정확한 정수 합동 분석을 위한 역방향 요약 연산자 정의*

서선애[O]
한국과학기술원 전자전산학과 / 프로그램 분석 시스템 연구단[†]
saseo@ropas.kaist.ac.kr

Backward Abstract Arithmetic Operations for Integer Congruence Analysis

Sunae Seo[O]
Dept. of EECS / Research On Program Analysis System, KAIST

요 약

정수 합동 분석(integer congruence analysis)은 프로그램 변수들의 의미 영역을 정수 합동 (integer congruence) 집합으로 정의하여 분석한다. 정수 합동 분석을 위한 정수 합동 격자(lattice of integer congruences)와 순방향 요약 산술 연산자에 대한 정의는 이미 P. Granger에 의해 소개되었다. 하지만, 분석의 정확도에 영향을 미치는 역방향 요약 산술 연산자에 대한 연구는 아직 되어 있지 않다.

이 논문에서는 정수 합동 분석을 위한 역방향 요약 산술 연산자를 정의한다. 역방향 요약 산술 연산자를 정의하는 방법은 정수 방정식을 푸는 방법을 기반으로 고안되었다. 정의된 역방향 요약 산술 연산자는 프로그램 분석의 정확도를 높이는데 기여를 할 수 있는데, 이 논문에서는 예제를 통해서 이 사실을 보인다.

1 서론

정수 합동 분석(integer congruence analysis)이란 정수 변수를 포함하는 프로그램의 계산 결과를 정수 합동(integer congruence)의 형태로 예측하는 분석이다. 즉, 프로그램에 어떤 정수 변수 x 가 포함될 때, 정수 합동 분석은 변수 x가 프로그램 내에서 가질 수 있는 값을 정수 합동 집합으로 계산한다. 따라서, 프로그램 내의 모든 정수 변수들은 $c+m\mathbb{Z}$ 형태의 값을 가지게 된다(단, c와 m은 주어진 정수이다). $c+m\mathbb{Z}$ 은 m 으로 나누었을 때, c 가 남는 모든 정수의 집합을 말한다. 예를 들어, $1+3\mathbb{Z}$ 은 3 으로 나누었을 때 1 이 남는 정수들의 집합이다. $-2+3\mathbb{Z}$ 도 3 으로 나누었을 때 1 이 남는 정수의 집합이지만, 논문에서는 $0 \le c < m$ 와 $0 < m$ 을 가정한다. $m=0$ 은 원소가 c 하나뿐인 집합 $\{c\}$ 을 의미한다.

P. Granger [Gra89] 는 요약 해석(abstract interpretation) [CC77]을 기반으로 하는 정수 합동 분석에 대해서 연구를 하였다. 정수 합동 분석을 위해서 분석 결과의 영역인 합동 격자(lattice of congruences)와 합동 격자 요소 간의 연산자들이 정의되었다. 또한, 프로그램에서 정수 산술 연산자(integer arithmetic operator)들이 사용되므로, P. Granger 는 요약된 정수 산술 연산자들을 정의하였다. 이 요약된 정수 산술 연산자는 피연산자가 주어졌을 때 연산의 결과를 예측하는데, 우리는 이런 산술 연산자를 순방향 요약 산술 연산자(forward abstract arithmetic operator)라고 한다.

순방향 요약 산술 연산자와 비교해서, 결과가 주어졌을 때 그 결과를 갖도록 하는 피연자들을 예측하는 것을 역방향 요약 산술 연산자(backward abstract arithmetic operator)라고 한다. 순방향 산술 연산자가 프로그램의 수행 결과를 예측하는 반면, 역방향 산술 연산자는 주어진 수행 결과를 가능하게 하는 입력을 예측하므로 분석의 정확도를 높인다. 예를 들어, 다음 프로그램에서 $code_1(x)$ 은 $condition(x)$ 이 참일때만 수행이 되므로, 분석기는 $condition(x)$ 이 참이 되게 하는 x의 값만을 예측하여 $code_1(x)$ 에서 이용하면 된다.

... x ...;
if $condition(x)$ **then** $code_1(x)$ **else** $code_2(x)$

*본 연구는 저자가 프랑스 Ecole Polytechnique 에 교환 학생으로 방문시 연구한 내용으로, 과학기술부 창의적 연구진흥사업과 프랑스 CNRS(http://www.cnrs.fr)의 지원에 의한 결과임.

†http://ropas.kaist.ac.kr

```
x:=4; y:=10; z:=8; i:=?;
while i=0 do
    x := x+2; y := y+7; z := z+6
od;
if y * z = 6 then
    if y mod 3 = 0 then
        x := y*z
    else
        if y mod 3 = 1 then
            x := y*z + 2
        else
            x := y*z + 2
        fi
    fi
else
    x := y
fi
```

그림 1: 예제 프로그램

이 연구에서 우리는 정수 합동 분석을 위한 역방향 요약 산술 연산자를 정의한다. 우리가 정의한 역방향 요약 산술 연산자는 정수 방정식을 푸는 방법을 정수 합동 방정식을 푸는 것에 적용하여 고안되었다. 정의된 역방향 요약 산술 연산자는 분석 결과의 정확도를 높이는데 그것을 예제를 통해서 보인다.

그림 1의 프로그램을 역방향 요약 산술 연산자가 없는 분석기를 이용해서 분석을 하면, 그림 2와 같다. 우리는 이 분석기를 P. Cousot 의 Marktoberdorf 분석기 [Cou99] 를 바탕으로 제작하였다. 각 코드의 아랫 부분에 {} 괄호로 묶여 있는 것은 프로그램의 환경을 의미한다. 예를 들어, $\{x:[0,2]; y:[3,7]\}$는 x 의 값이 $2\mathbb{Z}$ 에, y 의 값이 $3+7\mathbb{Z}$ 에 포함된다는 것을 말한다. 같은 예제를 우리의 연산자로 분석한 결과를 3장에서 소개하고 있다.

2 역방향 요약 산술 연산자의 정의

이 장에서는 역방향 요약 산술 연산자들을 안전하게 정의한다. 이 장에서 다루는 산술 연산자는 두 단항 연산자와 덧셈, 뺄셈, 곱셈, 나눗셈 그리고 나머지(modular) 연산자이다.

단항 역방향 요약 산술 연산자는 두 개의 피연산자를 갖는다. 첫번째 피연산자는 산술 연산의 입력(p)을 의미하고, 두번째 피연산자는 산술 연산의 결과(q)를 의미한다. 그래서, 역방향 산술 연산자가 생성하는 값은 산술 연산의 결과 q가 되도록 하는 예상 입력 중에서 p 에 포함되는 값이다.

	$c=0, m=0$	$c=0, m\neq 0$	$c\neq 0, m=0$	$c\neq 0, m\neq 0$
$c_i=0, m_i=0$	$\mathbb{Z}$	$\mathbb{Z}$	$\{\}$	$\{\}$
$c_i=0, m_i\neq 0$	$\mathbb{Z}$	$\mathbb{Z}$	if $m_i \mid c$ then if c/m_i is odd then $1+2\mathbb{Z}$ else $\mathbb{Z}$ else $\{\}$	if $g.c.d.(m, m_i) \mid c$ then let $c'+m'\mathbb{Z} = \{x \mid xm_i \in c+m\mathbb{Z}\}_c$ in if c' is odd and m' is even then $1+2\mathbb{Z}$ else $\mathbb{Z}$ else $\{\}$
$c_i\neq 0, m_i=0$	$\{0\}$	$\{x \mid xc_i \equiv 0[m]\}_c$	if $c_i \mid c$ then $\{c/c_i\}$ else $\{\}$	$\{x \mid xc_i \equiv c[m]\}_c$
$c_i\neq 0, m_i\neq 0$	$\{0\}$	$\{x \mid xc_i \equiv c[g.c.d.(m, m_i)]\}_c$		

표 1: $\{x \mid \exists e \in c_i + m_i\mathbb{Z}, e' \in c+m\mathbb{Z}.\ e \times x = e'\}_c$ 을 포함하는 정수 합동 집합들.

```
{x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
if y * z = 6 then
{x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
    if y mod 3 = 0 then
    {x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
        x := y*z
        {x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
    else
    {x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
        if y mod 3 = 1 then
        {x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
            x := y*z + 2
            {x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
        else
        {x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
            x := y*z + 2
            {x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
        fi
        {x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
    fi
    {x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
else
{x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
    x := y
    {x : [3,7]; y : [3,7]; z : [2,6]; i : [0,1]}
fi
{x : [0,1]; y : [3,7]; z : [2,6]; i : [0,1]}
```

그림 2: 역방향 요약 연산자를 이용하지 않은 분석 결과

정의 1 (역방향 요약 단항 연산자) *두 정수 합동 집합 $c_1+m_1\mathbb{Z}$ 와 $c+m\mathbb{Z}$ 에 대해서,*

$$+^{\triangleleft}_{u}(c_1+m_1\mathbb{Z}, c+m\mathbb{Z}) \triangleq (c_1+m_1\mathbb{Z}) \cap (c+m\mathbb{Z})$$
$$-^{\triangleleft}_{u}(c_1+m_1\mathbb{Z}, c+m\mathbb{Z}) \triangleq (c_1+m_1\mathbb{Z}) \cap (-c+m\mathbb{Z}).$$

다항 역방향 요약 산술 연산자는 세 개의 피연산자를 갖는데, 앞의 두 피연산자는 산술 연산의 입력(p_1 과 p_2)을 의미하고, 세번째 피연산자는 산술 연산의 결과(q)를 의미한다. 역방향 산술 연산자가 생성하는 값은 p_1 과 p_2 의 값 중에서 산술 연산을 하였을 때, q 를 생성하는 값들이다. 즉, 다항 역방향 요약 산술 연산자는 두 개의 결과를 생성한다.

정의 2 (역방향 요약 덧셈과 뺄셈 연산자) *세 정수 합동 집합 $c_1+m_1\mathbb{Z}, c_2+m_2\mathbb{Z}$ 와 $c+m\mathbb{Z}$ 에 대하여,*

$$\begin{aligned}&+^{\triangleleft}_{b}(c_1+m_1\mathbb{Z}, c_2+m_2\mathbb{Z}, c+m\mathbb{Z})\\ &\triangleq \langle((c-c_2)+(g.c.d.(m,m_2))\mathbb{Z}) \cap (c_1+m_1\mathbb{Z}),\\ &\quad ((c-c_1)+(g.c.d.(m,m_1))\mathbb{Z}) \cap (c_2+m_2\mathbb{Z})\rangle\end{aligned}$$

$$\begin{aligned}&-^{\triangleleft}_{b}(c_1+m_1\mathbb{Z}, c_2+m_2\mathbb{Z}, c+m\mathbb{Z})\\ &\triangleq \langle((c+c_2)+(g.c.d.(m,m_2))\mathbb{Z}) \cap (c_1+m_1\mathbb{Z}),\\ &\quad ((c_1-c)+(g.c.d.(m,m_1))\mathbb{Z}) \cap (c_2+m_2\mathbb{Z})\rangle\end{aligned}$$

정의 3 (역방향 요약 곱셈 연산자) *세 정수 합동 집합 $c_1+m_1\mathbb{Z}, c_2+m_2\mathbb{Z}$ 와 $c+m\mathbb{Z}$ 에 대하여,*

$$\begin{aligned}&\times^{\triangleleft}(c_1+m_1\mathbb{Z}, c_2+m_2\mathbb{Z}, c+m\mathbb{Z})\\ &\triangleq \langle(c_1+m_1\mathbb{Z}) \cap C_{i=2},\ (c_2+m_2\mathbb{Z}) \cap C_{i=1}\rangle.\end{aligned}$$

이때, $C_{i=j} (j \in \{1,2\})$ 는 두 입력 합동 집합 $c+m\mathbb{Z}$ 와 $c_j+m_j\mathbb{Z}$ 에 대해서 정의된 표 1을 말한다.*

다른 연산자들과 달리, 나눗셈 연산자는 좋은 역방향 요약 연산자를 정의하는 것이 어렵다. 정수 나눗셈의 성질은 좋은 연산자를 정의하기 어렵게 하기 때문이다. 예를 들어서, 정수 방정식 $x \div 6 = 5$ 의 x 에 대한 해는 30, 31, 32, 33, 34, 35의 5 가지인데, 이 5 가지 숫자를 포함하는 가장 작은 정수 합동 집합은 정수 전체 집합($\mathbb{Z}$) 뿐이다. 또한, 주어진 상수 a 와 c 에 대하여, 정수 방정식 $a \div x = c$ 의 해를 구하는 방법이 없기 때문에, 두번째 피연산자에 대해서도 좋은 결과를 얻을 수 없다. 단지, 우리가 알 수 있는 것은 $|c| > |a|$ 이면, x 에 대한 해가 없다는 것이다. 이것을 정수 합동 집합으로 확장하면 다음과 같은 식을 얻을 수 있다.

- $m > 0$, $M = c+m((|d|-c)\div m)$, $N = c+m((-|d|-c)\div m)$ 와 $C_x = \{x \mid \exists e \in (c+m\mathbb{Z}).\ d \div x = e\}_c$ 에 대해서,

  만일 $M < -|d|$ 이거나 $N > |d|$ 이면, $C_x = \{\}$.

M 은 $|d|$ 보다 작거나 같은 $c+m\mathbb{Z}$ 내의 가장 큰 정수이고, N 은 $-|d|$ 보다 크거나 같은 $c+m\mathbb{Z}$ 내의 가장 작은 정수이다.

정의 4 (역방향 요약 나눗셈 연산자) *세 정수 합동 집합 $c_1+m_1\mathbb{Z}, c_2+m_2\mathbb{Z}(\neq \{0\})$ 와 $c+m\mathbb{Z}$ 에 대하여,*

- $\pi_1(\div^{\triangleleft}(c_1+m_1\mathbb{Z}, c_2+m_2\mathbb{Z}, c+m\mathbb{Z}))$ *은 $c_2+m_2\mathbb{Z}$ 의 각 경우에 따라*

  $$\begin{array}{lcl}\{1\} & \rightarrow & (c_1+m_1\mathbb{Z}) \cap (c+m\mathbb{Z})\\ \{-1\} & \rightarrow & (c_1+m_1\mathbb{Z}) \cap ((-c)+m\mathbb{Z})\\ \textit{otherwise} & \rightarrow & c_1+m_1\mathbb{Z}\end{array}$$

- $\pi_2(\div^{\triangleleft}(\{d\}, c_2+m_2\mathbb{Z}, c+m\mathbb{Z}))$ *은*
  만일 $M < -|d|$ 이거나 $N > |d|$ 면, $\{\}$
  그렇지 않으면, $c_2+m_2\mathbb{Z}$.

- *그밖에 $\pi_2(\div^{\triangleleft}(c_1+m_2\mathbb{Z}, c_2+m_2\mathbb{Z}, c+m\mathbb{Z}))$ 은 $c_2+m_2\mathbb{Z}$.*

위의 정의에서 π 는 두 원소를 갖는 짝을 받아서 짝의 첫번째 원소 (π_1), 혹은 짝은 두번째 원소 (π_2)를 넘겨주는 함수이다.

정의 5 (역방향 요약 나머지 연산자) *세 정수 합동 집합 $(c_1+m_1\mathbb{Z}), (c_2+m_2\mathbb{Z})(\neq \{0\})$ 와 $(c+m\mathbb{Z})$ 에 대하여,*

- *만일 $c-c_1$ 이 홀수이고, m_1, m 이 짝수이면,*

  $$\begin{aligned}&\mathbf{mod}^{\triangleleft}((c_1+m_1\mathbb{Z}), (c_2+m_2\mathbb{Z}), (c+m\mathbb{Z}))\\ &\triangleq \langle(c_1+m_1\mathbb{Z}) \cap C_2,\ (c_2+m_2\mathbb{Z}) \cap (1+2\mathbb{Z})\rangle.\end{aligned}$$

- *그렇지 않으면,*

  $$\begin{aligned}&\mathbf{mod}^{\triangleleft}((c_1+m_1\mathbb{Z}), (c_2+m_2\mathbb{Z}), (c+m\mathbb{Z}))\\ &\triangleq \langle(c_1+m_1\mathbb{Z}) \cap C_2,\ (c_2+m_2\mathbb{Z}) \cap \mathbb{Z}\rangle.\end{aligned}$$

단, C_x 는 표 2에 정의된 것이다.

*집합 기호 $\{\}_c$ 는 정수 합동 집합을 말한다. 예를 들어, $\{1,3\}_c$은 1과 3을 포함하는 가장 작은 정수 합동 집합, 즉 $1+2\mathbb{Z}$를 의미한다.

Cases	Definitions
$m = m_2 = 0$	if $0 \leq c < c_2$ then $c + c_2\mathbb{Z}$ else if $c_2 < c \leq 0$ then $c + c_2\mathbb{Z}$ else $\{\}$

$m \neq 0, m_2 = 0$	if $0 < c_2$ then
	let $A_+ = \{\lambda \in \mathbb{Z} \mid 0 \leq c + \lambda m < c_2\}_c$ in
	case $\lvert A_+ \rvert$ of
	$0 \to \{\}$
	$1 \to (c + \lambda m) + c_2\mathbb{Z}$, where $\lambda \in A_+$
	otherwise $\to c + (g.c.d.(m, c_2))\mathbb{Z}$
	if $c_2 < 0$ then
	let $A_- = \{\lambda \in \mathbb{Z} \mid c_2 < c + \lambda m \leq 0\}_c$ in
	case $\lvert A_- \rvert$ of
	$0 \to \{\}$
	$1 \to (c + \lambda' m) + c_2\mathbb{Z}$, where $\lambda' \in A_-$
	otherwise $\to c + (g.c.d.(m, c_2))\mathbb{Z}$

$m = 0, m_2 \neq 0$	$c + (g.c.d.(c_2, m_2))\mathbb{Z}$
$m \neq 0, m_2 \neq 0$	$c + (g.c.d.(m, c_2, m_2))\mathbb{Z}$

표 2: $\{x \mid \exists e \in c_2 + m_2\mathbb{Z}, \exists e' \in c + m\mathbb{Z}.\ x \textbf{ mod } e = e'\}_c$ 를 포함하는 정수 합동 집합들.

정의된 요약 연산자들의 안전성은 기존의 정수 산술 연산자의 연산 결과를 요약 연산자들이 포함하는지를 보이는 것으로 증명이 된다.

정리 1 (연산자들의 안전성)
정의된 역방향 요약 산술 연산자들은 원래의 산술 연산자에 대해서 안전하다.

$$\begin{array}{rcl} +^{\triangleleft}_{u}(c_1 + m_1\mathbb{Z}, c + m\mathbb{Z}) & = & \{i \in c_1 + m_1\mathbb{Z} \mid i \in c + m\mathbb{Z}\}_c \\ -^{\triangleleft}_{u}(c_1 + m_1\mathbb{Z}, c + m\mathbb{Z}) & = & \{i \in c_1 + m_1\mathbb{Z} \mid (-i) \in c + m\mathbb{Z}\}_c. \end{array}$$

세 정수 합동 집합 q_1, q_2 와 p 에 대하여†,

$$\begin{array}{rcl} +^{\triangleleft}_{b}(q_1, q_2, p) & = & \mathbf{nr}(\{\langle i_1,\ i_2\rangle \in \langle q_1,\ q_2\rangle \mid i_1 + i_2 \in p\}) \\ -^{\triangleleft}_{b}(q_1, q_2, p) & = & \mathbf{nr}(\{\langle i_1,\ i_2\rangle \in \langle q_1,\ q_2\rangle \mid i_1 - i_2 \in p\}) \\ \times^{\triangleleft}(q_1, q_2, p) & \supseteq & \mathbf{nr}(\{\langle i_1,\ i_2\rangle \in \langle q_1,\ q_2\rangle \mid i_1 \times i_2 \in p\}) \\ \div^{\triangleleft}(q_1, q_2, p) & \supseteq & \mathbf{nr}(\{\langle i_1,\ i_2\rangle \in \langle q_1,\ q_2\rangle \mid i_1 \div i_2 \in p\}) \\ \mathbf{mod}^{\triangleleft}(q_1, q_2, p) & \supseteq & \mathbf{nr}(\{\langle i_1,\ i_2\rangle \in \langle q_1,\ q_2\rangle \mid i_1 \textbf{ mod } i_2 \in p\}). \end{array}$$

3 예제

그림 3은 정의된 역방향 요약 연산자를 이용한 분석 결과이다. 역방향 요약 연산자를 이용하면, 분석기는 line 5 의 `if` 문 내부에서 변수 y 는 합동 집합 $3 + 21\mathbb{Z}$ 에 속한다고 분석한다. $3 + 21\mathbb{Z}$ 의 모든 숫자들은 3의 배수이므로, 변수 y 가 3의 배수가 아닐 때 수행되는 line 9 에서 line 13 까지의 코드는 실행되지 않는 코드(dead-code)이다. 우리의 역방향 요약 연산자를 사용한 결과는 이 실행되지 않는 코드를 찾는다: 그림 3에서 프로그램의 line 9 에서 line 13 까지의 환경 $\{x : \bot; y : \bot; z : \bot; i : \bot\}$ 은 실행되지 않는 코드임을 의미한다. 반면에, 역방향 요약 연산자를 사용하지 않는 분석기(그림 2)는 line 5 의 `if` 문 내부에서 변수 y 가 합동 집합 $3 + 7\mathbb{Z}$ 의 값을 갖는다고 하여, 실행되지 않는 코드가 있음을 찾아내지 못한다.

4 결론

이 연구에서는 정수 변수 기반 프로그램 분석을 위한 정수 합동 분석의 역방향 요약 산술 연산자를 정의하였다. 또한, 정의된 역방향 요약 산술 연산자가 산술 연산의 원래 의미에 대해 안전함을 증명하였다. 역방향 요약 산술 연산자는 특히 `if` 문이나 `while` 문과 같은 조건문에서 분석의 정확도를 높인다.

정의된 역방향 요약 산술 연산자는 대부분 가장 작은 정수 합동 집합을 결과로 내 준다. 곱셈이나, 나눗셈 그리고 나머지 연산자의 일부는 가장 작은 결과를 주지 못하지만, 가장 작은 결과를 포함하는 결과를 내 주기 때문에 안전하다.

† nr 은 짝을 원소로 하는 집합을 짝의 첫 원소끼리 묶고, 둘째 원소끼리 묶어서, 두 개의 집합으로 나누는 함수이다. 엄밀하게는 다음과 같이 정의된다 : $\mathbf{nr}(p) \triangleq \langle\{x \mid \exists y.\ \langle x,\ y\rangle \in p\}_c,\ \{y \mid \exists x.\ \langle x,\ y\rangle \in p\}_c\rangle$.

```
                {x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
line 5:         if y * z = 6 then
                {x : [0,2]; y : [3,21]; z : [2,42]; i : [0,1]}
line 6:            if y mod 3 = 0 then
                   {x : [0,2]; y : [3,21]; z : [2,42]; i : [0,1]}
line 7:               x := y*z
                      {x : [6,42]; y : [3,21]; z : [2,42]; i : [0,1]}
line 8:            else
                   {x : ⊥; y : ⊥; z : ⊥; i : ⊥}
line 9:               if y mod 3 = 1 then
                      {x : ⊥; y : ⊥; z : ⊥; i : ⊥}
line 10:                 x := y*z + 2
                         {x : ⊥; y : ⊥; z : ⊥; i : ⊥}
line 11:              else
                      {x : ⊥; y : ⊥; z : ⊥; i : ⊥}
line 12:                 x := y*z + 2
                         {x : ⊥; y : ⊥; z : ⊥; i : ⊥}
line 13:              fi
                      {x : ⊥; y : ⊥; z : ⊥; i : ⊥}
line 14:           fi
                   {x : [6,42]; y : [3,21]; z : [2,42]; i : [0,1]}
line 15:        else
                {x : [0,2]; y : [3,7]; z : [2,6]; i : [0,1]}
line 16:           x := y
                   {x : [3,7]; y : [3,7]; z : [2,6]; i : [0,1]}
line 17:        fi
                {x : [0,1]; y : [3,7]; z : [2,6]; i : [0,1]}
```

그림 3: 역방향 요약 연산자를 이용한 분석 결과

참고 자료

[CC77] P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Conference Record of the Fourth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 238–252, 1977.

[Cou99] P. Cousot. The calculational design of a generic abstract interpreter. In M. Broy and R. Steinbrüggen, editors, *Calculational System Design*. NATO ASI Series F. IOS Press, Amsterdam, 1999.

[Gra89] P. Granger. Static analysis of arithmetical congruences. In *International Journal of Computer Mathematics*, volume 30, pages 165–190. 1989.