Predicate Abstraction

Hossein Hojjat

LARA

SAV 2012

Abstraction and Concretization

- $\overrightarrow{\mathcal{V}} = (x_0, \cdots, x_{n-1})$ program variables
- $\mathcal{S}$: set of concrete program states
  - Example: $\overrightarrow{\mathcal{V}} = (x_0, x_1)$, $\mathcal{S} = \{(4,5), (1,2), (-7,8)\}$
- $\mathcal{P} = \{P_0, \cdots, P_{m-1}\}$ finite set of predicates on program variables
  - Example $\mathcal{P} = \{P_0, P_1, P_2\}$, $P_0 = \mathit{false}$, $P_1 = x_1 > 0$, $P_2 = x_0 < x_1$
- $\mathcal{A} = 2^{\mathcal{P}}$: abstract domain
- Abstraction function $\alpha : 2^{\mathcal{S}} \to \mathcal{A}$

$$\alpha(c) = \{P_i | \forall \overrightarrow{v} \in c. P_i(\overrightarrow{v})\}$$

- Concretization function $\gamma : \mathcal{A} \to 2^{\mathcal{S}}$

$$\gamma(a) = \{\overrightarrow{v} | \bigwedge_{P \in a} P(\overrightarrow{v})\}$$

Abstraction

Example

- Let $\mathcal{P} = \{x > y, x = 2\}$
- What is the abstraction after executing the following piece of code?

$\{\text{true}\}$
val x: Int
val y: Int
$\mathsf{x} = \mathsf{y} + 1$

- $\forall x, y, x', y' \in \mathbb{Z}.(x' = y + 1) \wedge (y' = y) \rightarrow (x' > y')$ (valid)
- $\forall x, y, x', y' \in \mathbb{Z}.(x' = y + 1) \wedge (y' = y) \rightarrow (x' = 2)$ (not valid)

The abstraction is $\{(x > y)\}$

Abstraction

Example

- Let $\mathcal{P} = \{x < 2\}$
- What is the abstraction after executing the following piece of code?

$\{\mathrm{x} = 2\}$
if $(\mathrm{x} > 2)\ \mathrm{x} = \mathrm{x} - 1$

- $\forall x, x'.(x = 2) \wedge (x > 2) \wedge (x' = x - 1) \rightarrow (x' < 2)$
  $\equiv \forall x, x'. \perp \rightarrow (x' < 2) \quad \text{(valid)}$
- The abstraction in this case :$\{(x < 2)\}$
- The predicate $\perp$ is usually included in $\mathcal{P}$

Abstract Reachability Tree

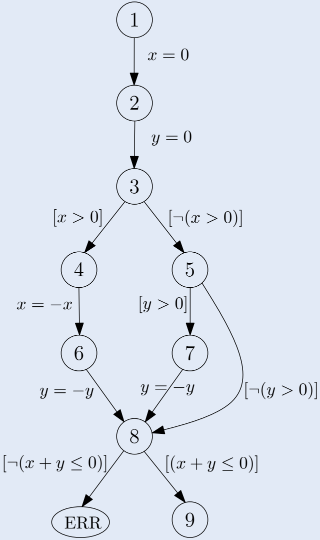
- Abstract state (l, ψ)
  - l: location in control flow graph
  - ψ: predicate abstraction

- Abstract reachability tree (ART) is a tree $G = (V_{\mathcal{A}}, \rightarrow, I)$
- $V_{\mathcal{A}}$ is a set of abstract states
- $\rightarrow \subseteq V_{\mathcal{A}} \times V_{\mathcal{A}}$ is the transition relation
  - Let c be the command between l_i and l_j in the CFG
  - $((l_i, \psi), (l_j, \phi)) \in \rightarrow$ if $\phi = sp^{\#}(\psi, c)$
- $I \in V_{\mathcal{A}}$ is the initial abstract state

- (l_i, ψ) is leaf if there exists another node (l_j, ϕ) in the tree such that $\models \psi \rightarrow \phi$

ART Example

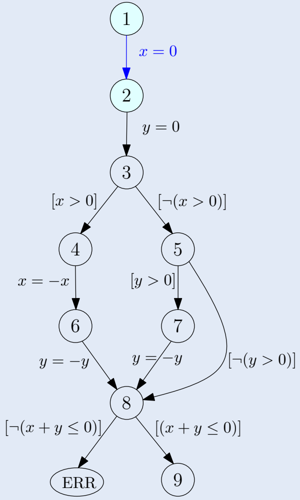
Control Flow Graph

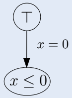



```
var x = 0
var y = 0
if( x > 0) {
  x = -x
  y = -y
} else {
    if( y > 0) y = -y
}
assert(x + y <= 0)
```

ART Example

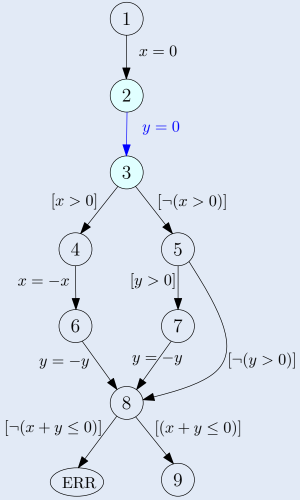
$$\mathcal{P} = \{\bot, (x <= 0), (x + y <= 0), (x - y <= 0)\}$$

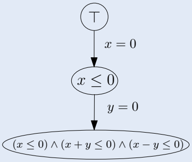




ART Example

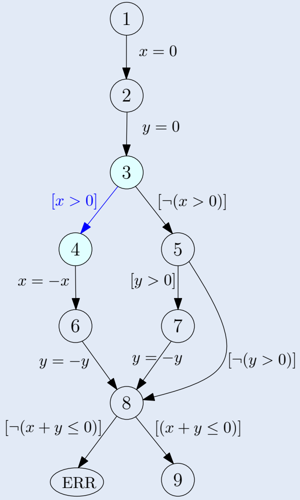
$$\mathcal{P} = \{\bot, (x <= 0), (x + y <= 0), (x - y <= 0)\}$$

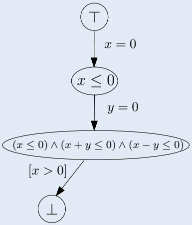




ART Example

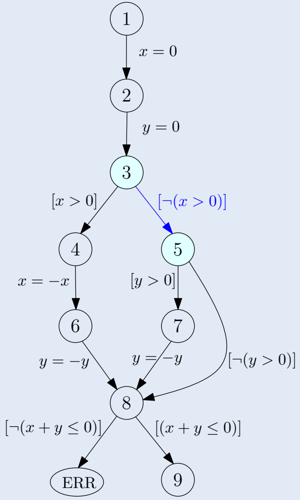
$$\mathcal{P} = \{\bot, (x <= 0), (x+y <= 0), (x-y <= 0)\}$$

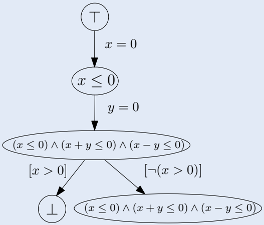




ART Example

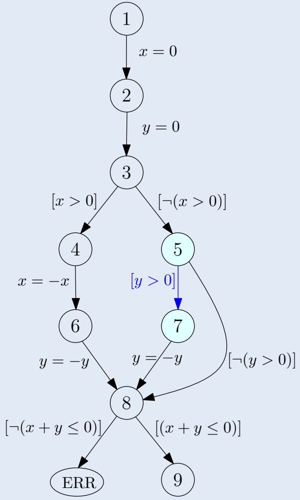
$$\mathcal{P} = \{\bot, (x <= 0), (x + y <= 0), (x - y <= 0)\}$$

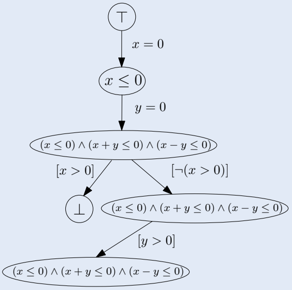




ART Example

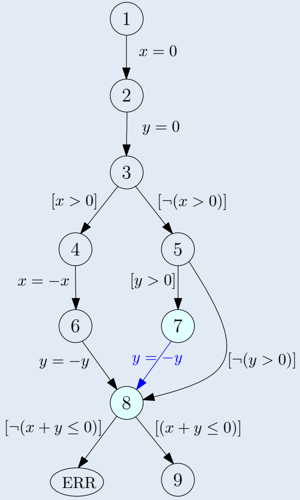
$$\mathcal{P} = \{\bot, (x <= 0), (x + y <= 0), (x - y <= 0)\}$$

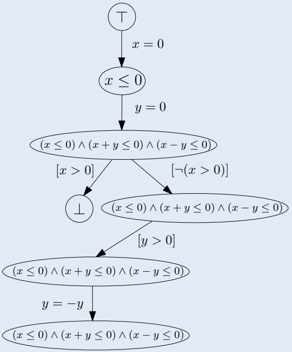




ART Example

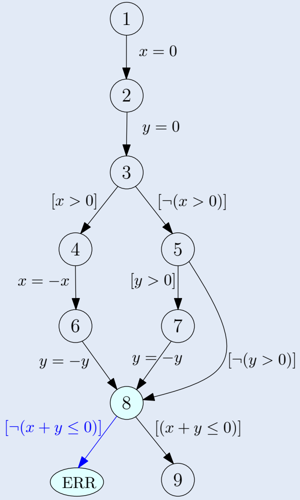
$$\mathcal{P} = \{\bot, (x <= 0), (x + y <= 0), (x - y <= 0)\}$$

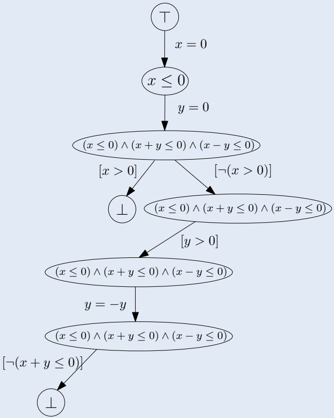




ART Example

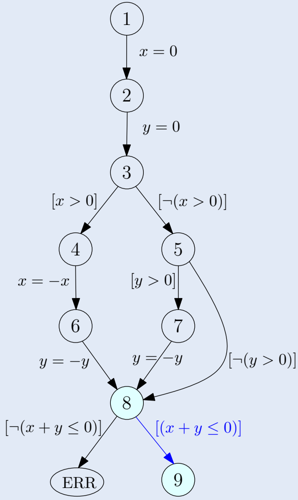
$$\mathcal{P} = \{\bot, (x <= 0), (x + y <= 0), (x - y <= 0)\}$$

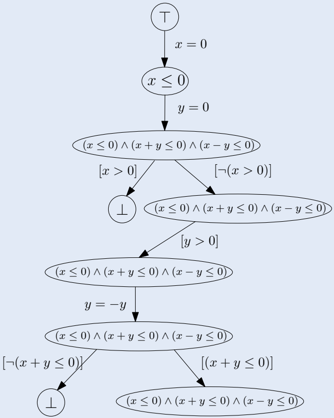




ART Example

$$\mathcal{P} = \{\bot, (x <= 0), (x + y <= 0), (x - y <= 0)\}$$





ART Example

$$\mathcal{P} = \{\bot, (x <= 0), (x + y <= 0), (x - y <= 0)\}$$