Fixed points. Abstract Interpretation

Viktor Kunčak

Fixpoints

Definition: Given a set A and a function $f : A \to A$ we say that $x \in A$ is a fixed point (fixpoint) of f if $f(x) = x$.

Definition: Let $(A, \leq)$ be a partial order, let $f : A \to A$ be a monotonic function on $(A, \leq)$, and let the set of its fixpoints be $S = \{x \mid f(x) = x\}$. If the least element of S exists, it is called the **least fixpoint**, if the greatest element of S exists, it is called the **greatest fixpoint**.

Fixpoints

Let $(A, \sqsubseteq)$ be a complete lattice and $G : A \rightarrow A$ a monotonic function.

Definition:
Post $= \{x \mid G(x) \sqsubseteq x\}$ - the set of *postfix points* of G
(e.g. $\top$ is a postfix point)
Pre $= \{x \mid x \sqsubseteq G(x)\}$ - the set of *prefix points* of G
Fix $= \{x \mid G(x) = x\}$ - the set of *fixed points* of G.

Note that Fix $\subseteq$ Post.

Tarski's fixed point theorem

Theorem: Let $a = \sqcap \mathrm{Post}$. Then a is the least element of Fix (dually, $\sqcup \mathrm{Pre}$ is the largest element of Fix).

Proof:

Let x range over elements of Post.

- applying monotonic G from $a \sqsubseteq x$ we get $G(a) \sqsubseteq G(x) \sqsubseteq x$
- so $G(a)$ is a lower bound on Post, but a is the greatest lower bound, so $G(a) \sqsubseteq a$
- therefore $a \in \mathrm{Post}$
- Post is closed under G, by monotonicity, so $G(a) \in \mathrm{Post}$
- a is a lower bound on Post, so $a \sqsubseteq G(a)$
- from $a \sqsubseteq G(a)$ and $G(a) \sqsubseteq a$ we have $a = G(a)$, so $a \in \mathrm{Fix}$
- a is a lower bound on Post so it is also a lower bound on a smaller set Fix

In fact, the set of all fixpoints Fix is a lattice itself.

Tarski's fixed point theorem

Tarski's Fixed Point theorem shows that in a complete lattice with a monotonic function G on this lattice, there is at least one fixed point of G, namely the least fixed point $\sqcap$Post.

- Tarski's theorem guarantees fixpoints in complete lattices, but the above proof does not say how to find them.
- How difficult it is to find fixpoints depends on the structure of the lattice.

Let G be a monotonic function on a lattice with a bottom element. Let $a_0 = \bot$ and $a_{n+1} = G(a_n)$. We obtain a sequence $\bot \sqsubseteq G(\bot) \sqsubseteq G^2(\bot) \sqsubseteq \cdots$. Let $a_* = \bigsqcup_{n \geq 0} a_n$.

Lemma: The value a_* is a prefix point. Proof hint: $G(G^n(\bot)) \sqsubseteq G(a_*)$

Observation: a_* need not be a fixpoint. Example: take $f : [0,1] \to [0,1]$,

$$f(x) = \begin{cases} \frac{x+1}{2}, & \text{if } 0 \leq x < 1 \\ 2, & \text{if } x \geq 1 \end{cases}$$

Omega continuity

Definition: A function G is ω-continuous if for every chain $x_0 \sqsubseteq x_1 \sqsubseteq \ldots \sqsubseteq x_n \sqsubseteq \ldots$ such that $\bigsqcup_{i \geq 0} x_i$ exists, we have that $\bigsqcup_{i \geq 0} G(x_i)$ exists and

$$G(\bigsqcup_{i \geq 0} x_i) = \bigsqcup_{i \geq 0} G(x_i)$$

Lemma: For an ω-continuous function G, the value $a_* = \bigsqcup_{n \geq 0} G^n(\bot)$ is the least fixpoint of G.
(Proof on the next slide.)

Iterating sequences and omega continuity

Lemma: For an ω-continuous function G, the value $a_* = \bigsqcup_{n\geq 0} G^n(\bot)$ is the least fixpoint of G. (Special case of (W) Kleene fixed-point theorem.)

Proof:

- By definition of ω-continuous we have $G(\bigsqcup_{n\geq 0} G^n(\bot)) = \bigsqcup_{n\geq 0} G^{n+1}(\bot) = \bigsqcup_{n\geq 1} G^n(\bot)$.
- But $\bigsqcup_{n\geq 0} G^n(\bot) = \bigsqcup_{n\geq 1} G^n(\bot) \sqcup \bot = \bigsqcup_{n\geq 1} G^n(\bot)$ because $\bot$ is the least element of the lattice.
- Thus $G(\bigsqcup_{n\geq 0} G^n(\bot)) = \bigsqcup_{n\geq 0} G^n(\bot)$ and a_* is a fixpoint.

Now let's prove it is the least. Let c be such that $G(c) = c$. We want $\bigsqcup_{n\geq 0} G^n(\bot) \sqsubseteq c$. This is equivalent to $\forall n \in \mathbb{N}. G^n(\bot) \sqsubseteq c$.
We can prove this by induction : $\bot \sqsubseteq c$ and if $G^n(\bot) \sqsubseteq c$, then by monotonicity of G and by definition of c we have $G^{n+1}(\bot) \sqsubseteq G(c) \sqsubseteq c$.

Iterating sequences and omega continuity

So, for an ω-continuous function G, the value $a_* = \bigsqcup_{n \geq 0} G^n(\bot)$ is the least fixpoint of G.

When the function is not ω-continuous, then we obtain a_* as above (we jump over a discontinuity) and then continue iterating. We then take the limit of such sequence, and the limit of limits etc., “ultimately” we obtain the fixpoint.

Patrick Cousot and Radhia Cousot (1979). Constructive versions of Tarski’s fixed point theorems. *Pacific Journal of Mathematics*. 82:1: 43–57.

Another Lemma

Lemma: For an ω-continuous monotonic function G, let $a_* = \bigsqcup_{n \geq 0} G^n(\bot)$ and suppose that s' is such that $G(s') \sqsubseteq s'$. Then $a_* \sqsubseteq s'$.

Proof hint: $G^n(\bot) \sqsubseteq s'$.

Reachable Sets of States

Given a transition system $M = (S, \mathit{Init}, r, A)$, with $\bar{r} = \{(s, s') \mid (s, a, s') \in r\}$, consider the function $G : 2^S \to 2^S$ defined by

$$G(X) = \mathit{Init} \cup \bar{r}[X]$$

Then G is monotonic and ω continuous. Its least fixed point is the set of reachable states of M.

Proving through Fixpoints of Approximate Functions

Meaning of a program (e.g. a set of reachable states) is a least fixpoint of F: $\mathit{lfp}(F)$.
Given specification s (e.g. $(pc = c) \to i \neq 42$), prove $\mathbf{lfp(F) \subseteq s}$ (s is invariant)

- if $F(s) \subseteq s$ then $\mathit{lfp}(F) \subseteq s$ and we are done
- $\mathit{lfp}(F) = \bigcup_{k \geq 0} F^k(\emptyset)$, but that is too hard to compute because it is infinite union unless, by some luck, $F^{n+1}(\emptyset) = F^n$ for some n

Instead, we search for an inductive strengthening of s: find s' such that:

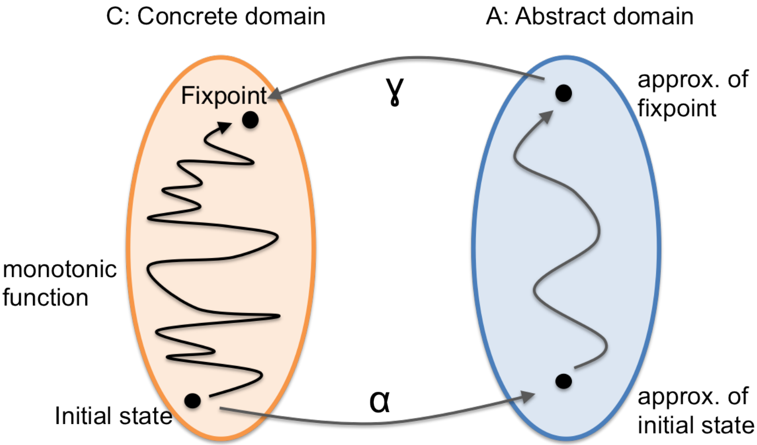
- $F(s') \subseteq s'$ (s' is inductive). If so, theorem says $\mathit{lfp}(F) \subseteq s'$
- $s' \subseteq s$ (s' implies the desired specification). Then $\mathit{lfp}(F) \subseteq s' \subseteq s$

How to find s' such that $F(s') \subseteq s'$? Iterating F is hard, so we try some simpler function $F_\#$:

- suppose $F_\#$ is *approximation*: $F(r) \subseteq F_\#(r)$ for all r
- we can find s' such that: $F_\#(s') \subseteq s'$ (e.g. $s' = F_\#^{n+1}(\emptyset) = F_\#^n(\emptyset)$)

Then: $F(s') \subseteq F_\#(s') \subseteq s'$. So, what we found is such that $s' \subseteq s$, we have $\mathit{lfp}(F) \subseteq s'$.
Abstract interpretation: automatically construct $F_\#$ using F and s

Abstract Interpretation Big Picture



Galois Connection

Galois connection (named after Évariste Galois) is defined by two monotonic functions $\alpha : C \to A$ and $\gamma : A \to C$ between partial orders $\leq$ on C and $\sqsubseteq$ on A, such that

$$\forall c, a. \quad \alpha(c) \sqsubseteq a \iff c \leq \gamma(a) \qquad (*)$$

(intuitively the condition means that c is approximated by a).

Lemma: The condition $(*)$ holds iff the conjunction of these two conditions:

$$\begin{array}{rcl} c & \leq & \gamma(\alpha(c)) \\ \alpha(\gamma(a)) & \sqsubseteq & a \end{array}$$

holds for all c and a.

Termination and Efficiency of Abstract Interpretation Analysis

Definition: A **chain** of length n is a sequence $s_0, s_1, \ldots, s_n$ such that

$$s_0 \sqsubset s_1 \sqsubset s_2 \sqsubset \ldots \sqsubset s_n$$

where $x \sqsubset y$ means, as usual, $x \sqsubseteq y \land x \neq y$

Definition: A partial order has a **finite height** n if it has a chain of length n and every chain is of length at most n.

A finite lattice is of finite height.

Example

The constant propagation lattice $\mathbb{Z} \cup \{\bot, \top\}$ is an infinite lattice of height 2. One example chain of length 2 is

$$\bot \sqsubset 42 \sqsubset \top$$

Here the γ function is given by

- $\gamma(k) = \ldots$ when $k \in \mathbb{Z}$
- $\gamma(\top) = \ldots$
- $\gamma(\bot) = \ldots$

The ordering is given by $a_1 \subseteq a_2$ iff $\gamma(a_1) \subseteq \gamma(a_2)$

Example

If a state of a (one-variable) program is given by an integer, then a concrete lattice element is a set of integers. This lattice has infinite height. There is a chain

$$\{0\} \subset \{0,1\} \subset \{0,1,2\} \subset \ldots \subset \{0,1,2,\ldots,n\}$$

for every n.

Convergence in Lattices of Finite Heignth

Consider a finite-height lattice $(L, \sqsubseteq)$ of height n and function

$$F : L \to L$$

What is the maximum length of sequence $\bot, F(\bot), F^2(\bot), \ldots$?
Give an effectively computable expression for $lfp(F)$.

Computing the Height when Combining Lattices

Let $H(L, \leq)$ denote the height of the lattice $(L, \leq)$.

Product

Given lattices $(L_1, \sqsubseteq_1)$ and $(L_2, \sqsubseteq_2)$, consider product lattice with set $L_1 \times L_2$ and potwise order

$$(x_1, x_2) \sqsubseteq (x'_1, x'_2)$$

iff ...

What is the height of the product lattice?

Exponent

Given lattice $(L, \sqsubseteq)$ and set V, consider the lattice $(L^V, \sqsubseteq')$ defined by

$$g \sqsubset' h$$

iff $\forall v \in V . g(v) \sqsubseteq h(v)$.

What is the height of the exponent lattice?

Computing the Height when Combining Lattices

Let $H(L, \leq)$ denote the height of the lattice $(L, \leq)$.

Product

Given lattices $(L_1, \sqsubseteq_1)$ and $(L_2, \sqsubseteq_2)$, consider product lattice with set $L_1 \times L_2$ and potwise order

$$(x_1, x_2) \sqsubseteq (x_1', x_2')$$

iff . . .

What is the height of the product lattice?

Exponent

Given lattice $(L, \sqsubseteq)$ and set V, consider the lattice $(L^V, \sqsubseteq')$ defined by

$$g \sqsubset' h$$

iff $\forall v \in V.g(v) \sqsubseteq h(v)$.

What is the height of the exponent lattice?

Answer: height of L times the cardinality of V.

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

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

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

Example of Analysis Result

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

```
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: $\gamma(a_0) = \{s \mid s \models 0 < x \wedge x < y\}$. We often assume states are pairs (x, y). Then $\gamma(a_0) = \{(x, y) \mid 0 < x \wedge x < y\}$.

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: $\gamma(a_0) = \{s \mid s \models 0 < x \land x < y\}$. 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 $\gamma(a_2) \subseteq \gamma(a_1)$.

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:

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

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: $\gamma(a_0) = \{s \mid s \models 0 < x \wedge x < y\}$. 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: $a_1 \sqsubseteq a_2 \rightarrow \gamma(a_1) \subseteq \gamma(a_2)$

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

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

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?
What is the height of such lattice when $\mathcal{P} = \{P_0, P_1, \ldots, P_n\}$?

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?
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 $a_1 = \{\mathit{false}\}$ 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 $\alpha(\gamma(a)) = a$.
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

$\mathcal{P} = \{\mathit{false}, 0 < x, 0 < y, x < y\}$
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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

which, after expanding γ, gives

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

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

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

Example of Analysis Result

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

```
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