Lecturecise 17
Predicate Abstraction. Predicate Discovery

2013

Predicate Abstraction

Abstract interpretation domain is determined by a set of formulas (predicates) $\mathcal{P}$ on program variables.
Example: $\mathcal{P} = \{P_0, P_1, P_2, P_3\}$ where

$$\begin{array}{lcl} P_0 & \equiv & \text{false} \\ P_1 & \equiv & 0 < x \\ P_2 & \equiv & 0 < y \\ P_3 & \equiv & x < y \end{array}$$

Analysis tries to construct invariants from these predicates using

- conjunctions, e.g. $P_1 \wedge P_3$
- more generally, conjunctions and disjunctions, e.g. $P_3 \wedge (P_1 \vee P_2)$

Predicate Abstraction

Abstract interpretation domain is determined by a set of formulas (predicates) $\mathcal{P}$ on program variables.
Example: $\mathcal{P} = \{P_0, P_1, P_2, P_3\}$ where

$$
\begin{array}{lcl}
P_0 & \equiv & \text{false} \\
P_1 & \equiv & 0 < x \\
P_2 & \equiv & 0 < y \\
P_3 & \equiv & x < y
\end{array}
$$

Analysis tries to construct invariants from these predicates using

- conjunctions, e.g. $P_1 \wedge P_3$
- more generally, conjunctions and disjunctions, e.g. $P_3 \wedge (P_1 \vee P_2)$

For now: we consider only conjunctions.
We assume $P_0 \equiv \text{false}$, other predicates in $\mathcal{P}$ are arbitrary

- expressed in a logic for which we have a theorem prover

Example of Analysis Result

$$\mathcal{P} = \{false, 0 < x, 0 <= x, 0 < y, x < y, x = 0, y = 1, x < 1000, 1000 \leq x\}$$

```
x = 0;
y = 1;
// 0<y, x<y,x=0,y=1, x<1000
// 0<y, 0≤x, x<y
while (x < 1000) {
  // 0<y, 0≤x, x<y, x<1000
  x = x + 1;
  // 0<y, 0≤x, 0<x
  y = 2*x;
  // 0<y, 0≤x, 0<x, x<y
  y = y + 1;
  // 0<y, 0≤x, 0<x, x<y
  print(y);
}
// 0<y, 0≤x, x<y, 1000 ≤ x
```

Lattice of Conjunctions of Predicates and Concretization

$\mathcal{P} = \{P_0, P_1, \ldots, P_n\}$ - predicates

- formulas whose free variables denote program variables

$A = 2^{\mathcal{P}}$, so for $a \in A$ we have $a \subseteq \mathcal{P}$
Example: $a_0 = \{0 < x, x < y\}$.
$s \models F$ means: formula F is true for variables given by the program state s

$$\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$$

Shorthand: $\bigwedge a$ means $\bigwedge_{P \in a} P$
Example: $\gamma(a_0) =$

Lattice of Conjunctions of Predicates and Concretization

$\mathcal{P} = \{P_0, P_1, \ldots, P_n\}$ - predicates

- formulas whose free variables denote program variables

$A = 2^{\mathcal{P}}$, so for $a \in A$ we have $a \subseteq \mathcal{P}$
Example: $a_0 = \{0 < x, x < y\}$.
$s \models F$ means: formula F is true for variables given by the program state s

$$\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$$

Shorthand: $\bigwedge a$ means $\bigwedge_{P \in a} P$
Example: $\gamma(a_0) = \{s \mid s \models 0 < x \wedge x < y\}$.

Lattice of Conjunctions of Predicates and Concretization

$\mathcal{P} = \{P_0, P_1, \ldots, P_n\}$ - predicates

- formulas whose free variables denote program variables

$A = 2^{\mathcal{P}}$, so for $a \in A$ we have $a \subseteq \mathcal{P}$
Example: $a_0 = \{0 < x, x < y\}$.
$s \models F$ means: formula F is true for variables given by the program state s

$$\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$$

Shorthand: $\bigwedge a$ means $\bigwedge_{P \in a} P$
Example: $\gamma(a_0) = \{s \mid s \models 0 < x \wedge x < y\}$. We often assume states are pairs (x, y). Then $\gamma(a_0) =$

Lattice of Conjunctions of Predicates and Concretization

$\mathcal{P} = \{P_0, P_1, \ldots, P_n\}$ - predicates

- formulas whose free variables denote program variables

$A = 2^{\mathcal{P}}$, so for $a \in A$ we have $a \subseteq \mathcal{P}$
Example: $a_0 = \{0 < x, x < y\}$.
$s \models F$ means: formula F is true for variables given by the program state s

$$\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$$

Shorthand: $\bigwedge a$ means $\bigwedge_{P \in a} P$
Example: $\gamma(a_0) = \{s \mid s \models 0 < x \land x < y\}$. We often assume states are pairs (x, y). Then $\gamma(a_0) = \{(x, y) \mid 0 < x \land x < y\}$.

Lattice of Conjunctions of Predicates and Concretization

$\mathcal{P} = \{P_0, P_1, \ldots, P_n\}$ - predicates

- formulas whose free variables denote program variables

$A = 2^{\mathcal{P}}$, so for $a \in A$ we have $a \subseteq \mathcal{P}$
Example: $a_0 = \{0 < x, x < y\}$.
$s \models F$ means: formula F is true for variables given by the program state s

$$\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$$

Shorthand: $\bigwedge a$ means $\bigwedge_{P \in a} P$
Example: $\gamma(a_0) = \{s \mid s \models 0 < x \wedge x < y\}$. We often assume states are pairs (x, y). Then $\gamma(a_0) = \{(x, y) \mid 0 < x \wedge x < y\}$.
If $a_1 \subseteq a_2$ then $\bigwedge a_2$ implies $\bigwedge a_1$, so $\gamma(a_2) \subseteq \gamma(a_1)$.

Lattice of Conjunctions of Predicates and Concretization

$\mathcal{P} = \{P_0, P_1, \ldots, P_n\}$ - predicates

- formulas whose free variables denote program variables

$A = 2^{\mathcal{P}}$, so for $a \in A$ we have $a \subseteq \mathcal{P}$
Example: $a_0 = \{0 < x, x < y\}$.
$s \models F$ means: formula F is true for variables given by the program state s

$$\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$$

Shorthand: $\bigwedge a$ means $\bigwedge_{P \in a} P$
Example: $\gamma(a_0) = \{s \mid s \models 0 < x \wedge x < y\}$. We often assume states are pairs (x, y). Then $\gamma(a_0) = \{(x, y) \mid 0 < x \wedge x < y\}$.
If $a_1 \subseteq a_2$ then $\bigwedge a_2$ implies $\bigwedge a_1$, so $\gamma(a_2) \subseteq \gamma(a_1)$.
Define:

$$a_1 \sqsubseteq a_2 \quad \iff \quad a_2 \subseteq a_1$$

Lemma: $a_1 \sqsubseteq a_2 \rightarrow \gamma(a_1) \subseteq \gamma(a_2)$

Lattice of Conjunctions of Predicates and Concretization

$\mathcal{P} = \{P_0, P_1, \ldots, P_n\}$ - predicates

- formulas whose free variables denote program variables

$A = 2^{\mathcal{P}}$, so for $a \in A$ we have $a \subseteq \mathcal{P}$
Example: $a_0 = \{0 < x, x < y\}$.
$s \models F$ means: formula F is true for variables given by the program state s

$$\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$$

Shorthand: $\bigwedge a$ means $\bigwedge_{P \in a} P$
Example: $\gamma(a_0) = \{s \mid s \models 0 < x \land x < y\}$. We often assume states are pairs (x, y). Then $\gamma(a_0) = \{(x, y) \mid 0 < x \land x < y\}$.
If $a_1 \subseteq a_2$ then $\bigwedge a_2$ implies $\bigwedge a_1$, so $\gamma(a_2) \subseteq \gamma(a_1)$.
Define:

$$a_1 \sqsubseteq a_2 \iff a_2 \subseteq a_1$$

Lemma: $a_1 \sqsubseteq a_2 \rightarrow \gamma(a_1) \subseteq \gamma(a_2)$
Does the converse hold?

Size of the Lattice

$$\{\mathit{false}, 0 < x, x < y\} \sqsubseteq \{0 < x, 0 < y\} \sqsubseteq \{0 < x\} \sqsubseteq \emptyset$$

Draw the Hasse diagram for the lattice $(A, \sqsubseteq)$ i.e. $(2^{\mathcal{P}}, \supseteq)$ for $\mathcal{P} = \{P_0, P_1, P_2\}$ a three-element set.

Size of the Lattice

$$\{\mathit{false}, 0 < x, x < y\} \sqsubseteq \{0 < x, 0 < y\} \sqsubseteq \{0 < x\} \sqsubseteq \emptyset$$

Draw the Hasse diagram for the lattice $(A, \sqsubseteq)$ i.e. $(2^{\mathcal{P}}, \supseteq)$ for $\mathcal{P} = \{P_0, P_1, P_2\}$ a three-element set.

What is the top and what is the bottom element of this lattice?

Size of the Lattice

$$\{\mathit{false}, 0 < x, x < y\} \sqsubseteq \{0 < x, 0 < y\} \sqsubseteq \{0 < x\} \sqsubseteq \emptyset$$

Draw the Hasse diagram for the lattice $(A, \sqsubseteq)$ i.e. $(2^{\mathcal{P}}, \supseteq)$ for $\mathcal{P} = \{P_0, P_1, P_2\}$ a three-element set.

What is the top and what is the bottom element of this lattice?
What is the height of the lattice?

Size of the Lattice

$$\{false, 0 < x, x < y\} \sqsubseteq \{0 < x, 0 < y\} \sqsubseteq \{0 < x\} \sqsubseteq \emptyset$$

Draw the Hasse diagram for the lattice $(A, \sqsubseteq)$ i.e. $(2^{\mathcal{P}}, \supseteq)$ for $\mathcal{P} = \{P_0, P_1, P_2\}$ a three-element set.

What is the top and what is the bottom element of this lattice?
What is the height of the lattice?
What is the height of such lattice when $\mathcal{P} = \{P_0, P_1, \ldots, P_n\}$?

Size of the Lattice

$$\{false, 0 < x, x < y\} \sqsubseteq \{0 < x, 0 < y\} \sqsubseteq \{0 < x\} \sqsubseteq \emptyset$$

Draw the Hasse diagram for the lattice $(A, \sqsubseteq)$ i.e. $(2^{\mathcal{P}}, \supseteq)$ for $\mathcal{P} = \{P_0, P_1, P_2\}$ a three-element set.

What is the top and what is the bottom element of this lattice?
What is the height of the lattice?
What is the height of such lattice when $\mathcal{P} = \{P_0, P_1, \ldots, P_n\}$?
Do $\sqcup$ and $\sqcap$ exist?

Galois Connection

For $\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$ we define

$$\alpha(c) = \{P \in \mathcal{P} \mid \forall s \in c.\ s \models P\}$$

$$\alpha(\{(-1, 1)\}) \quad =$$

Galois Connection

For $\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$ we define

$$\alpha(c) = \{P \in \mathcal{P} \mid \forall s \in c.\ s \models P\}$$

$$\alpha(\{(-1, 1)\}) \quad = \quad \{y > 0\}$$

Galois Connection

For $\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$ we define

$$\alpha(c) = \{P \in \mathcal{P} \mid \forall s \in c.\ s \models P\}$$

$$\begin{array}{rcl} \alpha(\{(-1,1)\}) & = & \{y > 0\} \\ \alpha(\{(1,1),(2,2),(3,6)\}) & = & \end{array}$$

Galois Connection

For $\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$ we define

$$\alpha(c) = \{P \in \mathcal{P} \mid \forall s \in c.\ s \models P\}$$

$$\begin{array}{rcl}
\alpha(\{(-1,1)\}) & = & \{y > 0\} \\
\alpha(\{(1,1),(2,2),(3,6)\}) & = & \{x > 0, y > 0\}
\end{array}$$

Galois Connection

For $\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$ we define

$$\alpha(c) = \{P \in \mathcal{P} \mid \forall s \in c.\ s \models P\}$$

$$\begin{array}{rcl} \alpha(\{(-1,1)\}) & = & \{y > 0\} \\ \alpha(\{(1,1),(2,2),(3,6)\}) & = & \{x > 0, y > 0\} \\ \alpha(\{(1,0),(0,1)\}) & = & \end{array}$$

Galois Connection

For $\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$ we define

$$\alpha(c) = \{P \in \mathcal{P} \mid \forall s \in c.\ s \models P\}$$

$$\begin{array}{rcl} \alpha(\{(-1,1)\}) & = & \{y > 0\} \\ \alpha(\{(1,1),(2,2),(3,6)\}) & = & \{x > 0, y > 0\} \\ \alpha(\{(1,0),(0,1)\}) & = & \emptyset \end{array}$$

Galois Connection

For $\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$ we define

$$\alpha(c) = \{P \in \mathcal{P} \mid \forall s \in c.\ s \models P\}$$

$$\begin{array}{rcl}
\alpha(\{(-1,1)\}) & = & \{y > 0\} \\
\alpha(\{(1,1),(2,2),(3,6)\}) & = & \{x > 0, y > 0\} \\
\alpha(\{(1,0),(0,1)\}) & = & \emptyset \\
\gamma(\emptyset) & = &
\end{array}$$

Galois Connection

For $\gamma(a) = \{s \mid s \models \bigwedge_{P \in a} P\}$ we define

$$\alpha(c) = \{P \in \mathcal{P} \mid \forall s \in c.\ s \models P\}$$

$$\begin{array}{rcl} \alpha(\{(-1,1)\}) & = & \{y > 0\} \\ \alpha(\{(1,1),(2,2),(3,6)\}) & = & \{x > 0, y > 0\} \\ \alpha(\{(1,0),(0,1)\}) & = & \emptyset \\ \gamma(\emptyset) & = & S \text{ (set of all states, empty conjunction)} \end{array}$$

Is (α, γ) a Galois connection between $(A, \sqsubseteq)$ and $(C, \subseteq)$?

Galois Connection for Predicate Abstraction

We show (α, γ) is a Galois Connection. We need to show that

$$c \subseteq \gamma(a) \iff \alpha(c) \supseteq a$$

Galois Connection for Predicate Abstraction

We show (α, γ) is a Galois Connection. We need to show that

$$c \subseteq \gamma(a) \iff \alpha(c) \supseteq a$$

But both conditions easily reduce to

$$\forall P \in a.\ \forall s \in c.\ s \models P$$

Galois Connection for Predicate Abstraction

We show (α, γ) is a Galois Connection. We need to show that

$$c \subseteq \gamma(a) \iff \alpha(c) \supseteq a$$

But both conditions easily reduce to

$$\forall P \in a. \, \forall s \in c. \; s \models P$$

Shorthand: in logic, if M is a set of assignments to variables (structures) and $\mathcal{A}$ is a set of formulas (e.g. axioms), then $M \models \mathcal{A}$ means

$$\forall m \in M. \forall F \in \mathcal{A}. \; m \models F$$

So, both conditions of Galois connection reduce to $c \models a$

Not a Galois Insertion

Is it the case that $\alpha(\gamma(a)) = a$?

Not a Galois Insertion

Is it the case that $\alpha(\gamma(a)) = a$?
We show this is not the case. This is because γ is not injective.
Indeed, take $a_1 = \{\mathit{false}\}$ and $a_2 = \{\mathit{false}, x > 0\}$. Then

$$\gamma(a_1) = \emptyset = \gamma(a_2)$$

Note $\alpha(\gamma(a_1)) = \alpha(\mathcal{P}) = \alpha(\gamma(a_2))$, but $a_1 \neq a_2$, but it is not the case that $a_1 = a_2$. In this particular case,

$$\alpha(\emptyset) = \mathcal{P}$$

and $a_1 \neq \mathcal{P}$ so

$$\alpha(\gamma(a_1)) \neq a_1$$

Not a Galois Insertion

Is it the case that $\alpha(\gamma(a)) = a$?
We show this is not the case. This is because γ is not injective.
Indeed, take $a_1 = \{\mathit{false}\}$ and $a_2 = \{\mathit{false}, x > 0\}$. Then

$$\gamma(a_1) = \emptyset = \gamma(a_2)$$

Note $\alpha(\gamma(a_1)) = \alpha(\mathcal{P}) = \alpha(\gamma(a_2))$, but $a_1 \neq a_2$, but it is not the case that $a_1 = a_2$. In this particular case,

$$\alpha(\emptyset) = \mathcal{P}$$

and $a_1 \neq \mathcal{P}$ so

$$\alpha(\gamma(a_1)) \neq a_1$$

However, the approach works and is sound, even without the condition $\alpha(\gamma(a)) = a$.

Not a Galois Insertion

Is it the case that $\alpha(\gamma(a)) = a$?
We show this is not the case. This is because γ is not injective.
Indeed, take $a_1 = \{false\}$ and $a_2 = \{false, x > 0\}$. Then

$$\gamma(a_1) = \emptyset = \gamma(a_2)$$

Note $\alpha(\gamma(a_1)) = \alpha(\mathcal{P}) = \alpha(\gamma(a_2))$, but $a_1 \neq a_2$, but it is not the case that $a_1 = a_2$. In this particular case,

$$\alpha(\emptyset) = \mathcal{P}$$

and $a_1 \neq \mathcal{P}$ so

$$\alpha(\gamma(a_1)) \neq a_1$$

However, the approach works and is sound, even without the condition $\alpha(\gamma(a)) = a$.
Can you find an example of non-injectivity in our 4 predicates that does not involve false?

Monotonicity of α

Let $c_1 \subseteq c_2$.
We wish to prove that $\alpha(c_1) \supseteq \alpha(c_2)$.

Monotonicity of α

Let $c_1 \subseteq c_2$.
We wish to prove that $\alpha(c_1) \supseteq \alpha(c_2)$.
Let $P \in \alpha(c_2)$. Then for all $(x, y) \in c_2$ we have $P(x, y)$.

Monotonicity of α

Let $c_1 \subseteq c_2$.
We wish to prove that $\alpha(c_1) \supseteq \alpha(c_2)$.
Let $P \in \alpha(c_2)$. Then for all $(x, y) \in c_2$ we have $P(x, y)$.
Then also for all $(x, y) \in c_1$ we have $P(x, y)$, because $c_1 \subseteq c_2$.

Monotonicity of α

Let $c_1 \subseteq c_2$.
We wish to prove that $\alpha(c_1) \supseteq \alpha(c_2)$.
Let $P \in \alpha(c_2)$. Then for all $(x, y) \in c_2$ we have $P(x, y)$.
Then also for all $(x, y) \in c_1$ we have $P(x, y)$, because $c_1 \subseteq c_2$.
Therefore $P \in \alpha(c_1)$. We showed $c_2 \subseteq c_1$, so $c_1 \sqsubseteq c_2$.

Computing Approximate Strongest Postcondition

$\mathcal{P} = \{false, 0 < x, 0 < y, x < y\}$
Consider computing $sp^{\#}(\{0 < x\}, y := x + 1)$. We can test for each predicate $P' \in \mathcal{P}$ whether

$$x > 0 \land (y' = x + 1 \land x' = x) \implies P'(x', y')$$

We obtain that the condition holds for $0 < x$, $0 < y$, and for $x < y$, but not for *false*. Thus,

$$sp^{\#}(\{0 < x\}, y := x + 1) = \{0 < x, 0 < y, x < y\}$$

Compute

$$sp^{\#}(\{0 < x\}, y := x - 1) =$$

Computing Approximate Strongest Postcondition

$\mathcal{P} = \{\mathit{false}, 0 < x, 0 < y, x < y\}$
Consider computing $sp^{\#}(\{0 < x\}, y := x + 1)$. We can test for each predicate $P' \in \mathcal{P}$ whether

$$x > 0 \wedge (y' = x + 1 \wedge x' = x) \implies P'(x', y')$$

We obtain that the condition holds for $0 < x$, $0 < y$, and for $x < y$, but not for *false*. Thus,

$$sp^{\#}(\{0 < x\}, y := x + 1) = \{0 < x, 0 < y, x < y\}$$

Compute

$$sp^{\#}(\{0 < x\}, y := x - 1) = \{0 < x\}$$

Computing Approximate Strongest Postcondition

$\mathcal{P} = \{false, 0 < x, 0 < y, x < y\}$
Consider computing $sp^{\#}(\{0 < x\}, y := x + 1)$. We can test for each predicate $P' \in \mathcal{P}$ whether

$$x > 0 \wedge (y' = x + 1 \wedge x' = x) \implies P'(x', y')$$

We obtain that the condition holds for $0 < x$, $0 < y$, and for $x < y$, but not for *false*. Thus,

$$sp^{\#}(\{0 < x\}, y := x + 1) = \{0 < x, 0 < y, x < y\}$$

Compute

$$sp^{\#}(\{0 < x\}, y := x - 1) = \{0 < x\}$$
$$sp^{\#}(\{0 < x, x < y\}, x := x - 1) =$$

Computing Approximate Strongest Postcondition

$\mathcal{P} = \{\mathit{false}, 0 < x, 0 < y, x < y\}$
Consider computing $sp^{\#}(\{0 < x\}, y := x + 1)$. We can test for each predicate $P' \in \mathcal{P}$ whether

$$x > 0 \wedge (y' = x + 1 \wedge x' = x) \implies P'(x', y')$$

We obtain that the condition holds for $0 < x$, $0 < y$, and for $x < y$, but not for *false*. Thus,

$$sp^{\#}(\{0 < x\}, y := x + 1) = \{0 < x, 0 < y, x < y\}$$

Compute

$$sp^{\#}(\{0 < x\}, y := x - 1) = \{0 < x\}$$
$$sp^{\#}(\{0 < x, x < y\}, x := x - 1) = \{0 < y, x < y\}$$

Computing Approximate Strongest Postcondition

$\mathcal{P} = \{false, 0 < x, 0 < y, x < y\}$
Consider computing $sp^{\#}(\{0 < x\}, y := x + 1)$. We can test for each predicate $P' \in \mathcal{P}$ whether

$$x > 0 \land (y' = x + 1 \land x' = x) \implies P'(x', y')$$

We obtain that the condition holds for $0 < x$, $0 < y$, and for $x < y$, but not for *false*. Thus,

$$sp^{\#}(\{0 < x\}, y := x + 1) = \{0 < x, 0 < y, x < y\}$$

Compute

$$sp^{\#}(\{0 < x\}, y := x - 1) = \{0 < x\}$$
$$sp^{\#}(\{0 < x, x < y\}, x := x - 1) = \{0 < y, x < y\}$$

What is the relation between $\{0 < x, x < y\}$ and $\{0 < x, 0 < y, x < y\}$?

Deriving Rule for Computing sp

Fix some command given by relation r.
Denote $a' = sp^{\#}(a, r)$. We are computing a'. For correctness we need

$$sp(\gamma(a), r) \subseteq \gamma(a')$$

Thanks to Galois connection, this is equivalent to

$$\alpha(sp(\gamma(a), r)) \sqsubseteq a'$$

We wish to find the smallest lattice element a', which is the largest set (this gives the tightest approximation). So we let

$$a' = \alpha(sp(\gamma(a), r))$$

Given that $\gamma(a) = \{s \mid s \models \bigwedge a\}$, and $\alpha(c) = \{P \in \mathcal{P} \mid \forall s \in c.\ s \models P\}$,

$$a' = \{P' \in \mathcal{P} \mid \forall (x', y') \in sp(\gamma(a), r).\ P'(x', y')\}$$

Continuing the Derivation of sp

$$a' = \{P' \in \mathcal{P} \mid \forall (x', y').(x', y') \in sp(\gamma(a), r) \rightarrow P'(x', y')\}$$

Let $R(x, y, x', y')$ denote the meaning of relation r

Continuing the Derivation of sp

$$a' = \{P' \in \mathcal{P} \mid \forall(x', y').(x', y') \in sp(\gamma(a), r) \rightarrow P'(x', y')\}$$

Let $R(x, y, x', y')$ denote the meaning of relation r
Then $(x', y') \in sp(\gamma(a), r)$ means

$$\exists x, y.(x, y) \in \gamma(a) \land R(x, y, x', y')$$

which, after expanding γ, gives

$$\exists x, y.(\bigwedge_{P \in a} P(x, y)) \land R(x, y, x', y')$$

We then plug this expression back into a' definition. Because the existentials are left of implication, the result is:

$$a' = \{P' \in \mathcal{P} \mid \forall x, y, x' y'. \ (\bigwedge_{P \in a} P(x, y)) \land R(x, y, x', y') \rightarrow P'(x', y')\}$$

Example of Analysis Result

$\mathcal{P} = \{false, 0 < x, 0 <= x, 0 < y, x < y, x = 0, y = 1, x < 1000, 1000 \leq x\}$

```
x = 0;
y = 1;
// 0<y, x<y,x=0,y=1, x<1000
// 0<y, 0≤x, x<y
while (x < 1000) {
   // 0<y, 0≤x, x<y, x<1000
   x = x + 1;
   // 0<y, 0≤x, 0<x
   y = 2*x;
   // 0<y, 0≤x, 0<x, x<y
   y = y + 1;
   // 0<y, 0≤x, 0<x, x<y
   print(y);
}
// 0<y, 0≤x, x<y, 1000 ≤ x
```

Formulation in terms of Removing Predicates

At program entry: $\top$, which is:

Formulation in terms of Removing Predicates

At program entry: $\top$, which is: $\emptyset$ of predicates
At all other points: $\bot$, which is:

Formulation in terms of Removing Predicates

At program entry: $\top$, which is: $\emptyset$ of predicates
At all other points: $\bot$, which is: the set of all predicates $\mathcal{P}$
Lattice elements grow in CFG $\leadsto$ the set of predicates decrease

Formulation in terms of Removing Predicates

At program entry: $\top$, which is: $\emptyset$ of predicates

At all other points: $\bot$, which is: the set of all predicates $\mathcal{P}$

Lattice elements grow in CFG $\leadsto$ the set of predicates decrease

We remove predicates that do not hold

Limitations of Conjunctions

```
if (x > 0) {
  y = x
}
if (x > 0) {
  if(y > 0) 1/x
  else error
}
```

Disjunctive Analysis

Disjunction of conjunctions.
Sets of sets.
α and γ
Approximations: apply per disjunct.
Powerdomain. Power and cost of powerdomains.

Reachability tree

Path Feasibility Checking

Adding Predicates to Remove Infeasible Paths

Adding weakest preconditions
Adding strongest postconditions
Increasing the power of generalization:

- do not add complex formulas, but their parts
- no need to add sp or wp, but anything that forms a sufficiently annotated Hoare proof that this path is infeasible