Synthesis, Analysis, and Verification

Abstract Interpretation Ideas for Lecture 8

(see also wiki)

Lectures:

Viktor Kuncak

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

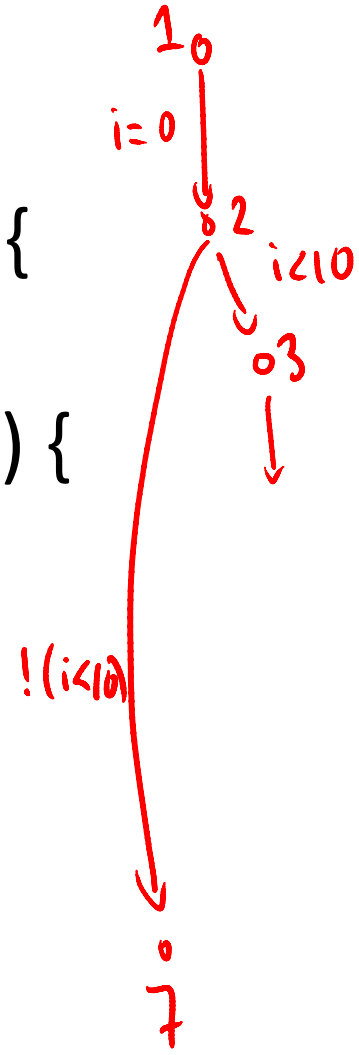
While Theorem: One Loop

Loop invariants become one invariant

```
(1)
  i = 0
(2)
  while (i < 10) {
(3)
    j = i + 1
(4)
    while (j < 10) {
(5)
      println(i,j)
(6)
      j = j + 1
    }
  }
(7)
```

I_2

I_4




```
while (pc != 7) {
  if (pc==1) { i=0; pc=2 }
  else if (pc==2 && i < 10) {
     pc=3 }
  else if (pc==2 && !(i < 10) {
     pc = 7 }
  else (
}
```

$\vee$ pc ∉ {1,2,3}

$(pc = 2 \wedge I_2) \vee (pc = 4 \wedge I_4)$

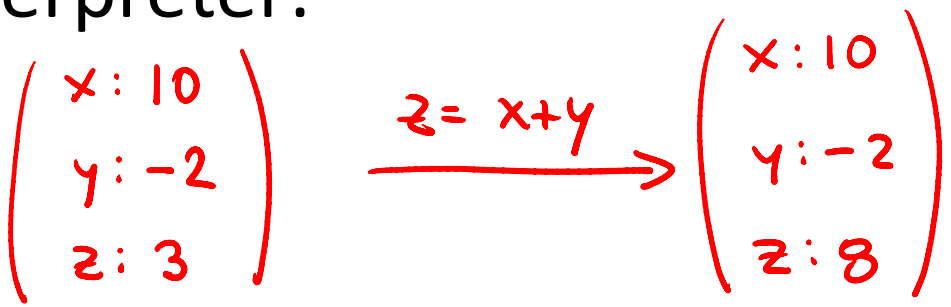
$(pc = 2 \rightarrow I_2) \wedge (pc = 4 \rightarrow I_4)$

Abstract Interpretation

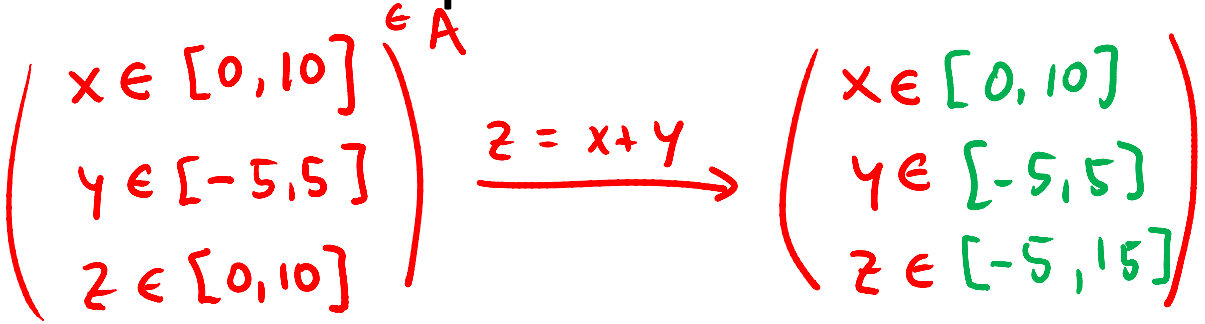
Way to infer properties of e.g. computations

Consider assignment: z = x+y

Interpreter:



Abstract interpreter:



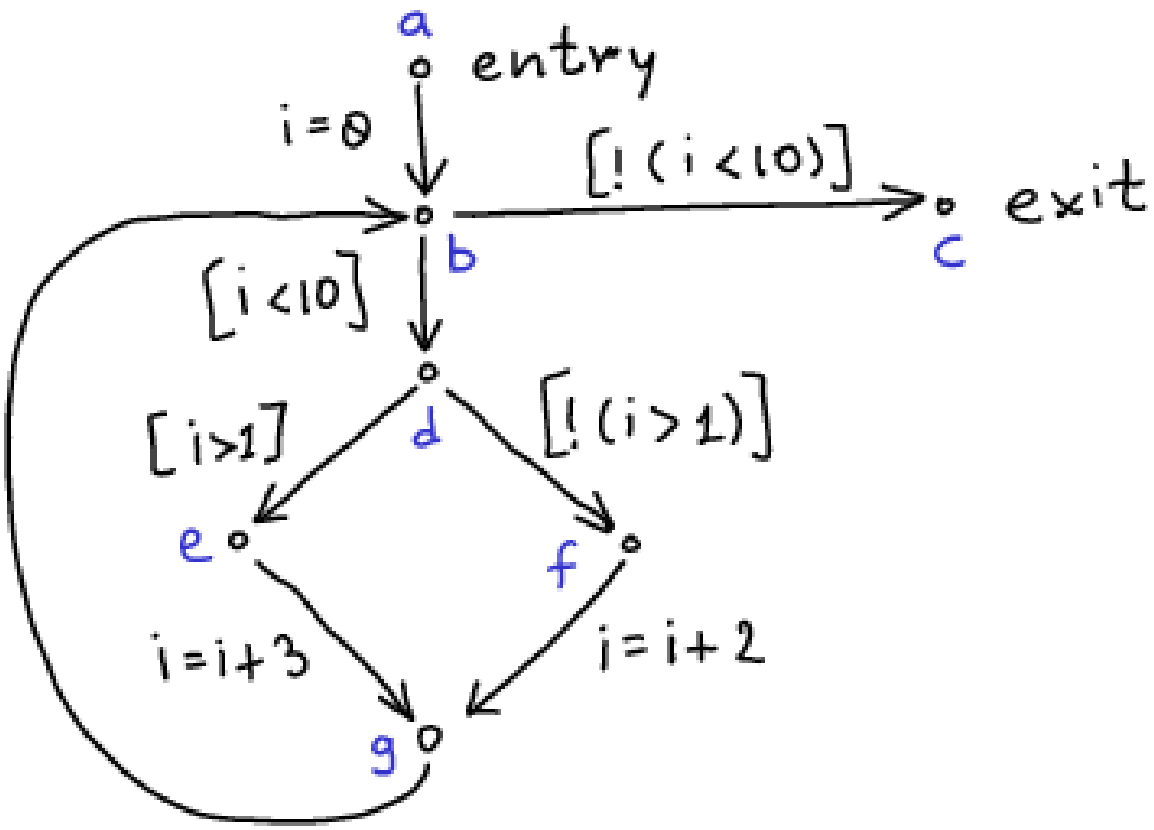
Adding and Multiplying Intervals

$$\begin{pmatrix} x \in [a_x, b_x] \\ y \in [a_y, b_y] \\ z \in \dots \end{pmatrix} \xrightarrow{z = x + y} \begin{pmatrix} x \in [a_x, b_x] \\ y \in [a_y, b_y] \\ z \in [a_x + a_y \ , \ b_x + b_y] \end{pmatrix}$$

$$\begin{pmatrix} x \in [a_x, b_x] \\ y \in [a_y, b_y] \\ z \in \dots \end{pmatrix} \xrightarrow{z = x * y} \begin{pmatrix} x \in [a_x, b_x] \\ y \in [a_y, b_y] \\ z \in [a_x * a_y \ , \ b_x * b_y] \quad a_x, a_y, b_x, b_y > 0 \\ B = \{ a_x \cdot a_y, a_x \cdot b_y, b_x \cdot a_y, b_x \cdot b_y \} \\ z \in [\min(B), \max(B)] \end{pmatrix}$$

Programs as Control-Flow Graphs

```
a
i = 0;
while(i < 10) {
b
  if (i > 1)
d
    i = i + 3;
e
  else
    i = i + 2;
f
}
g
```



- Suppose
  - program state given only by the value of i
  - initially, it is possible that i has any value
- Task: for each point, find set S of possible states

$S(a) = \{\ldots, -2, -1, 0, 1, 2, 3, \ldots\} = \mathbb{Z}$

$S(b) = \{0,$

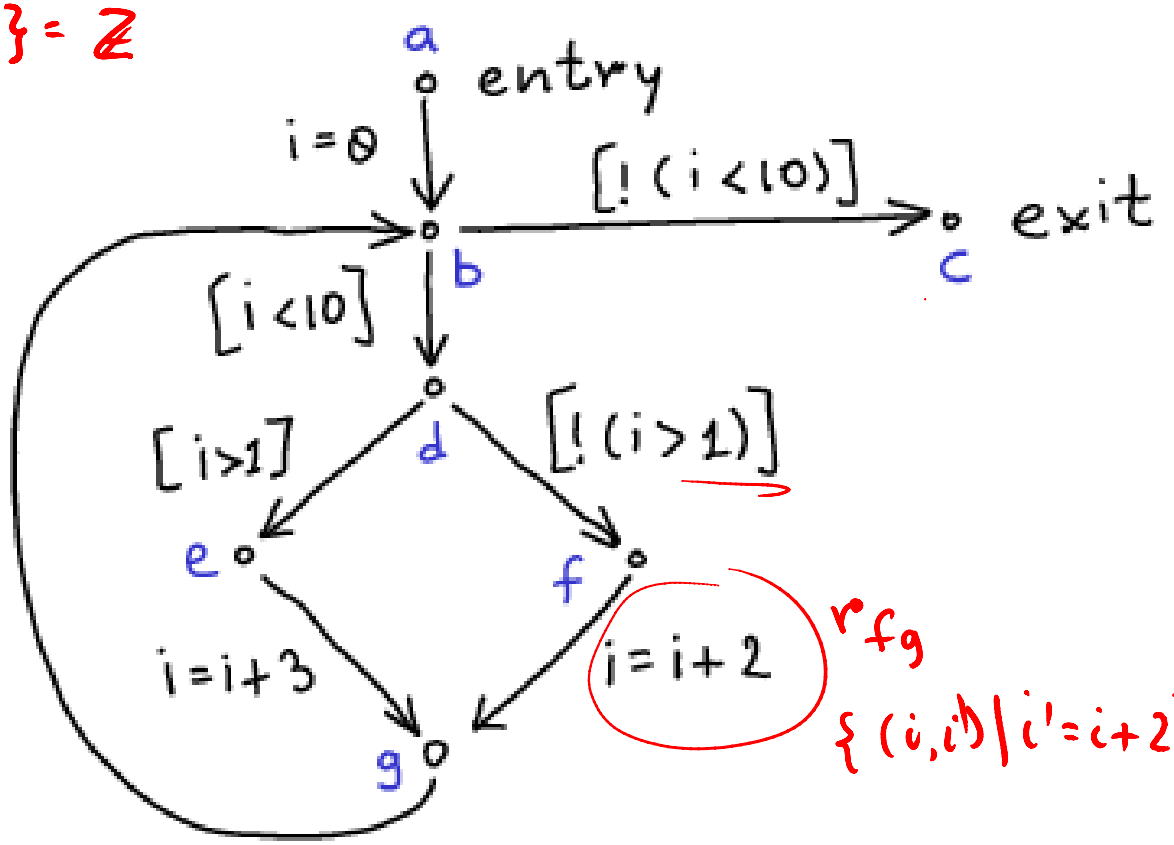
$S(d) = \{0,$

$S(e) = \{$

$S(f) = \{0$

$S(g) = \{2$

$S(c) = \{$



r_{fg}

$\{(i, i') \mid i' = i + 2\}$

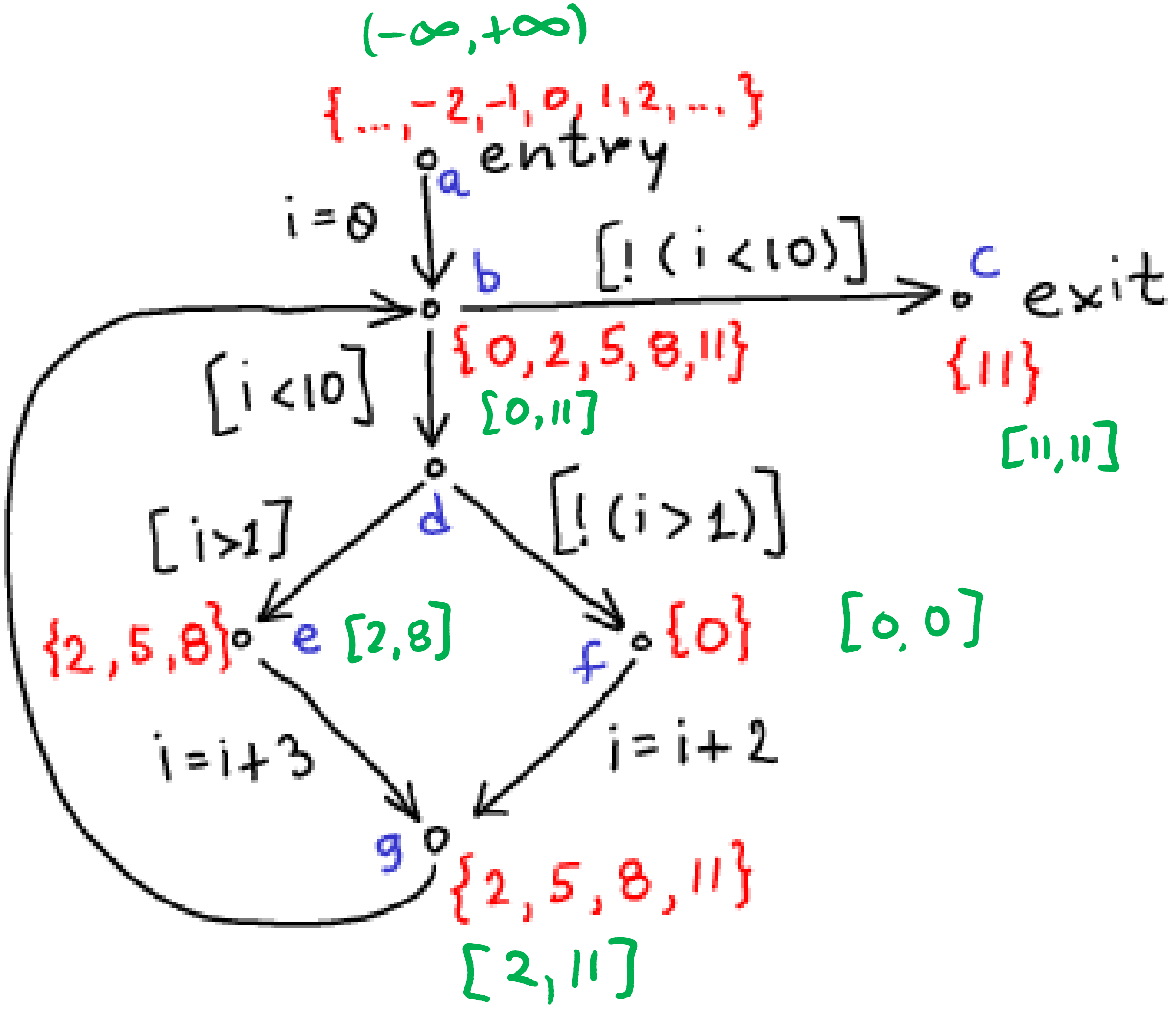
$S(g) = S(f) \bullet r_{fg}$

$\quad\quad S(e) \bullet r_{eg}$

$S(d) = \{0, 2, 5, 8\}$

$S^{\#}(d) = [0, 8]$

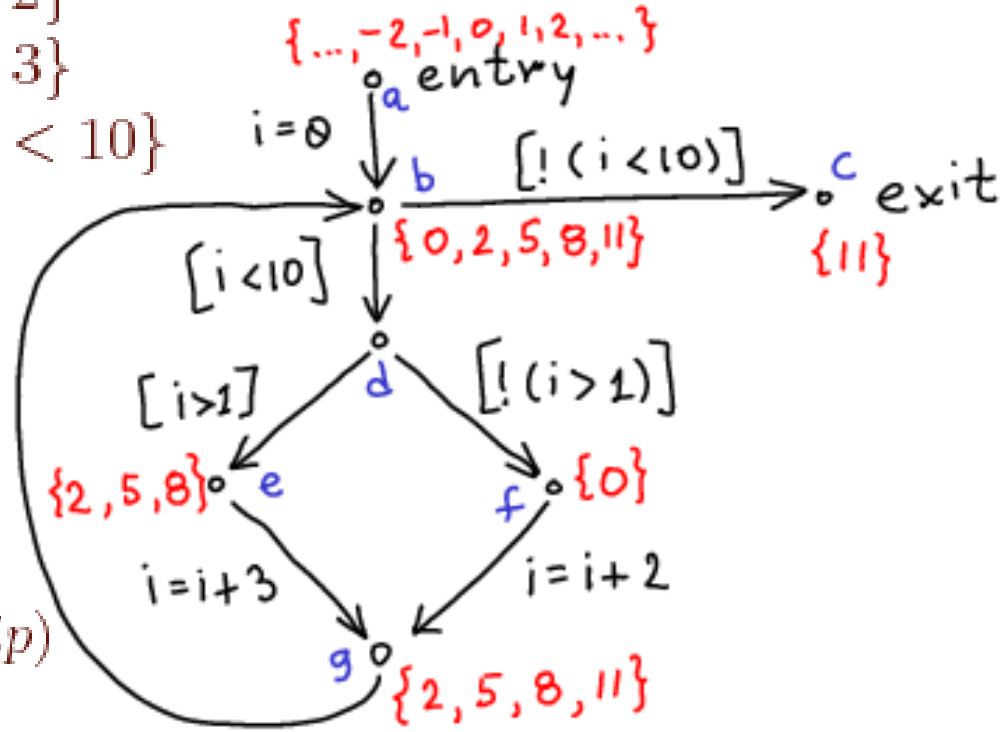
```
i = 0;
while (i < 10) {
  if (i > 1)
    i = i + 3;
  else
    i = i + 2;
}
```



Sets are Given by Equations

$$R(i = 0) = \{(i, i') \mid i' = 0\}$$
$$R(i = i + 2) = \{(i, i') \mid i' = i + 2\}$$
$$R(i = i + 3) = \{(i, i') \mid i' = i + 3\}$$
$$R([i < 10]) = \{(i, i') \mid i' = i \wedge i < 10\}$$

$T(S, r) = sp(S, r) = S.r$



$$S(a) = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$$
$$S(b) = T(S(a), i = 0) \cup T(S(g), skip)$$
$$S(c) = T(S(b), [\neg(i < 10)])$$
$$S(d) = T(S(b), [i < 10])$$
$$S(e) = T(S(d), [i > 1])$$
$$S(f) = T(S(d), [\neg(i > 1)])$$
$$S(g) = T(S(e), i = i + 3) \cup T(S(f), i = i + 2)$$

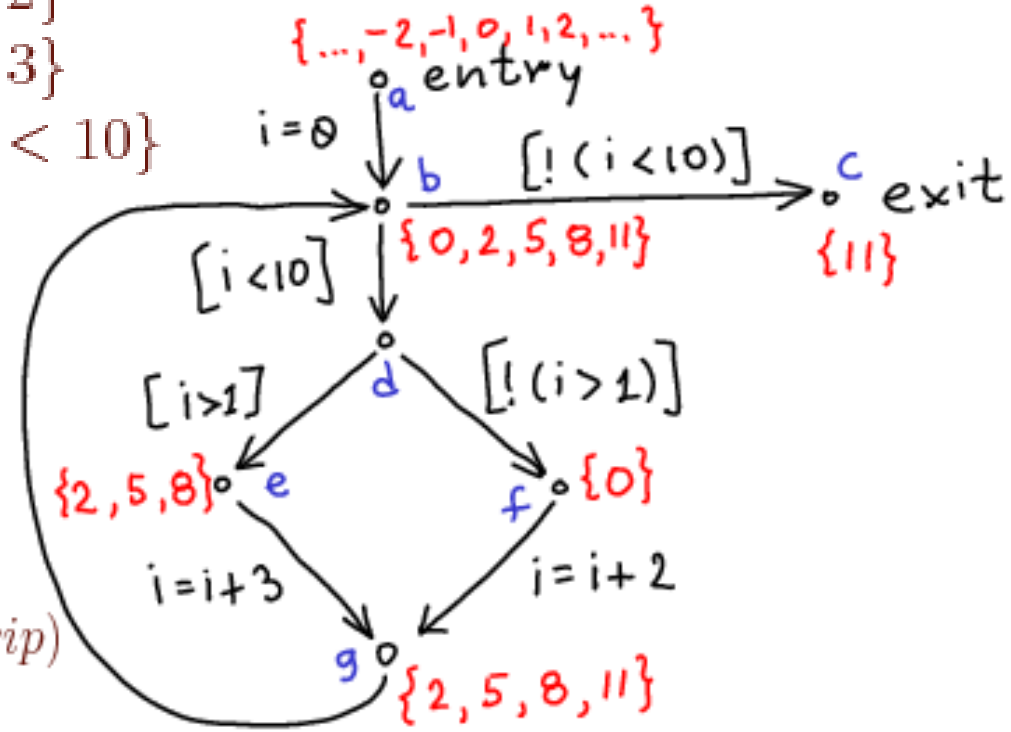
Sets are Given by Equations

$$R(i = 0) = \{(i, i') \mid i' = 0\}$$
$$R(i = i + 2) = \{(i, i') \mid i' = i + 2\}$$
$$R(i = i + 3) = \{(i, i') \mid i' = i + 3\}$$
$$R([i < 10]) = \{(i, i') \mid i' = i \wedge i < 10\}$$

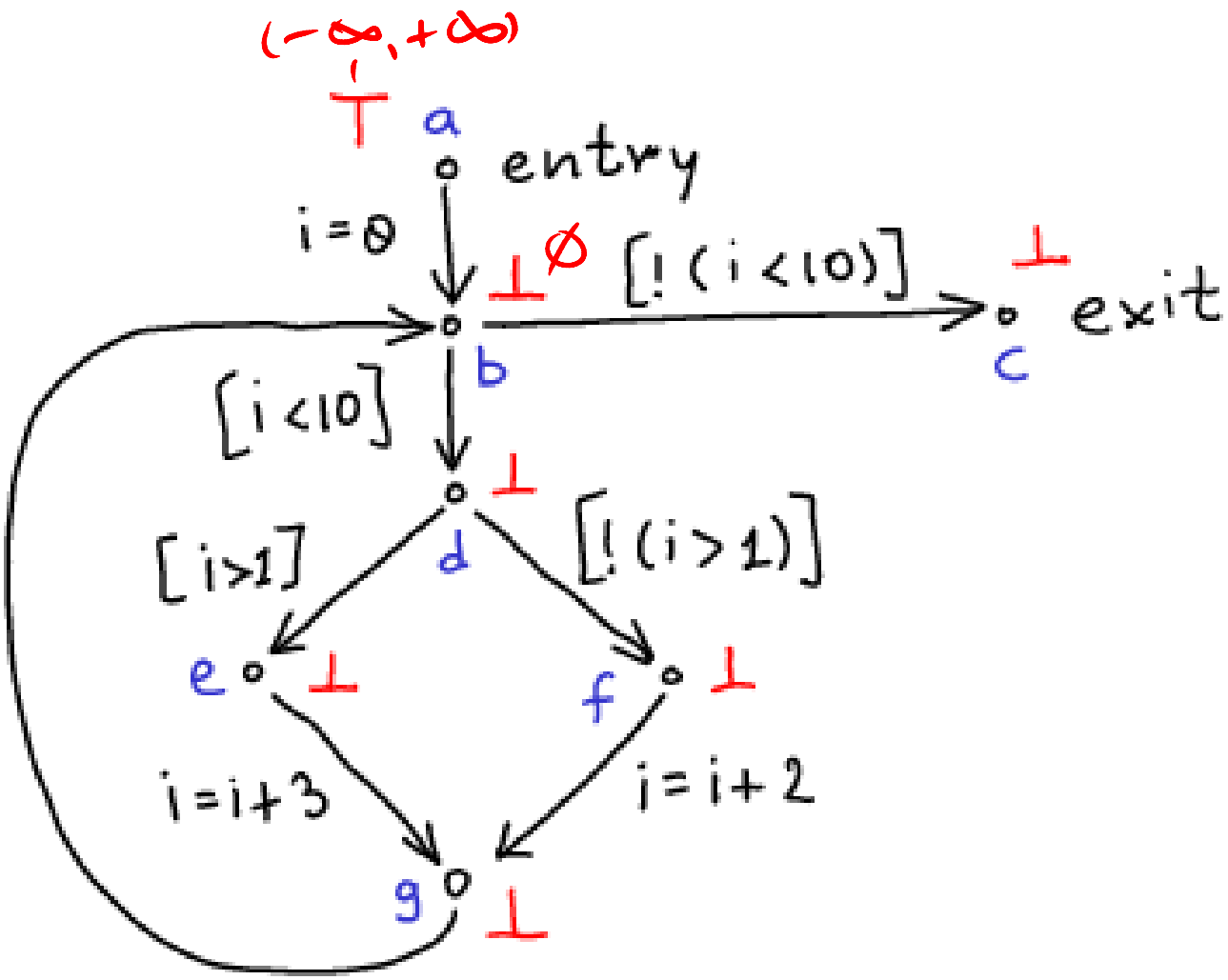
$T(S, r) = sp(S, r) = S.r$

$T^{\#}(S^{\#}, r) = sp^{\#}(S^{\#}, r) \supseteq sp(S, r)$

safe approximation



$$S^{\#}(a) = \top$$
$$S^{\#}(b) = T^{\#}(S^{\#}(a), i = 0) \sqcup T(S(g), skip)$$
$$S^{\#}(c) = T^{\#}(S^{\#}(b), [\neg(i < 10)])$$
$$S^{\#}(d) = T^{\#}(S^{\#}(b), [i < 10])$$
$$S^{\#}(e) = T^{\#}(S^{\#}(d), [i > 1])$$
$$S^{\#}(f) = T^{\#}(S^{\#}(d), [\neg(i > 1)])$$
$$S^{\#}(g) = T^{\#}(S^{\#}(e), i = i + 3) \sqcup T(S(f), i = i + 2)$$


(−∞, +∞)
⊤
a
entry
i = 0
⊥
∅
[! (i < 10)]
⊥
exit
c
b
[i < 10]
⊥
d
[i > 1]
[! (i > 1)]
e
⊥
f
⊥
i = i + 3
i = i + 2
g
⊥

grow the intervals

$[0,0] \cup [2,2] = \{0,2\}$

$[0,0] \sqcup [2,2] = [0,2]$

$s^{\#}(f).r_{fg} \subseteq s^{\#}(g)$