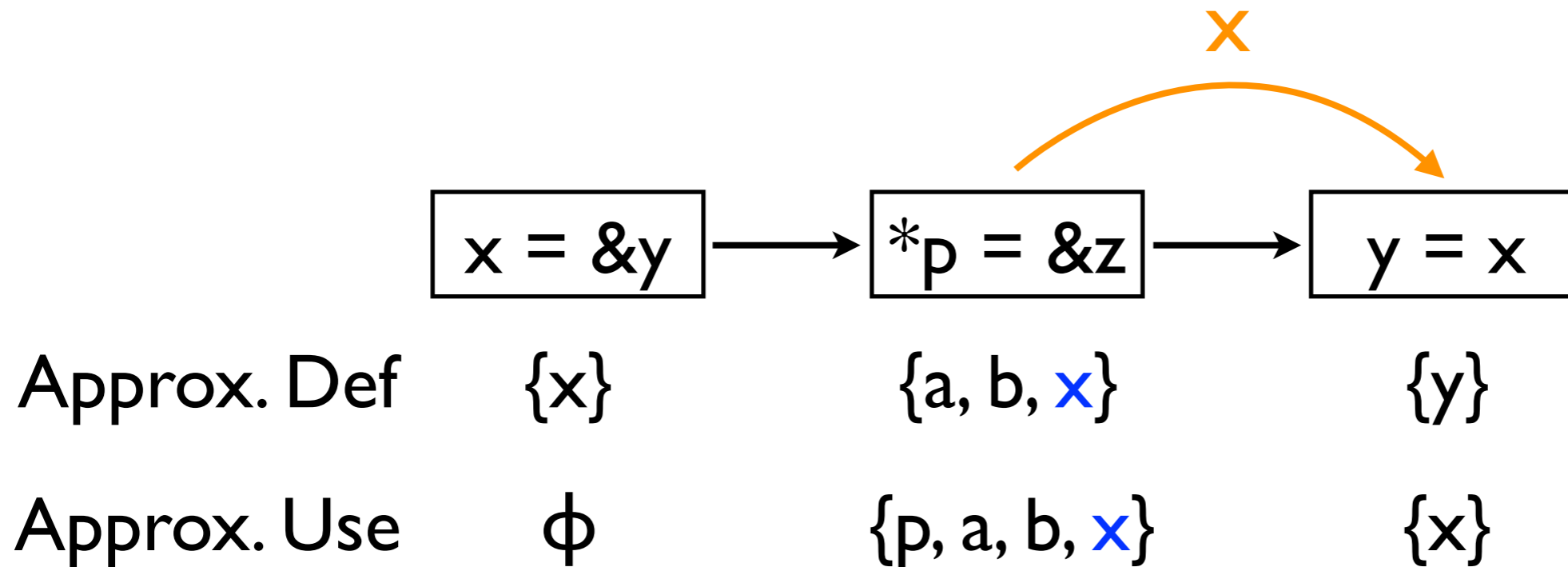
$$\frac{\hat{D}(c) - D(c)}{\{x\}} \not\subseteq \hat{U}(c)$$

두번째 조건이 필요한 이유



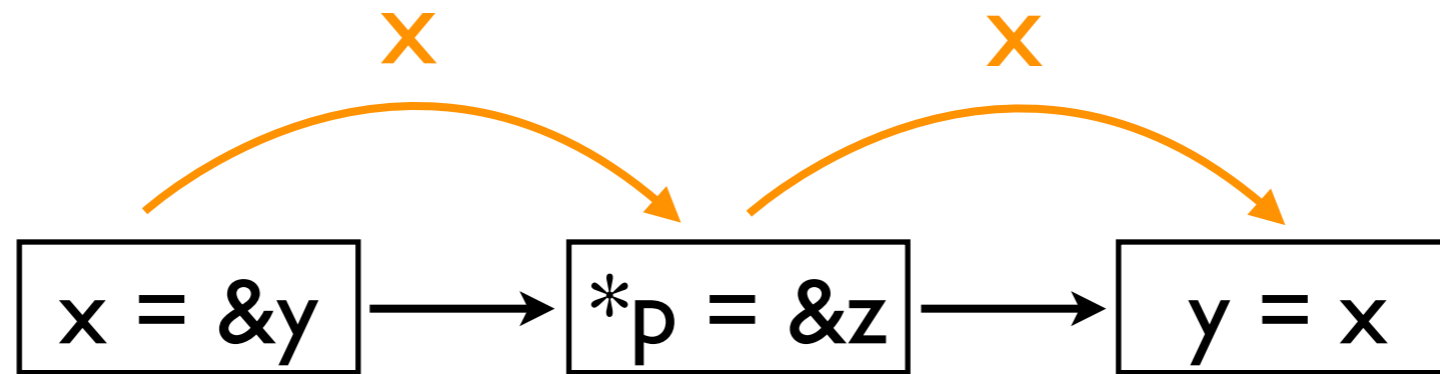
$$\frac{\hat{D}(c) - D(c)}{\{x\}} \not\subseteq \hat{U}(c)$$

두번째 조건이 필요한 이유



$$\frac{\hat{D}(c) - D(c)}{\{x\}} \subseteq \hat{U}(c)$$

두번째 조건이 필요한 이유



Approx. Def	{x}	{a, b, x}	{y}
Approx. Use	ϕ	{p, a, b, x}	{x}

$$\frac{\hat{D}(c) - D(c)}{\{x\}} \subseteq \hat{U}(c)$$

Performance

구현

- 안전한 버전의  위에서 구현

- 스파스 인터벌 분석

$$\hat{\mathcal{S}} = \text{AbsLoc} \rightarrow \text{Interval}$$

- 스파스 옥타곤 분석

$$\hat{\mathcal{S}} = \text{Packs} \rightarrow \text{Octagon}$$

성능 비교: 인터벌 분석

Program	LOC	Non-sparse		Sparse		Spd↑	Mem↓
		Time	Mem	Time	Mem		
gzip-1.2.4a	7 K	772	240	3	63	257 x	74 %
bc-1.06	13 K	1,270	276	7	75	181 x	73 %
less-382	23 K	9,561	1,113	33	127	289 x	86 %
make-3.76.1	27 K	24,240	1,391	21	114	1,154 x	92 %
wget-1.9	35 K	44,092	2,546	11	85	4,008 x	97 %
a2ps-4.14	64 K	∞	N/A	40	353	N/A	N/A
sendmail-8.13.6	130 K	∞	N/A	744	678	N/A	N/A
nethack-3.3.0	211 K	∞	N/A	16,373	5,298	N/A	N/A
emacs-22.1	399 K	∞	N/A	37,830	7,795	N/A	N/A
python-2.5.1	435 K	∞	N/A	11,039	5,535	N/A	N/A
linux-3.0	710 K	∞	N/A	33,618	20,529	N/A	N/A
gimp-2.6	959 K	∞	N/A	3,874	3,602	N/A	N/A
ghostscript-9.00	1,363 K	∞	N/A	14,814	6,384	N/A	N/A

성능 비교: 인터벌 분석

Program	LOC	Non-sparse		Sparse		Spd↑	Mem↓
		Time	Mem	Time	Mem		
gzip-1.2.4a	7 K	772	240	3	63	257 x	74 %
bc-1.06	13 K	1,270	276	7	75	181 x	73 %
less-382	23 K	9,561	1,113	33	127	289 x	86 %
make-3.76.1	27 K	24,240	1,391	21	114	1,154 x	92 %
wget-1.9	35 K	44,092	2,546	11	85	4,008 x	97 %
a2ps-4.14	64 K	∞	N/A	40	353	N/A	N/A
sendmail-8.13.6	130 K	∞	N/A	744	678	N/A	N/A
nethack-3.3.0	211 K	∞	N/A	16,373	5,298	N/A	N/A
emacs-22.1	399 K	∞	N/A	37,830	7,795	N/A	N/A
python-2.5.1	435 K	∞	N/A	11,039	5,535	N/A	N/A
linux-3.0	710 K	∞	N/A	33,618	20,529	N/A	N/A
gimp-2.6	959 K	∞	N/A	3,874	3,602	N/A	N/A
ghostscript-9.00	1,363 K	∞	N/A	14,814	6,384	N/A	N/A

성능 비교: 옥타곤 분석

Program	LOC	Non-sparse		Sparse		Spd↑	Mem↓
		Time	Mem	Time	Mem		
gzip-1.2.4a	7 K	2,078	2,832	21	269	98 x	91 %
bc-1.06	13 K	9,536	6,987	55	358	173 x	95 %
tar-1.13	20 K	∞	N/A	188	526	N/A	N/A
less-382	23 K	∞	N/A	432	458	N/A	N/A
make-3.76.1	27 K	∞	N/A	331	666	N/A	N/A
wget-1.9	35 K	∞	N/A	288	646	N/A	N/A
screen-4.0.2	45 K	∞	N/A	16,433	9,199	N/A	N/A
a2ps-4.14	64 K	∞	N/A	8,546	1,996	N/A	N/A
sendmail-8.13.6	130 K	∞	N/A	64,808	29,658	N/A	N/A

마무리

안전성과 정확성을 유지하면서 대형 프로그램을 고속 분석하는 방법

- 요약해석으로 베이스라인 분석을 디자인
 - **sound, precise, unscalable**
- 스파스 분석으로 변환
 - **scalable, preserving soundness & precision**

진행중, 향후 계획

- 프레임워크 확장
- 실행과정 분할(trace partitioning), 함수형/OOP 지원
- 스파스 분석 생성기
- 요약공간/실행함수 정의로부터 스파스 분석 자동 생성

Thank you