Abstraction II

Property Abstraction

(based on Patrick Cousout's 2005 course "Abstract Interpretation")

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Manuel Geffken

2014-06-24

Objects

When analyzing/proving programs we have to consider "objects" that represent some part of the computation state, such as:

- Values: booleans, integers, ... $\mathcal{V}$
- Variable names: $\mathbb{X}$
- Environments: $\mathbb{X} \to \mathcal{V}$
- Stacks: assigning values to variables in the context of block-structured languages: $\bigcup_{n \leq 0}([1, n] \to (\mathbb{X} \rightharpoonup \mathcal{V}))$

```
begin
  new X,Y;
  X := 10; Y := 0;
  begin
    new X;
    X := 20;
     ...
  end;
  ...
end;
```

1:	x	10	y	0
2:	x	20		

$$\{1 \mapsto \{x \mapsto 10, y \mapsto 0\},$$
$$2 \mapsto \{x \mapsto 20\}\}$$

- Heaps: dynamic allocation;
- Control points: procedure names, labels, . . . ;
- States: control & memory states.

- Finite prefix traces;
- Maximal finite or infinite traces for deterministic programs);
- Sets of maximal finite or infinite traces (for nondeterministic programs);
- ...

Properties

Properties are "sets of objects" (which have that property).
Examples:

- odd naturals: $\{1, 3, 5, \ldots, 2n+1, \ldots\}$
- even integers: $\{2z \mid z \in \mathbb{Z}\}$
- values of integer variables: $\{z \in \mathbb{Z} \mid \text{minint} \leq z \leq \text{maxint}\}$
- values of maybe uninitialized integer variables: $\{z \in \mathbb{Z} \mid \text{minint} \leq z \leq \text{maxint}\} \cup \{\Omega_m \mid m \in \mathcal{M}\}$ where $\mathcal{M}$ is a set of error messages

Properties (continued)

- equality of two variables $\mathtt{x}$ and $\mathtt{y}$:
  $\{\rho \in \mathbb{X} \rightharpoonup \mathcal{V} \mid \mathtt{x}, \mathtt{y} \in \mathit{dom}(\rho) \wedge \rho(\mathtt{x}) = \rho(\mathtt{y})\}$
- invariance property: (of a program with states in Σ):
  $I \in \mathcal{P}(\Sigma)$
- trace property: $T \in \mathcal{P}(\Sigma^{\vec{\infty}})$
- trace semantics property: $P \in \mathcal{P}(\mathcal{P}(\Sigma^{\vec{\infty}}))$
- ...

The complete lattice of concrete properties

The set of properties $\mathcal{P}(\Sigma)$ of objects in Σ is a complete boolean lattice:

$$\langle \mathcal{P}(\Sigma), \subseteq, \emptyset, \Sigma, \cup, \cap, \neg \rangle$$

where

- A property $P \in \mathcal{P}(\Sigma)$ is the set of objects which have the property P.
- $\subseteq$ is logical implication since $P \subseteq Q$ means that all objects with property P have property Q ($o \in P \implies o \in Q$)

- $\emptyset$ is false
- Σ is true
- $\cup$ is disjunction (objects which have property P or have property Q belong to $P \cup Q$)
- $\cap$ is conjunction (objects which have property P and have property Q belong to $P \cap Q$)
- $\neg$ is negation (objects not having property P are those in $\Sigma \setminus P$)

Abstraction, informal introduction

- Abstraction replace someting "concrete" [1] by a schematic description that account for some, and in general not all properties, either known or inferred i.e. an "abstract" model or concept
- In practice, such an abstract model of a concrete object o
  - can describe some of the properties of the concrete object
  - cannot describe all properties of this concrete object [2]

[1] real, actual, material, corporeal, ...

[2] since otherwise this property would have to be "exactly that object" i.e. $\{o\}$

- So an abstraction of properties in $\mathcal{P}(\Sigma)$ of objects in Σ is essentially a subset $A \subseteq \mathcal{P}(\Sigma)$ such that:
  - The properties in A are the concrete properties that can be described exactly by the abstraction, without any loss of information
  - The properties in $\mathcal{P}(\Sigma) \setminus A$ are the properties that cannot be described exactly by the abstraction, and have to be referred to by being approximated in some way or another by abstract properties in A

Intuitive example 1 of abstraction

Cars $\xrightarrow{\alpha}$ Color [3]

- A concrete property of cars is a set of cars
- It can be abstracted by the set of their colors
- A color is a set of cars
- An abstract property of cars is a set of cars which, whenever it contains one car of some color, also contains all cars of that color

[3] Formally, if $t \in$ Cars $\to$ Color yields the color $t(c)$ of a car $c \in$ Cars then the abstraction $P \in \mathcal{P}(\text{Cars})$ is $\alpha(P) = \{t(c) \mid c \in P\}$ and the set of cars described by an abstract property $T \subseteq$ Colors is $\gamma(T) = \{c \in Cars \mid t(c) \in T\}$.

Intuitive example 2 of abstraction

Scientific papers $\rightarrow$ set of keywords [4]

- A concrete property of scientific papers is a set of scientific papers
- Each scientific paper is abstracted by a list of keywords
- A property of scientific papers can be abstracted by the list of keywords appearing in all papers with that property
- An abstract property of scientific papers is therefore a set of papers which have all keywords belonging to the list

[4] Can be written formally as well.

Abstraction, definition of concrete and abstract properties

Abstraction is a reasoning/computation such that:

- Only some properties $A \subseteq \mathcal{P}(\Sigma)$ of the objects in Σ can be used;
- The properties $P \in A$ that can be used are called abstract;
- The properties $P \in \mathcal{P}(\Sigma)$ are called concrete;

- Abstract reasonings/computations involve sound approximations, in that:
    - The concrete properties that are also abstract can be used in the abstract reasoning/computation "as is", without any loss of information;
    - The concrete properties $P \in \mathcal{P}(\Sigma) \setminus A$ which are not abstract cannot be used in the reasoning/computation and therefore must be approximated by some other abstract property $\overline{P} \in A$, which, since $P \neq \overline{P}$, involves some form of approximation.

Direction of abstraction

- When approximating a concrete property $P \in \mathcal{P}(\Sigma)$, by an abstract property $\overline{P} \in A$, with $\overline{P} \neq P$, a relation must be established between the concrete P and abstract property P to estabblish that

  "$\overline{P} \in A$ is an approximation/abstraction of $P \in \mathcal{P}(\Sigma)$"

  so as to ensure the soundness of the reasoning in the abstract with respect to the concrete, exact one.

- We consider essentially two cases:
  - Approximation from above: $P \subseteq \overline{P}$
  - Approximation from below: $P \supseteq \overline{P}$
- Other relations can be considered (e.g. probabilistic properties)
- The two notions are dual so formally only one need to be studied formally (approximation from above)
- In practice, useful approximation from below are much harder to discover

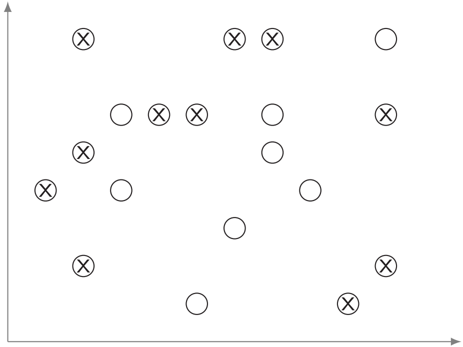
Abstraction from below

x: points which have the concrete property P

o: points which have the abstract property P

- To answer the question "$\langle x, y \rangle \in P$?" using only $\overline{P}$ (such that $P \supseteq \overline{P}$):
  - If $\langle x, y \rangle \notin \overline{P}$ then "I don't know"
  - If $\langle x, y \rangle \in \overline{P}$ then "Yes"

Abstraction from above



x: points which have the concrete property P

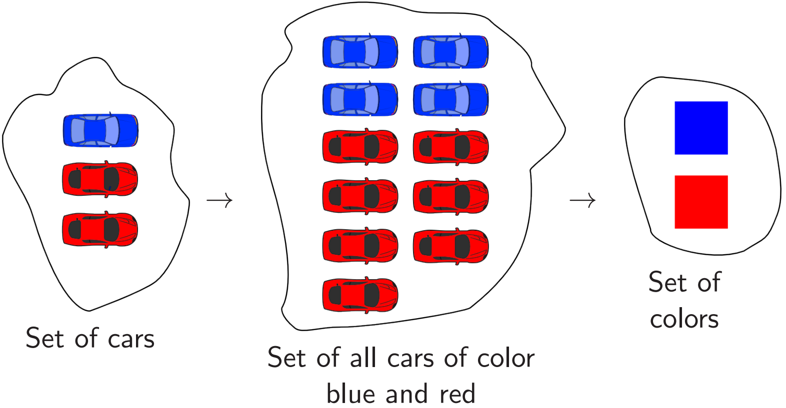
o: points which have the abstract property P

- To answer the question "$\langle x, y\rangle \in P$?" using only $\overline{P}$ (such that $P \subseteq \overline{P}$):
    - If $\langle x, y\rangle \in \overline{P}$ then "I don't know"
    - If $\langle x, y\rangle \notin \overline{P}$ then "Yes"

Why can an abstraction from above be “simpler” than the original concrete property?

- The concrete property is a set of objects
  - The objects are complex
  - The set can be infinite
  - In general their exists no suitable computer repre- sentation of the concrete property
- The abstract property is a larger set of objects
  - Larger structures are in general even more expensive to store in the computer memory/compute with than smaller ones

- but, well-chosen larger structures can have simpler encodings which can be exploited for memorization and computation
- Example:



What to do in absence of (upper) abstraction?

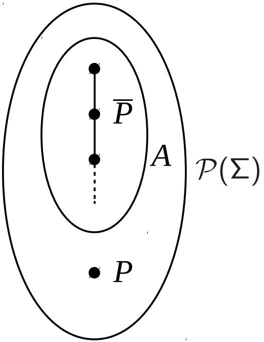
- Assume a mechanized reasoning about a computer sys- tems with objects/states Σ, we use an abstraction $A \in \mathcal{P}(\Sigma)$
- Assume concrete properties $P \in \mathcal{P}(\Sigma)$ which cannot be expressed in the abstract, must be approximated from above by $\overline{P} \in A : P \supseteq \overline{P}$
- How should the mechanized reasoning proceed when some property P has has <u>no</u> abstraction $P \in A$ from above $(\forall \overline{P} \in A : P \not\supseteq \overline{P})$?
  - loop?

  - block?
  - ash for help?
  - fail?
  - answer something sensible!
- The only way to be always able to say something sensible for all $P \in \mathcal{P}(\Sigma)$ is to assume that $\Sigma \in A$:

  Any concrete property should be approximable by "I don't know" (i.e. $\Sigma \in A, \Sigma$ meaning "true")

Minimal abstractions

- Assume concrete properties $P \in \mathcal{P}(\Sigma)$ must be approximated from above by $\overline{P} \in A \subset \mathcal{P}(\Sigma)$ such that $P \subseteq \overline{P}$
- The smaller the abstract property P is, the most precise the approximation will be
- There might be no minimal abstract property at all in A

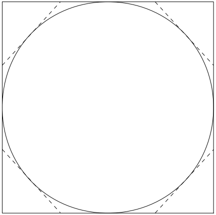


- If a concrete property $P \in \mathcal{P}(\Sigma)$ has minimal upper approximations $P \in A$:
  - $P \subseteq \overline{P}$
  - $\not\exists P' : P \subseteq P' \sqsubset \overline{P}$

  then such minimal approximations are more precise than the non-minimal ones
- So minimal abstract upper approximations, if any, should be prefered
- In particular, an abstract property $P \in A$ is best approximated by itself

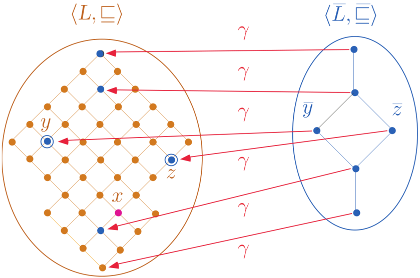
In absence of minimal abstraction

- A classical example of absence of minimal abstract upper-approximations is that of a disk with no minimal convex polyhedral approximation
- $\Sigma = \mathbb{R} \times \mathbb{R}$
- $A =$ convex polyhedra
- Absence of minimal approximation is shown by Euclide's construction:



- In absence of minimal approximations, the approximation $P \subset P_1$ can always be approximated by a better one $P \subset P_2 \subset P_1$!
- Some arbitrary choice has to be performed. This case will be studied later. So, in the following, we assume the existence of minimal approximations

Example of minimal abstractions in absence of a best approximation



- x can be approximated by $y = \gamma(\overline{y})$ and $z = \gamma(\overline{z})$ but x and z are not comparable

- The other possible upper approximations would be less precise (than both y and z in that particular example)
- Notice that γ cannot be the upper adjoint of a Galois connection since it is not a complete meet morphism:

$$\gamma(\overline{y}) \wedge \gamma(\overline{z}) \neq \gamma(\overline{y} \sqcap \overline{z})$$

Which minimal abstraction to choose?

- If there are several minimal possible abstract approximations $\overline{P}_1, \overline{P}_2, \ldots$ [5]
- Example: rule of signs
  - In "1+0", it is better to chose '+', because of the rule '+' + '+' = '+', while '+' + '-' yields no information ("I don't know")
  - In "(-1)+0", it is better to chose '-', because of the rule '-' + '-' = '-', while '-' + '+' yields no information ("I don't know")
  - Both cases have to be tried (backtracking)

[5] There can even be infinitely many ones

- In absence of unicity of the minimal approximation, it may be necessary to try all of them (at the cost of an exponential blow up of the mechanical reasoning).
- To start with, we will assume the existence of a best approximation (i.e. a unique minimal upper approximation).

Best abstraction

- A very handy choice of the abstract properties $A \subseteq \mathcal{P}(\Sigma)$ is when every concrete property P has a best approximation $\overline{P} \in A$:
  - $P \subseteq \overline{P}$
  - $\forall \overline{P}' \in A : (P \subseteq \overline{P}') \implies (\overline{P} \subseteq \overline{P}')$
- It follows that $\overline{P}$ is the glb of the over-approximations of P in A:

$$\overline{P} = \bigcap \{\overline{P}' \in A \mid P \subseteq \overline{P}'\} \in A$$

Proof.

- We have $\forall \overline{P} \in \{\overline{P}' \in A \mid P \subseteq \overline{P}'\} : P \subseteq \overline{P}$ so $P \subseteq \bigcap\{\overline{P}' \in A \mid P \subseteq \overline{P}'\}$ by definition of glb
- Moreover
  $\forall \overline{P}' \in A : (P \subseteq \overline{P}') \implies (\bigcap\{\overline{P}'' \in A \mid P \subseteq \overline{P}''\} \subseteq \overline{P}')$
  because from the premise we get $\overline{P}' \in \{\overline{P}'' \in A \mid P \subseteq \overline{P}''\}$
  and by definition of glb it holds $\bigcap\{\overline{P}'' \in A \mid P \subseteq \overline{P}''\} \subseteq \overline{P}'$.
  There can only be one such smallest abstraction of P.
- It follows that $\overline{P} = \bigcap\{\overline{P}' \in A \mid P \subseteq \overline{P}'\}$
- So $\left(\exists \overline{P} : (P \subseteq \overline{P}) \wedge (\forall \overline{P}' \in A : (P \subseteq \overline{P}') \implies (\overline{P} \subseteq \overline{P}')\right)$
  $\Leftrightarrow \overline{P} = \bigcap\{\overline{P}' \in A \mid P \subseteq \overline{P}'\} \in A$

$\square$

The abstract domain is a Moore family

Theorem

The hypothesis that any concrete property $P \in \mathcal{P}(\Sigma)$ has a best abstraction $P \in A$, implies that

The abstract domain A is a Moore family.

Proof.

Let $X \subseteq A$ be a set of abstract properties. Its intersection $\bigcap X$ has a best approximation $\overline{P} \in A$. We have therefore

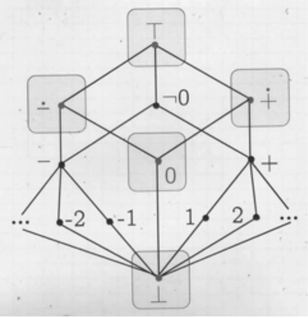
$$\overline{P} = \bigcap \{ \overline{P}' \in A \mid \bigcap X \subseteq \overline{P}' \}$$

But $\forall \overline{P}' \in X : \bigcap X \subseteq \overline{P}'$ and $X \subseteq A$ so $X \subseteq \{ \overline{P}' \in A \mid \bigcap X \subseteq \overline{P}' \}$ and therefore $\bigcap \{ \overline{P}' \in A \mid \bigcap X \subseteq \overline{P}' \} \subseteq \bigcap X$ by def. of glb. By antisymmetry ($\overline{P} \subseteq \bigcap X$ as $\overline{P}$ is an approximation), $\bigcap X = \bigcap \{ \overline{P}' \in A \mid \bigcap X \subseteq \overline{P}' \} = \overline{P} \in A$, proving A to be a Moore family. □

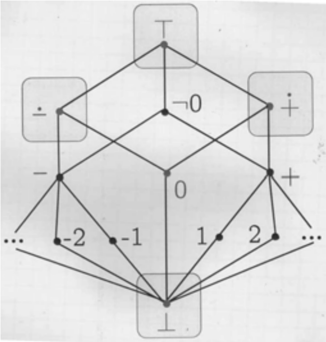
In particular $\bigcap \emptyset = \Sigma \in A$, which is consistent with our hypothesis that A should contain Σ to have the ability to express "I don't know".

Example and counter-example of Moore family based abstraction

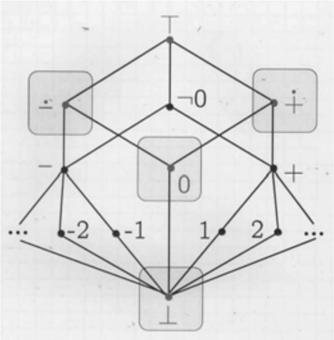
- Example: rule of signs with best approximation of 0



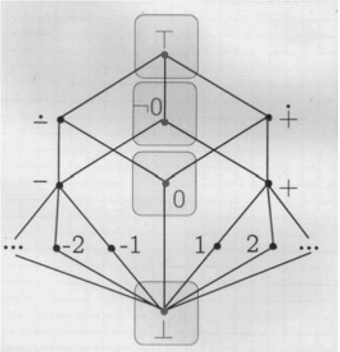
- Counter-example: rule of signs without best approximation of 0

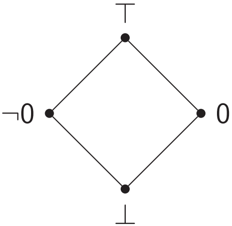


- Counter-example: rule of sign without upper approximation of "different from zero"



- Example: abstraction to 0 or different from 0





A Moore family in a poset is a complete lattice

Theorem

Let $\langle P, \sqsubseteq \rangle$ be a topped poset and $M \subseteq P$ be a Moore family then $\langle M, \sqsubseteq \rangle$ is a complete lattice $\langle M, \sqsubseteq, \sqcap M, \top \rangle$.

Proof.

Since $\langle P, \sqsubseteq \rangle$ is a poset and $M \subseteq P$, $\langle M, \sqsubseteq \rangle$ is a poset. Being a Moore family it is topped and any subset $S \subseteq M$ has $\sqcap S \in M$ so $\sqcap$ is the meet in M. It follows that M is a complete lattice, which lub is:

$$\sqcup S = \sqcap \{ y \in M \mid \forall x \in S : x \sqsubseteq y \} \in M$$

The infimum is $\sqcap M \in M$. $\square$

Closure operator based abstraction

Assume that the abstract domain A is a Moore family of the concrete domain $\langle \mathcal{P}(\Sigma), \subseteq, \emptyset, \Sigma, \cup, \cap, \neg \rangle$. The the abstraction map is

$$\rho \in \mathcal{P}(\Sigma) \rightarrow A$$
$$\rho(P) \stackrel{def}{=} \bigcap \{\overline{P} \in A \mid P \subseteq \overline{P}\}$$

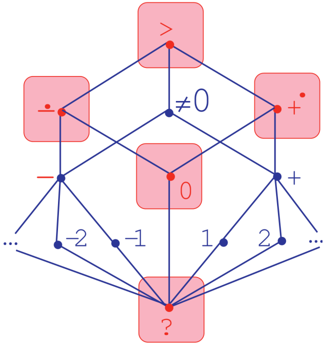
Then ρ is an upper closure operator on $\mathcal{P}(\Sigma)$. That is ρ is

- Extensive: $P \subseteq \rho(P)$
- Increasing: $P \subseteq P' \Rightarrow \rho(P) \subseteq \rho(P')$
- Idempotent: $\rho(\rho(P)) = \rho(P)$

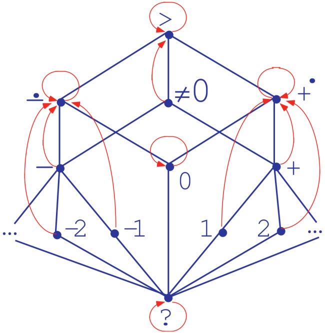
Proof.

ρ is the closure operator induced by the Moore family, a result simply depending on the fact that $\langle \mathcal{P}(\Sigma), \subseteq, \emptyset, \Sigma, \cup, \cap \rangle$ is a complete lattice. □

Example of abstraction map



Moore family



Abstraction map
(closure operator)

Equivalent specification of an abstraction by a Moore family and a closure operator

In case of existence of a best abstraction, it is equivalent to specify the abstraction domain A

1. as a Moore family $\mathcal{M}$
2. as a closure operator ρ

Proof.

- Given $\mathcal{M}$ define $\rho(P) = \cap\{\overline{P} \in \mathcal{M} \mid P \subseteq \overline{P}\} \in \mathcal{M}$ so that $A = \mathcal{M} = \rho(\mathcal{P}(\Sigma))$
- Conversely, given a closure operator ρ, define $A = \rho(\mathcal{P}(\Sigma)) = \{\rho(P) \mid P \in \mathcal{P}(\Sigma)\}$ which is therefore the set of fixpoints of ρ (because ρ is idempotent) whence a Moore family since ρ operates on a complete lattice.

□

Examples of specifications of an abstraction by a Moore family and a closure operator

- The most imprecise abstraction is “I don’t know”
  - $\mathcal{M} = \{\Sigma\}$
  - $\rho = \lambda P.\Sigma$
- The most precise abstraction is “identity”
  - $\mathcal{M} = \mathcal{P}(\Sigma)$
  - $\rho = \lambda P.P$

Generalizing to complete lattices

- The reasoning on abstractions of concrete properties $\langle \mathcal{P}(\Sigma), \subseteq, \emptyset, \Sigma, \cup, \cap, \neg \rangle$ to an abstract domain which, in case of best abstraction is a Moore family, whence a complete lattice, can be generalized to an arbitrary concrete complete lattice $\langle L, \sqsubseteq, \bot, \top, \sqcup, \sqcap \rangle$
- This allows a compositional approach where $\langle L, \sqsubseteq, \bot, \top, \sqcup, \sqcap \rangle$ is abstracted to $\langle A_1, \sqsubseteq_1, \bot_1, \top_1, \sqcup_1, \sqcap_1 \rangle$ which itself can be further abstracted to $\langle A_2, \sqsubseteq_2, \bot_2, \top_2, \sqcup_2, \sqcap_2 \rangle$, ...